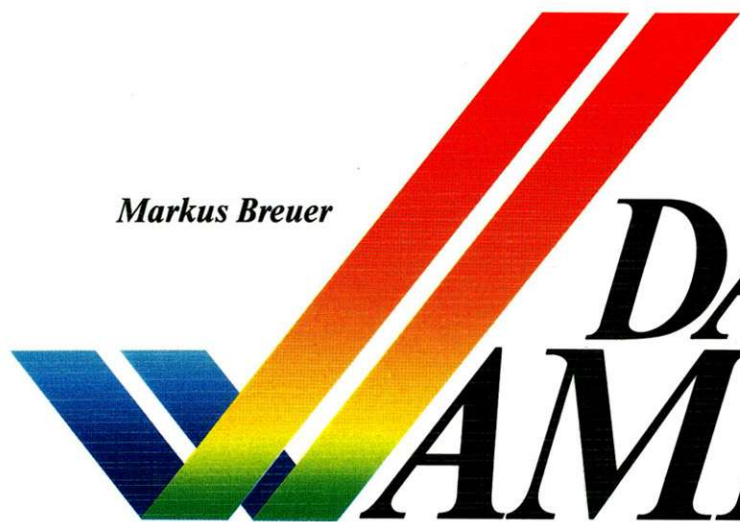


Markt&Technik

Markus Breuer

A stylized logo for the Amiga computer system. It features a blue 'V' shape on the left, followed by a diagonal bar with a rainbow gradient (red, orange, yellow, green) that extends upwards and to the right. The text 'DAS AMIGA HANDBUCH' is overlaid on the right side of the logo.

DAS AMIGA HANDBUCH

*Kennenlernen und Anwenden der neuen Computer-Technologie:
Workbench · Anwendersoftware · CLI
Grafik- und Soundfähigkeiten
Mausbedienung*

Das AMIGA-Handbuch

Markus Breuer

Das AMIGA-Handbuch

Kennenlernen und Anwenden
der neuen Computer-Technologie:
Workbench · Anwendersoftware · CLI
Grafik- und Soundfähigkeiten
Mausbedienung

Markt & Technik Verlag

Breuer, Markus

Das AMIGA-Handbuch : Kennenlernen u. Anwenden d. neuen Computer-Technologie: Workbench,
Anwendersoftware, CLI, Grafik- u. Soundfähigkeiten, Mausbedienung / Markus Breuer. —
Haar bei München : Markt-und-Technik-Verlag, 1986.

ISBN 3-89090-228-6

Die Informationen im vorliegenden Buch werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht.

Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.

Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen.

Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autoren können
für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine
Haftung übernehmen.

Für Verbesserungsvorschläge und Hinweise auf Fehler sind Verlag und Herausgeber dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien.
Die gewerbliche Nutzung der in diesem Buch gezeigten Modelle und Arbeiten ist nicht zulässig.

»Commodore-Amiga« ist eine Produktbezeichnung der Commodore Büromaschinen GmbH, Frankfurt,
die ebenso wie der Name »Commodore« Schutzrecht genießt. Der Gebrauch bzw. die Verwendung bedarf der Erlaubnis
des Schutzrechtsinhabers.

Amiga ist eine Produktbezeichnung der Commodore-Amiga Inc., USA.

Graphicraft, Textcraft, Musicraft sind Produktbezeichnungen der Commodore-Amiga Inc., USA.

Amiga-BASIC ist eine Produktbezeichnung der Microsoft Inc., USA.

Deluxe Paint ist eine Produktbezeichnung von Electronic Arts, USA.

15 14 13 12 11 10 9 8 7 6 5 4 3
89 88 87 86

ISBN 3-89090-228-6

© 1986 by Markt & Technik, 8013 Haar bei München

Alle Rechte vorbehalten

Einbandgestaltung: Grafikdesign Heinz Rauner

Druck: Jantsch, Günzburg

Printed in Germany

Inhaltsverzeichnis

	Vorwort	15
1	Vorhang auf: Der Amiga !	19
1.1	Äußerlichkeiten	20
1.2	Innere Werte	25
1.3	Verbindungen	27
1.4	Was man außer der Hardware noch bekommt	28
1.5	Fähigkeiten des Amiga	30
1.5.1	Grafik-Fähigkeiten	30
1.5.2	Sound-Fähigkeiten	33
1.5.3	Sprach-Fähigkeiten	34
1.5.4	Sound- und Grafik-Hardware	34
1.6	Amiga Software	35
1.6.1	Die Workbench	36
1.6.2	Das CLI	37
1.6.3	Multitasking	37
1.7	Historie	38
2	Auf der Werkbank	41
2.1	Starten des Amiga	41
2.1.1	Der Einschaltvorgang	41
2.1.2	Gezielter Neustart	45
2.2	Die Maus	46
2.2.1	Funktionsweise der Maus	47
2.2.2	Erste Maus-Spiele	49
2.2.3	Die Knöpfe der Maus	50
2.3	Menüs	53
2.3.1	Aufrufen von Menüs	53
2.3.2	Das Erteilen eines Menü-Befehls	54
2.4	Fenster	56

2.4.1	Bewegen von Fenstern	56
2.4.2	Überlappende Fenster	58
2.4.3	Fenster-Gadgets	58
2.5	Das Anlegen von Sicherungskopien	60
2.5.1	Vorbereitungen	61
2.5.2	Kopieren mit einem Diskettenlaufwerk	62
2.5.3	Kopieren mit zwei Diskettenlaufwerken	64
2.5.4	Abschluß des Kopiervorgangs	64
3	Objekte der Workbench	67
3.1	Disketten	67
3.1.1	Auswählen und Verschieben von Disketten	68
3.1.2	Das Diskettenfenster	69
3.1.3	Aufräumen einer Diskette	73
3.1.4	Umbenennen einer Diskette	73
3.1.5	Detaillierte Informationen über eine Diskette	75
3.1.6	Initialisieren und Löschen von Disketten	77
3.2	Schubladen	78
3.2.1	Aufgabe von Schubladen	78
3.2.2	Umgang mit Schubladen	79
3.2.3	Erzeugen neuer Schubladen	83
3.2.4	Löschen von Schubladen	83
3.2.5	Schnappschüsse und Schubladen	85
3.2.6	Spezielle Schubladen	86
3.2.6.1	Disketten als Schubladen	86
3.2.6.2	Mülleimer	87
3.3	Weitere Icons auf der Workbench	88
3.3.1	Werkzeuge und Projekte	89
3.3.2	Operationen für Werkzeuge und Projekte	90
3.4	Alle Befehle der Workbench	91
4.	Werkzeuge und Projekte	93
4.1	Werkzeuge	93
4.1.1	Öffnen eines Werkzeugs	94
4.1.2	Fenster eines Werkzeugs	95
4.1.3	Menü-Leiste eines Werkzeugs	96
4.1.4	Info-Fenster eines Werkzeugs	97
4.2	Projekte	99
4.2.1	Das Werkzeug Notepad	100
4.2.2	Ein Project-Menü	101
4.2.3	Möglichkeiten von Notepad	104
4.3	Zusammenhänge	105
4.3.1	Öffnen eines Projektes	105

4.3.2	Das Info-Fenster eines Projektes	106
4.4.	Multitasking	107
4.4.1	Vorteile des Multitaskings	108
4.4.2	Mögliche Probleme des Multitaskings	109
4.5	Screens	110
4.6	Ein wichtiges Werkzeug: Preferences	112
4.6.1	Das Hauptfenster von Preferences	112
4.6.2	Ändern des Mauszeigers	117
4.6.3	Die Druckeranpassung	117
5.	Intuition	121
5.1	Die Maus	122
5.1.1	Funktionsprinzip der Maus	123
5.1.2	Der Mauszeiger	123
5.1.3	Die schlafende Maus	125
5.1.4	Die Maustasten	125
5.1.5	Die "Tastatur-Maus"	126
5.2	Menüs	127
5.2.1	Ein Menü	128
5.2.2	Menü-Punkte	128
5.2.3	Sub-Menüs	130
5.2.4	Markierte Menü-Punkte	131
5.2.5	Tastatur-Äquivalente	132
5.3	Fenster	133
5.3.1	Philosophie von Fenstern	133
5.3.2	Ein typisches Fenster	134
5.3.3	Rollbalken	136
5.3.4	Fenster und Werkzeuge	137
5.4	Gadgets	138
5.4.1	Tasten und Sensoren-Gadgets	139
5.4.2	Schiebe- und Drehregler	140
5.4.3	Text-Gadgets	141
5.4.4	Kombinierte Gadgets	142
5.5	Requester	143
5.5.1	Erscheinen von Requestern	143
5.5.2	Requester und Fenster	144
5.5.3	Besonderheiten von Requestern	145
5.5.4	Alerts	146
5.6	Screens	146
5.6.1	Bedienung von Screens	147
5.6.2	Mögliche Probleme mit Screens	148

6	Arbeiten mit dem Amiga	151
6.1	Graphicraft	151
6.1.1	Grundzüge und Konzepte von Graphicraft	152
6.1.2	Überblick über die Menüs von Graphicraft	153
6.1.2.1	Das Project-Menü	155
6.1.2.2	Das Edit-Menü	157
6.1.2.3	Das Special-Menü	158
6.1.2.4	Das Color-Menü	161
6.1.2.5	Das Shape-Menü	163
6.1.2.6	Das Brush-Menü	165
6.1.3	Farbpaletten-Animation	165
6.2	Deluxe Paint	168
6.2.1	Das Starten von Deluxe Paint	169
6.2.2	Grundzüge von Deluxe Paint	170
6.2.3	Farben	172
6.2.3	Pinsel	175
6.2.3.1	Vordefinierte Pinselformen	175
6.2.3.2	Selbstdefinierte Pinselformen	176
6.2.4	Linien und Kurven	177
6.2.5	Flächen	179
6.2.6	Text	180
6.2.7	Andere Hilfsmittel	181
6.2.8	Die Menüs von Deluxe Paint	184
6.2.8.1	Das Picture-Menü	184
6.2.8.2	Das Brush-Menü	186
6.2.8.2	Das Modes-Menü	188
6.2.8.4	Das Font-Menü	189
6.2.8.5	Das Prefs-Menü	189
6.2.9	Farbpaletten-Animation	190
6.2.10	Tastatur-Abkürzungen	191
6.3	Textcraft	193
6.3.1	Ein erster Überblick über Textcraft	194
6.3.2	Text eingeben	196
6.3.3	Text ändern	197
6.3.4	Textcraft-Projekte	199
6.3.5	Ändern des Text-Stils	200
6.3.6	Formatieren des Textes	201
6.3.7	Seitenformat	203
6.3.8	Text-Formulare	205
6.3.9	Textpassagen finden und ersetzen	206
6.3.10	Tastaturabkürzungen	207
6.4	Musicraft	209
6.4.1	Aufbau des Programms	210

6.4.2	Der Kompositionsteil	211
6.4.3	Die Tastatur	214
6.4.4	Der Synthesizer	215
6.5	Ausblick	219
7	Grundlagen des CLI	223
7.1	Was ist das CLI	223
7.2	Das Einschalten des CLIs	225
7.2.1	Vorarbeiten	225
7.2.2	Sichtbarmachen des CLI	226
7.2.3	Starten des CLI und erste Schritte	227
7.2.4	Verlassen des CLI	229
7.2.5	Aufrufen von CLI-Befehlen	230
7.3	CLI-Tasks	232
7.3.1	Mehrere CLI-Fenster	232
7.3.2	Hintergrund-Befehle	233
7.4	Beeinflussung laufender Programme	234
8	Dateien und Dateiverzeichnisse	235
8.1	Grundlagen	235
8.1.1	Dateien	235
8.1.2	Dateiverzeichnisse	236
8.1.3	Das hierarchische Dateisystem	237
8.1.4	Geräte	238
8.1.4.1	Physikalische Geräte	238
8.1.4.2	Logische Geräte	239
8.1.5	Zusammenhänge	240
8.2	Anwendung	241
8.2.1	Gebrauch von Dateiverzeichnissen	241
8.2.1.1	Das aktuelle Dateiverzeichnis	241
8.2.1.2	Die Wurzel des Dateiverzeichnisses	242
8.2.1.3	Auf- und Absteigen im Dateibaum	244
8.2.2	Gebrauch von Geräte-Namen	246
8.2.2.1	Physikalische Geräte	246
8.2.2.2	Logische Geräte	250
8.2.3	Ein-/Ausgabe-Umleitung	253
9	Die Kommandos des CLI	255
9.1	Überblick über die CLI-Kommandos	255
9.2	Format der CLI-Kommandos	256
9.3	Die wichtigsten CLI-Kommandos	258
9.4	Verwendung von Mustern	306

9.5	Überblick	308
10	Der Text-Editor <i>ed</i>	313
10.1	Aufrufen und Verlassen von <i>ed</i>	314
10.2	Die direkten <i>ed</i> -Befehle	316
10.2.1	Bewegung des Cursors	317
10.2.2	Einfügen von Text	320
10.2.3	Löschen von Text	321
10.2.4	Verschieben des Text-Fensters im Text	322
10.2.5	Besondere Befehle	322
10.3	Escape-Befehle	323
10.3.1	Grundsätzliches zu Escape-Befehlen	323
10.3.2	Globale Escape-Befehle	324
10.3.3	Text-Blocks	325
10.3.4	Befehle zur Cursor-Bewegung	326
10.3.5	Suchen und Ändern von Text	326
10.3.6	Einfügen und Löschen von Text	327
10.3.7	Einige zeitsparende Tricks	328
10.4	Liste aller <i>ed</i> -Befehle	330
11	Die Automatisierung des Amiga	333
11.1	Aufgabe von Kommandofolgen	333
11.2	Eine einfache Kommandofolge	334
11.3	Kommandofolgen mit Parametern	336
11.4	Bedingte Anweisungen und Sprünge	340
12	Tips für das CLI	347
12.1	Die Startup-Sequence	347
12.1.1	Das Original	347
12.1.2	Modifikationen	348
12.1.3	Anderweitige Nutzung	349
12.2	Die RAM-Disk	349
12.2.1	Programme und Dateien in der RAM-Disk	350
12.2.2	Kombination von RAM-Disk und Startup-Sequence	351
12.2.3	Probleme mit der RAM-Disk	351
12.3	Maßschneidern einer Start-Diskette	352
12.3.1	Drucker	352
12.3.2	Zeichensätze	353
12.3.3	Kommandos	354
12.3.4	Reine CLI-Disketten	354
12.4	Dateien und Icons	355
12.4.1	Info-Dateien	355
12.4.2	Icons	355

12.4.3	IconEd	356
12.4.3.1	Starten und Verlassen von IconEd	357
12.4.3.2	Laden und Abspeichern von Icons	357
12.4.3.3	Editieren von von Icons	357
13	Chips	359
13.1	Die CPU	360
13.2	Funktionsweise der Chips	361
13.2.1	Koprozessoren	361
13.2.2	DMA	362
13.2.3	Problem und Lösungen	363
13.3	Paula	364
13.3.1	Interrupts	364
13.3.2	Ein- und Ausgabe	365
13.3.3	Klangerzeugung	365
13.4	Denise	366
13.4.1	Grafik und Farbe	367
13.4.2	Bewegung ins Bild!	367
13.5	Agnus	369
13.5.1	Was ist ein Adreß-Generator?	369
13.5.2	Der Blitter	369
13.5.3	Der Copper	371
13.6	Begriffs-Ver- und -Ent-Wirrungen	372
14	Grafik	375
14.1	Bitmapped Grafik	375
14.2	Farben	376
14.3	Mehr Farbe ins Bild!	377
14.4	Noch mehr Farbe ins Bild!	379
14.5	Der Interlace-Modus	380
14.6	Übergroße Grafiken	382
14.7	Der Dual-Playfield-Modus	382
14.8	Sprites	383
14.9	BOBs	387
14.10	AnimObjects	388
14.11	Der PAL-Amiga	389
14.12	Zusätzliche Grafik-Hardware	390
14.12.1	Das GenLock-Interface	390
14.12.2	Der FrameGrabber	391
14.13	Anwendungen der Amiga-Grafik	391
14.13.1	Fenster	392
14.13.2	Screens	394
14.13.3	Pinself	395

15	Klangerzeugung	397
15.1	Töne, wie sie der Amiga sieht	397
15.2	Die Hüllkurve eines Tons	400
15.3	Klangerzeugung mit Paula	401
15.4	Spezialeffekte	403
15.5	Sprachsynthese	403
15.6	MIDI	405
16	Die Programmierung des Amiga	407
16.1	Architektur der Amiga-Software	407
16.1.1	Exec	408
16.1.1.1	Listen	408
16.1.1.2	Tasks	409
16.1.1.3	Nachrichten	409
16.1.1.4	Speicherverwaltung	409
16.1.1.5	Bibliotheken	410
16.1.1.6	Geräte	410
16.1.2	Die Grafik-Bibliothek	411
16.1.3	Intuition	412
16.2	Grundlagen der Programmierung	413
16.2.1	Sprachen	413
16.2.2	Interpreter	414
16.2.3	Compiler	415
16.2.4	Vergleich von Compiler und Interpreter	415
16.3	Programmiersprachen für den Amiga	416
16.3.1	Assembler und Maschinencode	416
16.3.2	Basic	417
16.3.3	C	417
16.3.4	Lisp und Logo	418
16.3.5	Pascal und Modula	419
17	Amiga-Basic	421
17.1	Basic	422
17.2	Der Basic-Dialekt "Amiga-Basic"	424
17.2.1	Microsoft Basic	424
17.2.2	Moderne Features von Microsoft Basic	425
17.3	Spezielle Fähigkeiten des Amiga	426
17.3.1	Umgang mit Amiga-Basic	427
17.3.1.1	Eingabe und Editieren von Programmen	427
17.3.1.2	Fehlersuche	428
17.3.2	Unterstützung der Amiga-Fähigkeiten	429
17.3.2.1	Grafik	430

17.3.2.2	Fenster und Screens	430
17.3.2.3	Menüs	431
17.3.2.4	Die Maus	431
17.3.2.5	BOBs und Sprites	432
17.3.2.6	Abfangen von Ereignissen	432
17.3.2.7	Musik- und Sprachausgabe	433
17.3.3	Amiga-Basic ohne Grenzen	434
18	Schnittstellen	437
18.1	Überblick	437
18.2	Die parallele Schnittstelle	440
18.3	Die serielle Schnittstelle	442
18.4	Die Stereo-Anschlüsse	444
18.5	Die Video-Anschlüsse	445
18.6	Die Steuer-Schnittstellen	449
	Stichwortverzeichnis	453
	Übersicht weiterer Markt&Technik-Bücher	462

Vorwort

In vielerlei Hinsicht ist der Commodore-Amiga ein ganz besonderer Computer. Es ist schwer, die Begeisterung in Worte zu fassen, die man erlebt, wenn man die ersten Stunden und Tage damit arbeiten darf. Ich hoffe aber, daß ein wenig davon zumindest an manchen Stellen dieses Buches zu finden ist.

Merkwürdigerweise ist es aber viel einfacher, einen Menschen mit den Fähigkeiten des Amiga zu beeindrucken, wenn er schon mit anderen Mikrocomputern (und deren Beschränkungen) vertraut ist. Wenn Sie also schon andere Computer kennen, machen Sie sich auf einige Überraschungen gefaßt.

Der Amiga beeindruckt vor allem durch zwei Dinge: eine qualitativ hochwertige, vielseitige und schnelle Farbgrafik und die Möglichkeit, mehrere Programme parallel zu nutzen (*Multitasking*). Beides erscheint vielleicht auf den ersten Blick als eine Spielerei, wird aber zu einer nahezu unentbehrlichen und nur noch ungern wieder aufgegebenen Annehmlichkeit, wenn man eine Weile damit gearbeitet hat. Lassen Sie sich nicht von anders lautenden Meinungen beirren! Grafik und Multitasking sind keine grundsätzlich notwendigen Bestandteile der Computer-Bedienung — schließlich sind die Computer Jahrzehnte ohne sie ausgekommen. Sie tragen beide aber ungemein zu einer einfach zu erlernenden, schnellen und angenehmen Bedienung eines Computers bei.

Der Amiga setzt auf allen Gebieten — sowohl bei der Hardware, wie auch bei der Software — die allerneuesten Technologien ein. Für Sie als (potentiellen) Käufer bedeutet dies, daß Sie ungemein leistungsfähige Hardware zu einem vergleichsweise geringen Preis bekommen. (Obwohl dies in der Computerbranche heute nicht mehr allzu beeindruckend wirken dürfte — schon nächstes Jahr wird sicherlich ein noch leistungsfähiger Computer für noch weniger Geld auf den Markt kommen.)

Die Hauptaufgabe dieses Buches ist es aber nicht, Sie mit den phantastischen Fähigkeiten des Amiga beeindrucken. Vielmehr soll der Amiga und die Programme, die Sie brauchen, um ihn nutzbringend einzusetzen, exakt beschrieben werden. Dies soll Ihnen im täglichen Umgang mit dem Gerät und diesen Programmen helfen und Ihnen zugleich eine Entscheidungshilfe und Vergleichsmöglichkeiten mit anderen Computern geben, wenn Sie erst planen, sich (noch) einen anzuschaffen. (Direkte Vergleiche mit anderen Computern werden aber weitgehend vermieden. Sie würden zu unfair ausfallen...)

Überblick über das Buch:

Dieses Buch läßt sich grob in drei Teile gliedern, die auch manchmal im Inhalt direkt angesprochen werden. In jedem dieser Teile geht es ein wenig "tiefer" in die Materie hinein. Vielleicht ist es deshalb nicht sinnvoll, es in einem Zug durchzuarbeiten, sondern immer erst dann ein Kapitel zu lesen, wenn Sie bei Ihren eigenen Versuchen auf einen neuen Aspekt gestoßen sind, über den Sie mehr erfahren möchten. Das ist aber Geschmackssache! Wie Sie es durcharbeiten, bleibt aber letztlich Ihnen überlassen.

Im ersten Teil soll versucht werden, die wesentlichen Grundzüge der Bedienung des Amiga zu beschreiben. Sie sollten in diesem Teil mit der Philosophie des Amiga vertraut werden — seiner stark grafisch orientierten Bedienung, der Anwendung der Maus und aller wichtigen Handgriffe, die Sie später tagtäglich benötigen werden. Hierzu wird das Programm, das den Mittler zwischen Ihnen und der Hardware des Amiga spielt, die sogenannte **Workbench** (zu deutsch: *Werkbank*), ausführlich beschrieben. Am Ende des ersten Teils finden Sie dann auch Beschreibungen der ersten Anwendungsprogramme für den Amiga. Falls Sie diese Programme schon besitzen, können Ihnen ein paar Tips vielleicht helfen. Wenn Sie die Programme noch nicht kennen, können Ihnen diese Kapitel die Kaufentscheidung hoffentlich leichter machen.

Im zweiten Teil werden wir uns mit einer Bedienungsmethode beschäftigen, die nicht ganz so bildlich und intuitiv zu begreifen ist, wie die Workbench: dem sogenannten **CLI**. Diese zweite Benutzerschnittstelle des Amiga hat trotz ihrer Nachteile für den Anfänger einige Vorteile für den intensiven Anwender und diejenigen, die mit dem kleinen Nagetier namens Maus auf Kriegsfuß stehen.

Der dritte Teil des Buches schließlich enthält vor allem Informationen für die, die es immer ganz genau wissen wollen. Die **Hardware** des Amiga (und was

man damit machen kann) wird beschrieben und ein ungefährender Eindruck davon vermittelt, wie sich der Amiga dem **Programmierer** darbietet. Auch die Kontakte zur Außenwelt, Anschlüsse für Drucker und andere Zusatzgeräte, werden hier beschrieben und Tips für die entsprechenden Verbindungen gegeben.

Was dieses Buch sein will – und was nicht:

Ein einführendes Buch über einen neuen Computer stellt zwangsläufig immer ein Kompromiß dar. Es muß ganz verschiedene Leserkreise mit und ohne "computeristisches" Vorwissen, mit und ohne Interesse für die Hardware und die Programmierung und mit und ohne den Computer, um den es geht, ansprechen. Ich habe versucht, für jeden dieser potentiellen Leserkreise etwas einzubringen, mußte dafür aber zwangsläufig in Kauf nehmen, diejenigen, die etwas mehr zu einem bestimmten Aspekt wissen wollen, zu enttäuschen. Auch ein Leser, der bislang gar nichts über Computer weiß, wird in einigen Kapiteln manchmal Schwierigkeiten haben.

Für diese beiden enttäuschten Gruppen ist Rettung aber nahe. Es gibt sowohl über die Computertechnik "an sich" sehr gute einführende Werke (z.B. auch aus dem Markt&Technik-Verlag). Aber auch der an bestimmten Details interessierte Leser wird in anderen Büchern zum Amiga, die speziell auf diese Details — wie Programmierung, Hardware und Grafik — eingehen, finden können, was er sucht. Diese Bücher sind bereits erhältlich oder werden in Kürze im Markt & Technik-Verlag erscheinen.

In diesem Buch werden Sie aber einen "Rundumschlag" finden, der alle interessanten Aspekte des Amiga zunächst einmal streift und Ihnen damit vielleicht auch helfen kann, festzustellen, ob der Amiga ein Computer für Sie ist, oder nicht.¹

Manche Kapitel beschäftigen sich aber trotzdem recht detailliert mit einigen besonderen Aspekten des Amiga und sind durchaus für den täglichen Einsatz bei der Arbeit am Gerät gedacht. Ich habe dabei versucht, vor allem die Kapitel etwas mehr auszuarbeiten, die im Handbuch, das mit dem Gerät kommt, zu stiefmütterlich behandelt werden.

¹ Randbemerkungen, die wahrscheinlich nur für kleine Gruppen meiner Leser von Interesse sind, habe ich versucht, so weit wie möglich aus dem Text herauszuhalten und in Fußnoten — wie dieser — zu verlagern.

Danksagungen:

Welches Vorwort wäre komplett ohne Danksagungen? Dieses jedenfalls nicht. Ich möchte an dieser Stelle einer ganzen Reihe von Menschen danken, ohne die dieses Buch nicht zustande gekommen wäre — oder jedenfalls nicht in dieser Form.

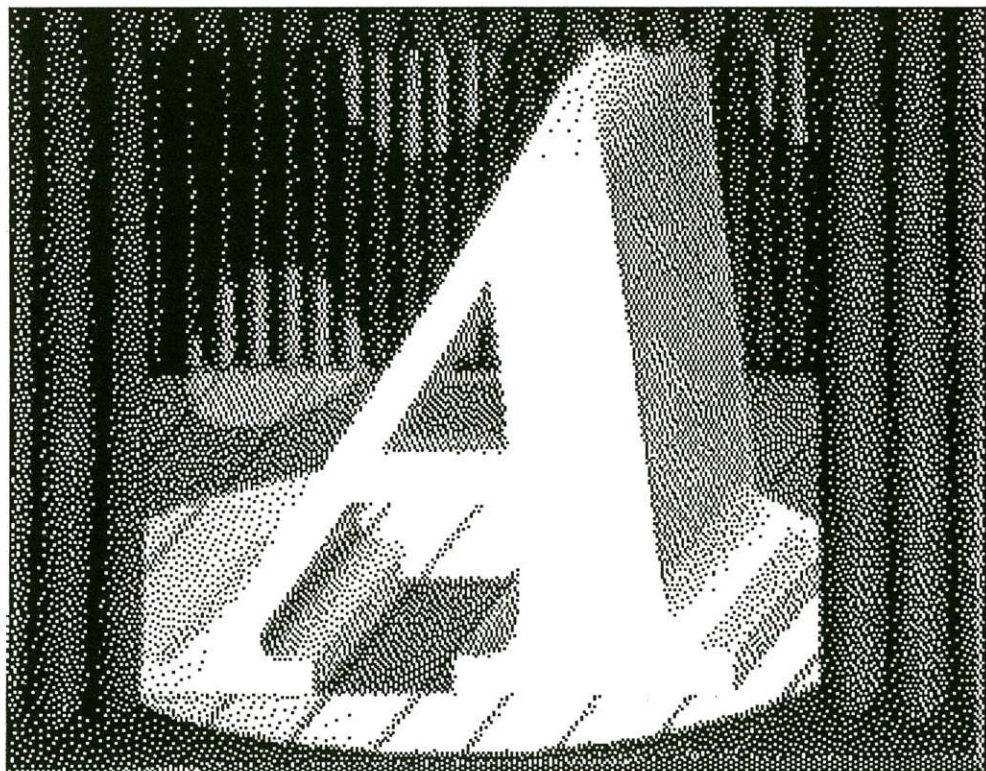
Für einen nie versiegenden Strom an Informationen und vielfältige Unterstützung beim Kampf mit den Herstellern möchte ich Christine Baumann vom *Markt&Technik*-Buchverlag danken. Ihre regelmäßigen Telefonanrufe hielten mich trotz widriger Umstände bei der Stange. Bei Horst Brandl von der *Happy*-Redaktion möchte ich für seelischen Beistand und die gute Versorgung mit wichtiger Software bedanken. Josef Steiner brachte das Manuskript in seine endgültige Form und sorgte dafür, daß Sie nun ein wirklich schönes und (hoffentlich weitgehend) fehlerfreies Buch lesen werden. Manfred Kohlen schulde ich besonderen Dank. Er trug große Teile zum letzten Buchteil und vor allem viele der schönen und informativen Zeichnungen in diesem Teil bei. Ohne ihn wäre dieses Buch (und meine Nächte) kürzer und inhaltsärmer geworden.

Und *last but not least* darf auch ein Wort des Dankes für viel Geduld und Verständnis an meine Freundin Maria Berhorst nicht fehlen. Sie dürfte nun wissen, was das Wort "Computer-Witwe" bedeutet!

Dortmund, im Februar 1986,

markus breuer

1 Vorhang auf: Der Amiga !



Der Commodore Amiga ist einer der ersten Vertreter einer neuen Computer-Generation. Diese neue Generation legt wesentlich mehr Wert als ihre Vorgänger auf die einfache Bedienung des Computers als auf die "nackten

Daten". Man könnte einen Vergleich mit Autos ziehen und sagen, daß die Polsterung der Sitze, die Qualität der Stereo-Anlage und auch die Bremsen an erster Stelle kommen und irgendwann viel später erst die PS- (alias kW-) Leistung des Motors. Fairerweise muß man allerdings zugeben, daß eine solche Umorientierung bei den Computern erst ab dem Moment möglich war, wo nach Erledigung der notwendigen Rechen- und Steuerungs-Aufgaben noch genügend Computerleistung "übrig" blieb, die sich um solche Komfort-Aspekte kümmern kann.

Deshalb darf man auch die "nackten Daten" eines solchen Computers nicht vergessen, denn erst sie machen den Komfort möglich. In diesem Kapitel soll deshalb zunächst ein kleiner Überblick über die Leistungsdaten des Amiga und eher "äußerliche Aspekte" gegeben werden. Dies beginnt zunächst mit der Hardware und dann mit der Software, die Sie erhalten, wenn Sie einen Amiga kaufen. Dieser Überblick führt Sie praktisch im Laufschrift durch die Welt des Amiga. Er wird Ihnen zwangsläufig etwas oberflächlich vorkommen. Aber keine Angst: Fast alle angeschnittenen Themen werden in den folgenden Kapiteln noch im Detail erläutert.

Das Wort "Amiga" ist spanisch und bedeutet ungefähr soviel wie "Freundin". Genau das sollte dieser Computer im Idealfall auch für Sie werden (natürlich nur im platonischen Sinn). Aber wie auch im wirklichen Leben, kann Ihnen diese neue Freundin viel Ärger bereiten. Betrachten wir aber zunächst einmal ganz unvoreingenommen ihr äußeres Erscheinungsbild. Man soll ja nicht zuviel auf Äußerlichkeiten geben; vielleicht ist es aber trotzdem ganz interessant für Sie.

Wenn Sie bereits längere Zeit einen Amiga besitzen, können Sie dabei wahrscheinlich wenig Neues lernen. Sie sollten die folgenden Abschnitte trotzdem überfliegen, weil einige Begriffe eingeführt werden, die auch im restlichen Buch immer wieder gebraucht werden.

1.1 Äußerlichkeiten

Äußerlich macht der Amiga ("Amiga 1000" um genau zu sein; Commodore/Amiga hat offensichtlich noch weitere Modelle in Planung) einen recht attraktiven Eindruck. Das "Basispaket", das natürlich ausgebaut werden kann, besteht aus insgesamt 4 Komponenten:

- Einer Haupteinheit mit eingebautem Diskettenlaufwerk
- Einer separaten Tastatur
- Einem Farbmonitor
- Einer Maus

Optional kann man (und sollte auch!) noch ein zusätzliches Diskettenlaufwerk erwerben, das außen an die Haupteinheit angeschlossen wird.¹



Bild 1 - 1: *Der Amiga*

Die einzelnen Komponenten sind über Kabel miteinander verbunden. An einem langen und flexiblen Spiralkabel ist z.B. die Tastatur mit der Haupteinheit verbunden. Die Maus ist über ein ebenfalls langes Kabel an der rechten Gehäuseseite der Haupteinheit angeschlossen. Der Bildschirm steht locker und frei beweglich auf der Haupteinheit und ist ebenfalls über ein

¹ Falls Sie sich schon einmal mit Computern beschäftigt haben, werden Ihnen diese Begriffe bestimmt bekannt vorkommen. Falls nicht, so sollten Sie zu einem einführenden Buch über Computer greifen, von denen es (auch aus dem M&T-Verlag) einige auf dem Markt gibt.

Kabel hinten an diese angeschlossen. Das zusätzliche Diskettenlaufwerk hat ein etwas kurzes und starres Kabel und kann rechts neben die Haupteinheit gestellt werden. Es paßt aber auch sehr gut hochkant rechts neben den Bildschirm auf die Haupteinheit.

Die Länge der verschiedenen Kabel läßt eine fast beliebige Positionierung der Komponenten zu. Ob Sie die Maus z.B. lieber rechts oder links neben der Tastatur liegen haben wollen, bleibt Ihnen überlassen — dicht neben der Tastatur sollte sie aber auf alle Fälle liegen (mehr dazu weiter unten).

Die (englische) Tastatur ist angenehm flach und enthält 89 Tasten (die deutsche Tastatur, die uns zum Test noch nicht vorlag, wird noch ein paar Tasten mehr enthalten, die vor allem die Umlaute aufnehmen werden). Sie enthält neben dem normalen Satz Schreibmaschinen-Tasten einen separaten Zehnerblock für die schnelle und komfortable Zahleneingabe, einen kreuzförmigen Block mit Tasten zur Steuerung einer Schreibmarke (Cursor) oder eines Zeigers und eine Reihe von 10 Funktionstasten am oberen Rand. Mit diesen Funktionstasten können Sie innerhalb von Programmen oft bestimmte wichtige Funktionen mit nur einem Tastendruck auslösen.

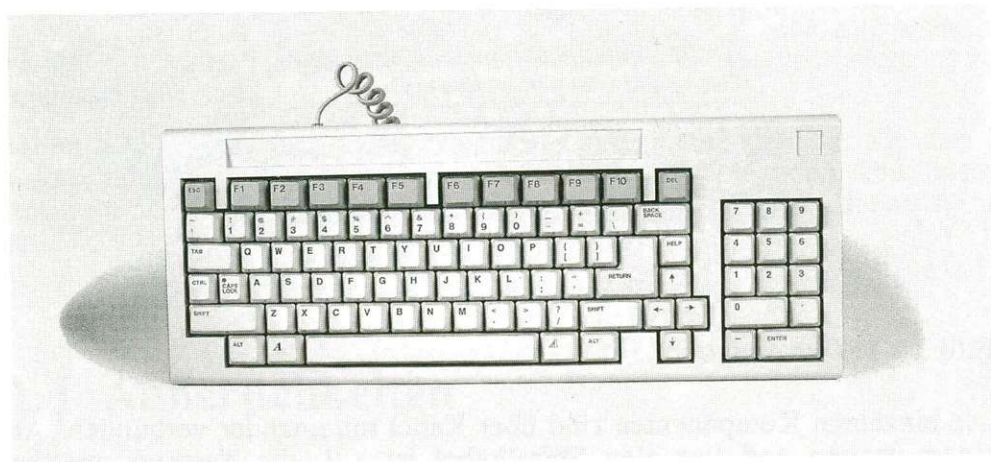


Bild 1 - 2: Die Tastatur des Amiga

Links und rechts neben der Leertaste liegen zwei besondere Tasten, die beide verschiedene Formen des Amiga-Markenzeichens **A** tragen. Es sind (in Kombination mit anderen Tasten) spezielle Funktionstasten, wie sie heute

auch bei vielen anderen Rechnern zu finden sind (z.B. Apple II, Apple Macintosh und Atari ST).

Direkt neben den Amiga-Tasten liegt auf beiden Seiten noch eine sog. "Alt-Taste". Hält man die Alt-Taste fest und drückt dann einen "normalen" Buchstaben, so erhält man besondere Zeichen, wie z.B. á, œ, Σ und ¿. Fast jede Buchstabentaste ist mit einem solchen alternativen Symbol versehen, weshalb man den Amiga wohl mit Recht als einen internationalen Computer bezeichnen kann. Fast jeder Buchstabe, den man sich vorstellen kann, steht mit der Alt-Taste zu Verfügung.

Mit zwei kleinen Beinchen am hinteren Rand kann die Tastatur bei Bedarf schräg gestellt werden, was viele Benutzer beim Tippen als angenehmer empfinden. Der Anschlag der Tasten ist leicht und recht angenehm sowohl für diejenigen, die nach dem Adler-System (Ziel suchen und dann hinabstoßen) die Buchstaben finden, als auch für die 10-Finger-Tipper. Der Tastenhub ist weder extrem lang noch besonders gering. Ein Buchstabe gilt allerdings schon als getippt, bevor man beim Anschlag ankommt. Beim Tippen jeder Taste hört man sanftes Klicken als Rückkopplung, daß der Buchstabe vom Rechner erkannt wurde. Dieses Klicken kommt nicht aus dem Lautsprecher, wie bei einigen anderen Rechnern üblich, sondern ist mechanischen Ursprungs.

Das Gesamtsystem erinnert ein wenig an einen IBM PC oder einen der vielen Nachbauten, von denen Sie ja bestimmt schon einmal irgendwo einen gesehen haben. Die Haupteinheit wirkt etwas zierlicher und ist auch leichter als das entsprechende IBM-Produkt. Sie hat zudem noch den entscheidenden Vorteil, daß sie die Tastatur aufnehmen kann, wenn das Gerät ausgeschaltet ist. Unten im Gehäuse der Haupteinheit ist dafür ein Platz ausgespart worden, in dem dann auch das Spiralkabel der Tastatur völlig verschwindet. Ist die Tastatur erst einmal in dieses Fach geschoben, so benötigt der Amiga angenehm wenig Platz auf dem Schreibtisch. Er macht dann auch einen wesentlich "aufgeräumteren" Eindruck.

Die mechanische Maus des Amiga besitzt eine elegante "Stromlinienform" und liegt gut in der Hand. Sie trägt an ihrer Unterseite eine kleine Gummikugel, die sich dreht, wenn Sie die Maus über eine glatte Oberfläche schieben. Diese Drehung wird über das angeschlossene Kabel dem Computer mitgeteilt und kann für Steuerungszwecke am Bildschirm verwendet werden. (Mehr darüber im nächsten Kapitel.) Die Kugel (und damit die Maus) rollt sehr flüssig, und falls Sie das einmal nicht mehr tut, ist sie mit Sicherheit

verschmutzt und sollte gereinigt werden.² An der Oberseite des Maus befinden sich zwei Knöpfe. Ein Drücken dieser Knöpfe wirkt in bestimmten Situationen als Befehl an den Computer.

Das eingebaute Diskettenlaufwerk des Amiga hat (genauso wie das zusätzliche) eine formatierte Kapazität von 880 KByte, was ziemlich viel für ein Diskettenlaufwerk ist.³ Die verwendeten Disketten mit 3,5 Zoll Durchmesser und Hartschale sind handlich, robust und zuverlässig. Sie setzen sich jetzt allmählich als Standard bei neueren Mikrocomputern durch.

Neben einem (oder auch mehreren) zusätzlichen 3,5-Zoll-Laufwerk(en) können auch 5,25-Zoll-Laufwerke angeschlossen werden, wie sie bis heute von vielen anderen Herstellern verwendet werden. Mit der entsprechenden Software wird der Amiga so in die Lage versetzt, auch Daten von Disketten zu lesen, die von anderen Computern, die 3,5-Zoll oder 5,25-Zoll-Laufwerke verwenden (z.B. ein IBM PC), darauf geschrieben wurden.

Der einzige Nachteil, den die Diskettenlaufwerke des Amiga vielleicht haben, ist ihre Lautstärke. Sie sind verhältnismäßig laut, was eindeutig auf die Art zurückzuführen ist, in der sie eingebaut wurden (Resonanz). Es wäre schön gewesen, wenn sich Commodore/Amiga da etwas mehr Mühe gegeben hätte.

Als Monitor (Bildschirm) für den Amiga können Sie ein spezielles Gerät verwenden, von dem Commodore/Amiga zwei verschiedene Formen anbietet, einen Video-Monitor oder auch ein Farbfernsehgerät. Im Moment können Sie die billigste Lösung (den Farbfernseher) noch nicht verwenden, da der Amiga bis jetzt nur einen NTSC-Ausgang (nach amerikanischer Farbfernsehnorm) besitzt. Aus den USA aber war zu hören, daß der Amiga an einem Farbfernseher ein erstaunlich gutes Bild macht und 60spaltiger Text sogar gut lesbar sein soll. Bei der endgültigen deutschen Version des Amiga wird der entsprechende Ausgang der PAL-Norm entsprechen und somit auch zu handelsüblichen Fernsehern passen. Diese endgültige Version wird aber erst ab Sommer 1986 vorliegen. Prototypen der PAL-Version haben wir aber bereits an einem PAL-Monitor gesehen und müssen sagen, daß sie wirklich ein erstaunlich gutes Bild liefern.

² Diese Reinigung können Sie im Gegensatz zu den meisten Wartungs -aufgaben an Ihrem Amiga selbst durchführen. Wie sie genau gemacht wird, erfahren Sie im Amiga-Benutzerhandbuch. Man kann aber auch nur sehr schwer etwas dabei falsch machen. Sie sollten also nicht vor häufiger Reinigung zurückschrecken.

³ 880 KByte sind mehr als 900 000 Buchstaben. Falls Sie nur Texte auf einer solchen Diskette speichern, so passen etwa 400 eineinhalbzeilig getippte Manuskriptseiten darauf.

Das beste Ergebnis (und die wenigsten Kopfschmerzen) werden Sie aber zweifellos erzielen, wenn Sie einen der speziellen Amiga-Monitore oder einen anderen guten RGB-Monitor verwenden.⁴ Der Amiga 1080 ist ein solcher analoger RGB-Monitor, der die volle Palette von 4096 Farben, zu der der Amiga fähig ist, darstellen kann — bei sehr guter Schärfe übrigens. Neben seinen grafischen Fähigkeiten kann er aber auch 80-spaltigen Text in akzeptabler Qualität zeigen. Das Bild ist stabil, flimmerfrei, und die Farbtrennung ist bis in die Ecken hervorragend (keine *Regenbogen*). Dieser Monitor und der Amiga stellen wohl die ideale Kombination dar. Der preiswertere Monitor 1070 wurde von uns nicht getestet. Wer bereits einen IBM-kompatiblen RGB-Monitor besitzt, von denen es eine große Auswahl gibt, kann sich eine Ausgabe sparen und diesen anschließen. Dann muß man allerdings mit 16 Farben auskommen (mehr schafft die herkömmliche Standard-IBM-Grafik nämlich nicht).

1.2 Innere Werte

Die Haupteinheit enthält unter anderem die folgenden Komponenten:

- Die CPU M 68 000
- 256 KByte RAM-Speicher für Programme und Daten
- 256 KByte spezieller KickStart-RAM-Speicher (s.u.)
- 3 spezielle Chips für Sound und Grafik
- Ein 3,5-Zoll-Diskettenlaufwerk mit 880 KByte Kapazität
- Diverse Zusatz-Chips für Hilfsaufgaben
- Ein Netzteil, das alles mit Strom versorgt

Für Fachleute: Bei der CPU handelt es sich um den inzwischen recht weit verbreiteten Motorola-Prozessor 68 000, der auch im Atari ST, dem Apple Macintosh und anderen leistungsfähigen PCs zu finden ist. Die CPU und der Speicher werden mit einer Taktfrequenz von etwas unter 14,5 MHz versorgt. Die CPU selbst bekommt davon aber nur jeden zweiten Takt ab, so daß Sie

⁴ Ein RGB- (Rot-Grün-Blau-) Monitor erhält die einzelnen Signale für die drei Grundfarben, aus denen sich die Farbe eines Bildschirmpunktes zusammensetzt, über drei getrennte Kabel. Die Bildqualität, besonders die Farbtrennung und Schärfe ist deshalb typischerweise wesentlich besser, als bei einem sogenannten Composit-Monitor oder gar einem einfachen Farbfernseher.

effektiv nur mit der Hälfte dieser Taktfrequenz (ziemlich exakt 7,19 MHz) läuft. Die anderen Takte sind für die drei Spezial-Chips reserviert.

Mit den 256 KByte speziellem RAM-Speicher in der Haupteinheit des Amiga hat es eine besondere Bewandnis. Wie die meisten 68 000-Computer wurde der Amiga zunächst mit einem großen ROM entworfen (Man muß dabei z.B. auch daran denken, daß der Amiga einmal als Homecomputer gedacht war, der auch ohne Diskettenlaufwerk funktionieren sollte). Dieser ROM sollte alle wesentlichen Teile des Betriebssystems, der Grafik- und Sound-Software sowie der Hilfsroutinen enthalten, die von den späteren Programmen genutzt werden können.

Wie scheinbar allmählich üblich, gelang es den Software-Ingenieuren aber nicht rechtzeitig zur Fertigstellung der Hardware, auch die Software in einem endgültigen Zustand fertigzustellen. Man hat deshalb einfach vorläufig (?) die ROMs aus den Design herausgenommen und zusätzlichen RAM-Speicher an dessen Stelle eingebaut. Dieser Speicherbereich hat eine Größe von 256 KByte und ersetzt den geplanten ROM, der eine Größe von 192 KByte haben sollte. Beim Einschalten des Amiga muß er von der KickStart-Diskette mit der vorläufigen Version der Software gefüllt werden, die eigentlich in den ROM gehen sollte. Nachdem er einmal gefüllt ist, kann er nur noch gelesen aber nicht mehr überschrieben werden (*Write Once Memory* oder kurz *WOM*).⁵ Da sich der WOM funktional äquivalent zu einem ROM verhält, wird in der Amiga-Dokumentation allerdings manchmal von ROM gesprochen, wo WOM gemeint ist.

Dieser Ansatz hat Vor- und Nachteile, obwohl Commodore im Moment verständlicherweise die Vorteile betont. Der Hauptvorteil des Kickstart/WOM-Prinzips liegt in der Flexibilität des Ansatzes. Neuere Versionen der ROM-Software können ohne Schwierigkeiten, einfach in Form einer Diskette, an die Kunden ausgeliefert werden. Auch gänzlich andere Betriebssysteme können realisiert werden, ohne dafür Teile des stets kostbaren RAMs "verschwenden" zu müssen. Diese Vorteile betreffen aber hauptsächlich den Hersteller bzw. die Softwarehäuser und machen ihnen die Arbeit einfacher. Für den Benutzer des Amiga bedeutet der Kickstart-Prozß in erster Linie einen zusätzlichen Arbeitsvorgang beim Einschalten des Geräts und natürlich zusätzliche Wartezeit. Nachdem die Kickstart-Software gelesen wurde, muß die Kickstart-Diskette nämlich ausgeworfen werden und die

⁵ Amiga selbst nennt diesen Bereich auch *Writeable Control Store*, beschreibbaren Kontroll-Speicher, oder kurz *WCS*. Dieser Name wird in der Computertechnik aber eigentlich schon für andere Dinge reserviert. Wir nennen diesen Speicher deshalb *WOM*.

eigentliche Start-Diskette eingelegt werden, die die "äußeren Teile" des Betriebssystems und die Programme, mit denen man arbeiten will, enthält.

1.3 Verbindungen

Der Amiga ist für die Kommunikation mit der Außenwelt — nicht nur mit dem Benutzer — gut gerüstet. An seiner Haupteinheit befinden sich reichlich Anschlüsse (sog. *Schnittstellen*), die für's erste den meisten Anwendern völlig genügen dürften.

Hinten an der Haupteinheit befinden sich unter anderem:

- Ein zusätzlicher Diskettenanschluß
- Der Anschluß für die Tastatur
- Eine serielle Schnittstelle (nach RS232-Standard)
- Eine parallele Schnittstelle (nach Centronics-Standard)
- Diverse Anschlüsse für die verschiedensten Monitore
- 2 Lautsprecheraustritte für Stereo-Betrieb

Die wichtigsten dieser Anschlüsse dürften für Sie wohl die parallele Schnittstelle sein, an die Sie eine Vielzahl handelsüblicher Drucker anschließen können, dann die serielle Schnittstelle, an die Sie ebenfalls Drucker oder z.B. ein MODEM (für die Kommunikation mit anderen Computern über Telefon-Leitungen) anschließen können; der Monitor-Ausgang und vielleicht auch der zweite Diskettenanschluß sind ja bei Ihnen wahrscheinlich schon belegt. Damit ist die Aufzählung der Verbindungen des Amiga zur Außenwelt aber noch nicht abgeschlossen.

An der rechten Seite der Amiga-Haupteinheit befinden sich zwei identische Steckbuchsen. Die vordere davon dürfte bei Ihnen bereits durch die Maus belegt sein. Mit der hinteren können Steuerknüppel (Joysticks) oder ähnliche Geräte an den Amiga angeschlossen werden. Vielleicht kommt ein findiger Ingenieur aber auch noch auf ganz neue Verwendungseinfälle dafür.

Ebenfalls an der rechten Seite findet sich, versteckt hinter einer herausnehmbaren Plastikleiste, die Erweiterungsschnittstelle. Hier werden praktisch alle Signale, die intern im Amiga erzeugt werden (für Fachleute: der komplette Systembus), nach außen gelegt. Hersteller von Zusatzgeräten, aber auch Amiga selbst, bieten bereits eine Reihe von Erweiterungen an, die Sie an dieser Stelle anschließen können, oder planen solche Erweiterungen. Zu diesen Erweiterungen gehören unter anderem mehr Speicher, eine Anschluß -

möglichkeit für Videogeräte und Bildplattenspieler sowie ein Festplatten - anschluß.⁶

An der Vorderseite der Haupteinheit schließlich findet sich, ebenfalls unter einer Plastikabdeckung verborgen, eine Steckerleiste, die speziell für Speichererweiterungen gedacht ist. Amiga selbst bietet eine Speicher - erweiterung an, die den Grundausbau von 256 KByte auf 512 KByte aufstockt. Mittelfristig ist aber damit zu rechnen, daß von anderen Herstellern bald auch Erweiterungen für diesen Stecker angeboten werden, die noch mehr bieten.

1.4 Was man außer der Hardware bekommt

Neben der reinen Hardware bekommt man schon beim Kauf des Basispakets eine ganze Menge Software dazu. In der amerikanischen Version sind es im Moment die folgenden Disketten (es ist noch nicht ganz klar, welche davon bis zur deutschen Premiere "eingedeutscht" sein werden):

- Die KickStart-Diskette
- Die Workbench-Diskette
- Die Extras-Diskette
- Die Polyscope-Diskette

Die **KickStart**-Diskette wird bei jedem Einschalten des Amiga nur einmal benötigt und enthält den "ROM"-Inhalt des Amiga.

Die **Workbench**-Diskette ist dann die eigentliche "Boot-Diskette" mit der der Amiga nach dem Start "hochgefahren" wird, wie die Computerleute sagen. Sie sorgt für das Erscheinen der grafischen Benutzerschnittstelle des Amiga, der *Workbench* oder zu deutsch *Werkbank*, mit der der Benutzer wohl die meiste Zeit zu tun haben wird. Sie enthält ansonsten nur einige kleine Hilfs- und Demonstrations-Programme, mit denen man die Möglichkeiten des Amiga etwas austesten kann.

Die **Extras**-Diskette enthält ein Tutorial, mit dessen Hilfe sich der Amiga selbst in Bild und Ton vorstellt. Dieses Tutorial macht einen frischgebackenen

⁶ Festplatten sind ein Speichermedium, das sowohl zuverlässiger als auch schneller ist als Disketten und meist eine wesentlich höhere Kapazität besitzt. Wenn Sie Ihren Amiga professionell nutzen wollen, werden Sie auf Dauer nicht um die Anschaffung einer Festplatte herumkommen.

Amiga-Benutzer mit den wesentlichen Grundzügen der Bedienung seines neuen Prachtstücks vertraut und zeigt zudem noch einige der Möglichkeiten, die die Maschine grafisch und akustisch bietet. Neben diesem Tutorial befindet sich noch das Amiga-Basic auf der Extras-Diskette. Amiga-Basic ist eine sehr komfortable Version des bekannten und auf eigentlich allen anderen Mikrocomputern zu findenden Microsoft-Basic, der am weitesten verbreiteten Computer-Software überhaupt. Mehr dazu aber in einem späteren Kapitel.

Die **Polyscope**-Diskette schließlich enthält zwei Programme des amerikanischen Softwarehauses "Electronic Arts". Das eine Programm zeigt Demos der Softwareprodukte, die dieses Softwarehaus im Moment schon für den Amiga anbietet oder in Planung hat. Das zweite Programm "Polyscope" ist eine reine Farborgie, die zeigt, was der Amiga bezüglich Farbe ohne große Schwierigkeiten zu leisten imstande ist. Wenn Sie ein Faible für psychodelische Farbspielereien haben, werden Sie dieses Software-Pröbchen bestimmt lieben.

Neben diesen drei Disketten können Händler noch ein oder zwei Disketten mit zusätzlicher Demo-Software bekommen. Die meisten Händler werden wohl auch nichts dagegen haben, ihren Kunden diese zu kopieren. Diese Demos zeigen vor allem, "was im Amiga steckt". Es sind viele phantastische Bilder, schnelle bewegte Grafiken, ein einfaches elektronisches Klavier und ähnliches.

Zu den Demos gehört natürlich auch der legendäre *bouncing ball*, ein rotierender, auf und ab hüpfender, rot/weiß-karierter Globus, der immer, wenn er vom Boden oder vom Rand abprallt, in Stereosound Aufschlagsgeräusche von sich gibt.

Mit dieser Grundausstattung werden Sie sich als frischgebackener Amiga-Besitzer aber wohl höchstens ein oder zwei Tage beschäftigen können, falls Sie nicht Amiga-Basic benutzen wollen, um eigene Programme zu entwickeln.

Für den Anwender (*Anwender* hier im Gegensatz zum Programmierer) gibt es bereits einige zusätzliche Software — wenn Sie dieses Buch lesen, wahrscheinlich noch einige mehr. Von Commodore/Amiga selbst können Sie z.B. die Textverarbeitung TextCraft und das Mal-Programm GrafiCraft erhalten. Auch das Softwarehaus Electronic Arts, von dem einige der Demos stammen, hat bereits einige Software im Angebot. Mehr dazu aber in einem folgenden Kapitel.

Außer den Disketten bekommen Sie natürlich auch noch Handbücher für die Bedienung Ihres neuen Computers, und zwar zwei Stück. Eines behandelt den Amiga (das Gerät) selbst und die Workbench (s.u.). Dieses Handbuch ist

sehr kurz und knapp gefaßt. Das zweite Handbuch behandelt Amiga-Basic und die wichtigsten Grundzüge der Programmierung des Amiga.

1.5 Fähigkeiten des Amiga

Wie wollen Ihnen nun auch noch kurz einen Eindruck davon geben, wozu der Amiga fähig ist, wenn er die richtige Software hat. Diese Daten sind sehr beeindruckend, besonders, wenn Sie bereits andere Computer kennen, die zumeist nicht so leistungsfähig sind. Sie sollten sich davon aber nicht blenden lassen. Das wichtigste bei jedem Computer ist doch, daß er Ihnen Arbeit abnimmt. Der Amiga hat mit seinen Grafik- und Sound-Fähigkeiten dazu aber noch das Potential, ihnen diese Arbeitserleichterung so angenehm und unterhaltsam wie möglich zu gestalten.

1.5.1 Grafik-Fähigkeiten

Der eine Aspekt, der am Amiga von allen Journalisten, die ihn natürlich zuerst sahen, nahezu uneingeschränkt gelobt wurde, sind seine Grafikfähigkeiten. Sie sind den Möglichkeiten vergleichbarer Computer in vieler Hinsicht weit voraus. Drei speziell für diesen Zweck entworfene und für Commodore/Amiga gebaute Chips (sog. *Custom VLSI-Chips*) sorgen für eine Vielfalt an Möglichkeiten und vor allem für eine Geschwindigkeit, wie man sie bisher — zumindest bei Computern in dieser Preislage — für unmöglich hielt.

Der Grafik des Amiga ist *bitmapped*. Dies bedeutet, daß die Farbe jedes Punktes auf dem Bildschirm durch ein oder mehrere Bits im Speicher bestimmt wird. Im simpelsten Fall gibt es eine 1:1-Zuordnung, bei der jeder Punkt auf dem Bildschirm genau einem Bit im Speicher entspricht. Da der Amiga mehrere Farben gleichzeitig in einer Grafik verwenden kann, liegt der Fall natürlich etwas komplizierter — aber im wesentlichen stimmt die 1:1-Zuordnung, von der auch der Begriff *bitmapped* abgeleitet ist

Andere Computer haben manchmal noch einen zusätzlichen Betriebsmodus, in dem sie nur Text (Buchstaben, Zahlen und Sonderzeichen) auf dem Bildschirm darstellen kann. Beim Amiga gibt es einen solchen Modus nicht. Auch Text muß, wie ein Kreis oder ein Rechteck, als ein Punktmuster auf den Bildschirm gemalt werden. Dies ist natürlich schwieriger, aber dafür auch

flexibler. Durch die Geschwindigkeit, die die speziellen Grafik-Chips dem Amiga verschaffen, spielt der zusätzliche Aufwand zudem kaum eine Rolle.

Die Amiga-Grafik kennt allerdings auch mehrere Modi. Die horizontale Auflösung⁷ kann 320 oder 620 Punkte betragen. Die vertikale Auflösung kann entweder 200 oder 400 Punkte betragen.⁸ Die vertikale 400-Punkte-Auflösung wird allerdings auf eine Art und Weise erreicht, die zu erhöhtem Bildschirm-Flimmern führt. Diese Methode nennt man *interlaced Video*, sie wird auch bei Fernsehern angewandt, weshalb man nicht allzu lange nahe vor einem Fernseher sitzen kann, ohne Kopfschmerzen zu bekommen.

Die bitmapped Grafik des Amiga ist mehrfarbig. Je nachdem, wieviel Speicher man opfern will, kann man zwischen 2, 4, 8, 16 und 32 möglichen Farben wählen, die gleichzeitig in einer Grafik erscheinen können. Bei mehr als 2 Farben wird auch mehr als 1 Bit pro Punkt benötigt. Deshalb auch der erhöhte Gesamtspeicherbedarf für ein Bild mit mehr Farben. Diese 32 Farben sind aber nicht fest "eingebaut", sondern können aus einer Palette von 4096 Farben ausgewählt werden. Man sucht sich also "normalerweise" zunächst aus einer Palette von 4096 Farben 32 aus und kann mit diesen dann sein Bild malen.

Das Wort "normalerweise" im letzten Satz haben wir deshalb in Anführungszeichen gesetzt, weil es einen Betriebsmodus gibt, in dem — mit gewissen Einschränkungen allerdings — alle 4096 möglichen Farben gleichzeitig am Bildschirm erscheinen können. Dies ist der sog. *hold and modify* oder kurz HAM-Modus, über den Sie in einem folgenden Kapitel noch mehr erfahren werden.

Wieviele Bits für einen Punkt "zuständig" sind, und damit, wieviele Farben zur Verfügung stehen, wie hoch die Auflösung des Bildschirms ist, und wie die Bits eines Punktes interpretiert werden, kann der Programmierer nach den Bedürfnissen seines Programms frei bestimmen. Tabelle 1-1 enthält eine Zusammenfassung der möglichen Grafikmodi.

⁷ *Auflösung* nennt man die Anzahl der Punkte, die einzeln angesteuert bzw. gefärbt werden können. Je höher diese Anzahl ist, desto höher die Auflösung.

⁸ Die europäische PAL-Version wird sogar die Möglichkeit bieten auch 256 oder 512 Punkte vertikale Auflösung zu wählen.

Grafikauflösung	maximal mögl. Farbanzahl
-----------------	--------------------------

320 (H) * 200 (V)	32
640 (H) * 200 (V)	16
320 (H) * 400 (V) (interlaced)	32
640 (H) * 400 (V) (interlaced)	16
320 (H) * 200 (V) (HAM)	4096
320 (H) * 400 (V) (HAM,interl.)	4096

Tabelle 1 - 1: Grafik-Modi des Amiga

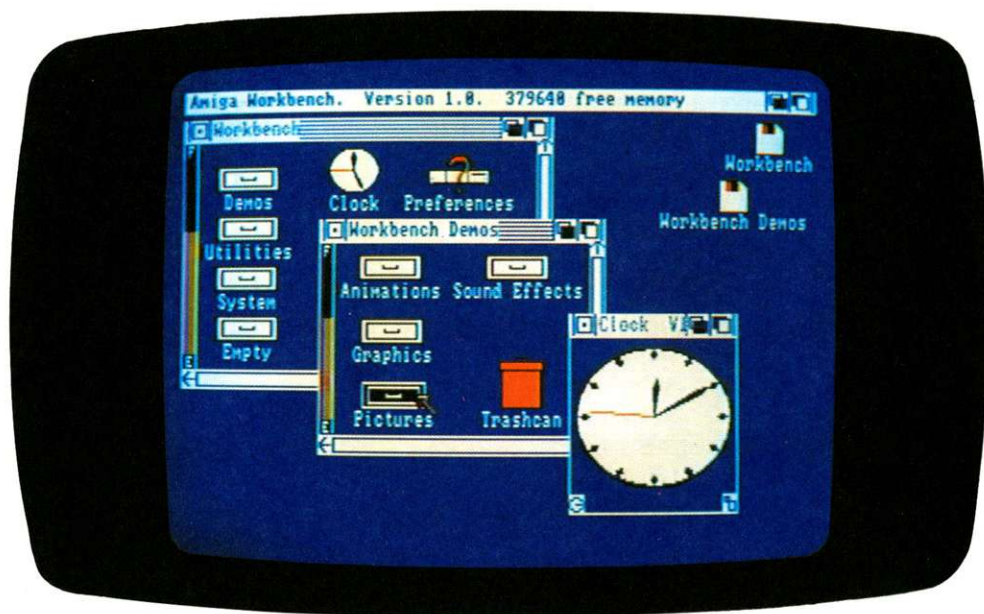
Zusätzlich ist es möglich, den Bildschirm in horizontale Streifen zu unterteilen, in denen verschiedene Modi gelten. Z.B. kann die obere Hälfte oder das obere Drittel des Bildschirms andere 32 Farben verwenden als der untere Teil. Wie Sie sehen, sind die Grafikmöglichkeiten des Amiga sehr vielseitig.

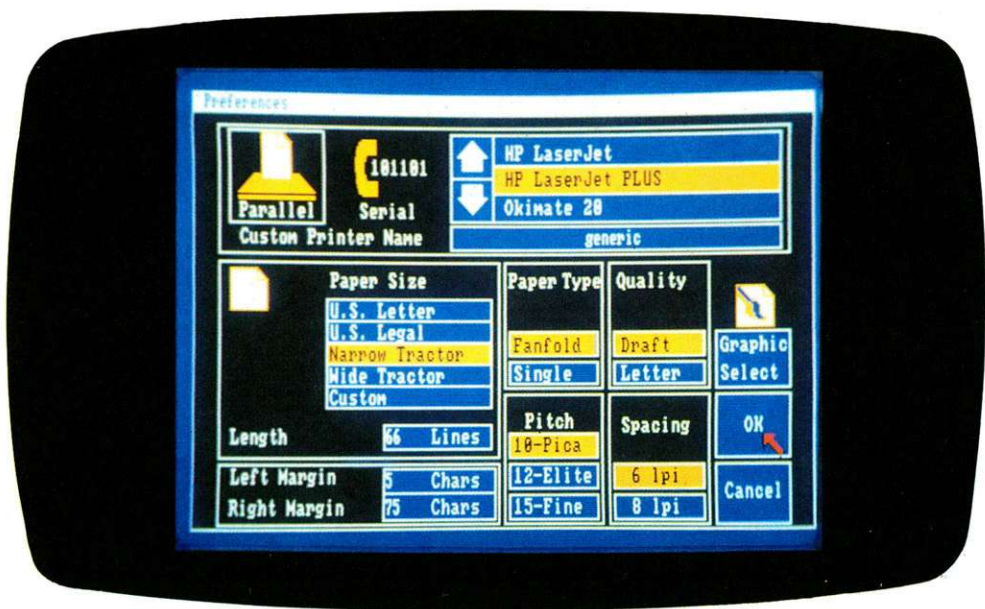
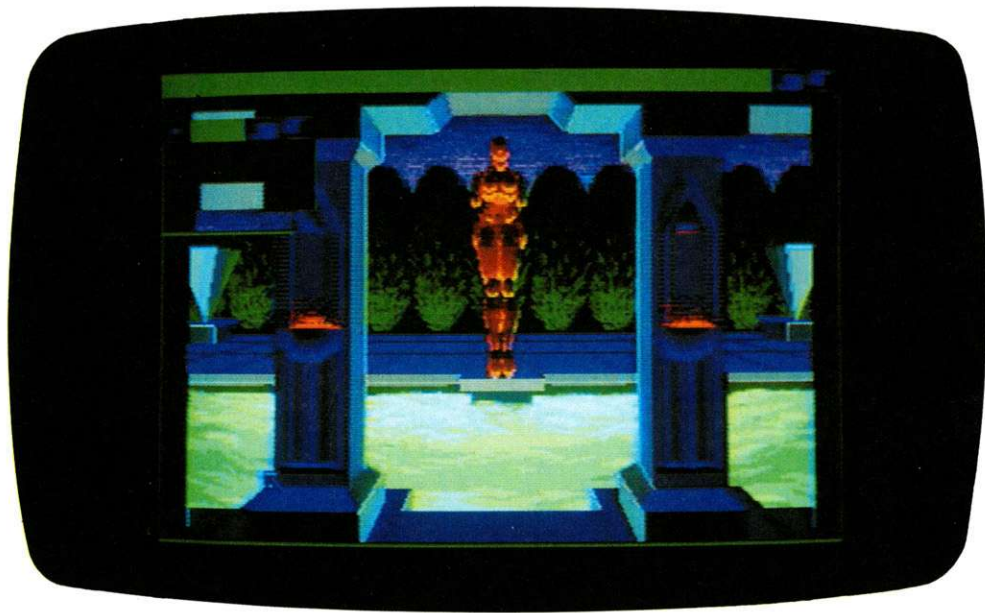
Bisher haben wir uns nur mit den statischen Aspekten der Amiga-Grafik beschäftigt, also ruhenden, unveränderlichen Bildern. Die Grafik-Chips des Amiga bieten zudem aber die Möglichkeit, 8 kleine Bilder (die sogenannten Sprites) mit hoher Geschwindigkeit über den Bildschirm zu bewegen. Dies ist nicht nur für Videospiele wichtig, wie Sie vielleicht meinen werden.

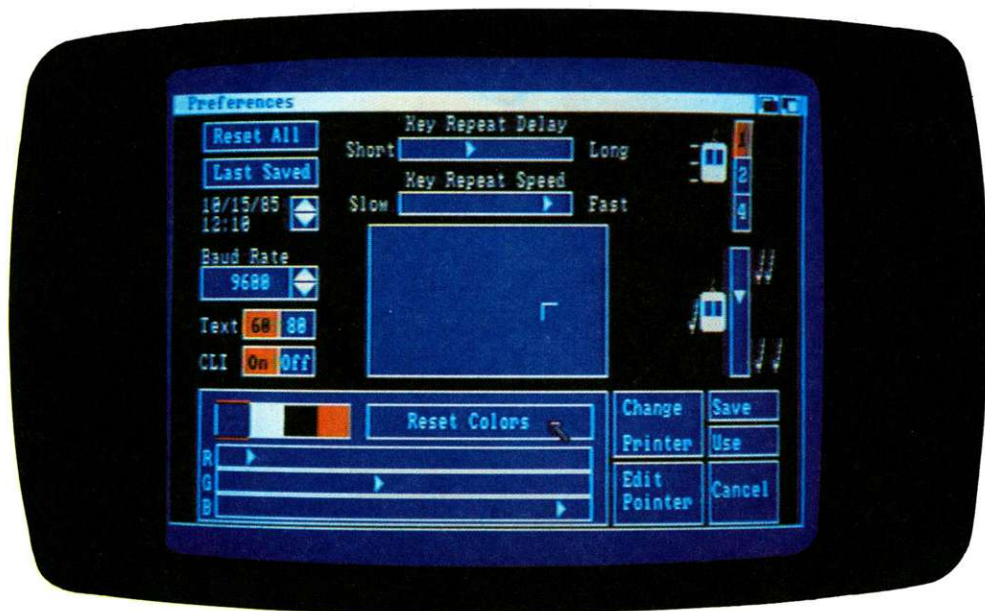
Die Software des Amiga bietet darüberhinaus aber in Zusammenarbeit mit anderen Teilen der Grafik-Chips noch weitergehende Möglichkeiten für solche bewegte (der Fachmann spricht von *animierten*) Grafiken, die dem Programmierer sehr viel Arbeit abnehmen. Hierzu gehören die **BOBs** (Blitter Objects) und **AnimObjects** (animierte Objekte).

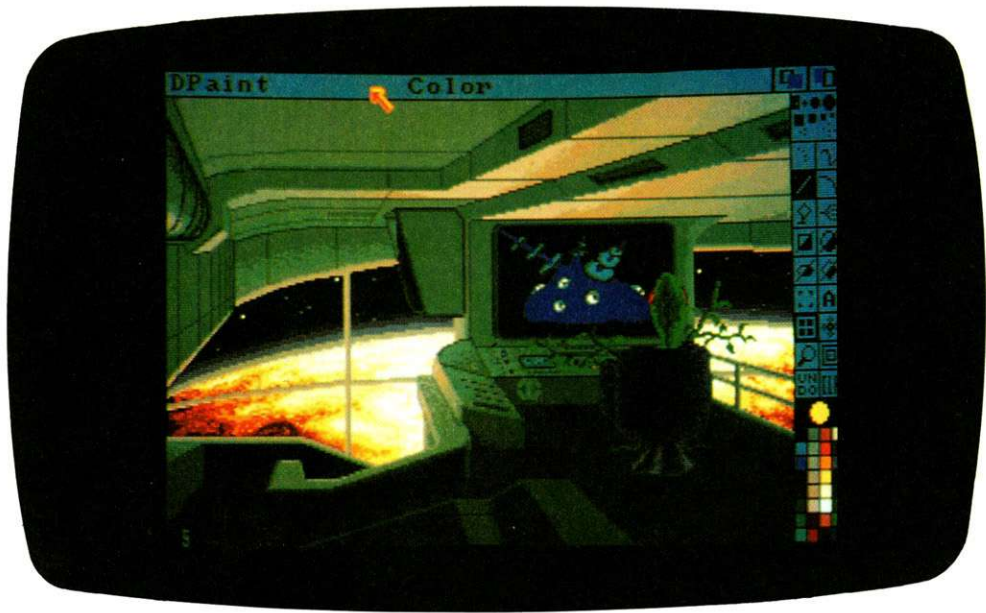
Ein AnimObject kann z.B. Bilder der verschiedenen Bewegungsphasen einer gehenden Person enthalten. Je mehr Phasen man verwendet, also je mehr Speicher man verwendet, desto flüssiger wirkt diese Bewegung. Gibt man dem AnimObject dann eine Geschwindigkeit, dann könnte diese Person z.B. langsam oder schnell über den Bildschirm schreiten. All dies geschieht, nach dem "Einschalten", ohne weiteres Zutun des eigentlichen Programms, das sich unterdessen um andere Dinge kümmern kann.

Das bekannte **Robocity**-Demo, das sich ebenfalls auf den Demo-Disketten befindet, die jeder Amiga-Händler bekommt, nutzt exakt diese Möglichkeit. Zwei Roboter und ein Hund bewegen sich darin recht flüssig vor einem festen Hintergrund. Die Farbseiten dieses Buches enthalten ein Bild aus **Robocity**.









1.5.2 Sound-Fähigkeiten

Nicht ganz so große Publizität wie die Grafik-Fähigkeiten haben die Sound-Möglichkeiten des Amiga gefunden. Sie sind nichtsdestotrotz aber auch recht beeindruckend. Der Amiga besitzt insgesamt 4 unabhängige Tonkanäle, von denen jeweils 2 zu einem Stereo-Kanal zusammengefaßt werden können. Jeder der 4 Kanäle kann eine völlig beliebige Wellenform mit beliebiger "Hüllkurve" ausgeben. Er benötigt dazu nur in sehr geringem Maße den Prozessor bzw. ein Eingreifen des Programms, das den Ton irgendwann einmal "angestoßen" hat.

Man darf von der Sound-Hardware des Amiga natürlich nicht zuviel verlangen. Es handelt sich nicht um einen Synthesizer, und die Klangqualität kann HiFi-Tests gewiß nicht standhalten. Für die Darstellung eines Klangs werden im Amiga z.B. 8 Bit breite Werte verwendet. HiFi-Geräte in Digital-Technik verwenden üblicherweise 16 Bit, was zu einer rund 250-mal besseren Klangqualität führt — von anderen Gesichtspunkten ganz abgesehen. Trotzdem können die vom Amiga erzeugten Töne oft recht natürlich klingen.

Ein Programm auf einer der Händler-Demo-Disketten und das Kompositionsprogramm MusiCraft von Amiga demonstrieren recht eindrucksvoll die musikalischen Fähigkeiten des Amiga. Für professionelle Kompositionsaufgaben ist das Programm *Deluxe Music Construction Set* gedacht, das von der Firma *Electronic Arts* vertrieben wird. Ihr Amiga-Händler führt Ihnen diese Programme gern vor.

Die vom Amiga erzeugten Töne können entweder vollständig "synthetisch" sein, also mehr oder weniger auf dem Papier bzw. am Computer entworfen worden sein. Oder man kann auch einen natürlichen Klang nehmen, seine Charakteristika analysieren (ihn *digitalisieren*), abspeichern und dann von der Amiga-Sound-Hardware wieder ausgeben lassen. Dabei kann man den Klang in sehr weiten Grenzen ändern und "mißbrauchen". Es wäre z.B. durchaus möglich, Beethovens 5. Symphonie mit simulierten Kochtopfdeckeln zu spielen.

Das *bouncing-ball*-Demo nutzt z.B. diese Möglichkeit, einen Klang aufzunehmen und dann verändert wiederzugeben, in recht eindrucksvoller Weise. Der Ton, den man hört, wenn der Ball vor eine der Wände oder den Boden schlägt, ist in Wirklichkeit das Geräusch eines Mikrophons, das von Amigas Software-Chef Bob Pariseau vor ein Garagentor geschlagen wird.

Wie es inzwischen scheinbar schon guter Ton bei neuen Computern zu werden scheint, ist der Amiga natürlich auch für die Kommunikation mit MIDI-Geräten (MIDI = Musical Instruments Digital Interface; der wohl wichtigste Standard für die Verbindung digitaler Musikinstrumente, wie

Synthesizer und Keyboards) vorbereitet. Hierzu muß ein entsprechendes Zusatzgerät an die serielle Schnittstelle des Amiga angeschlossen werden. Wem also die Klangqualität des Amiga auch dann noch nicht ausreicht, wenn er ihn an seine Stereoanlage angeschlossen hat, der sollte sich die Anschaffung des MIDI-Interfaces und eines entsprechenden MIDI-fähigen Synthesizers überlegen.

1.5.3 Sprach-Fähigkeiten

Eine Sound-Fähigkeit des Amiga darf man auf keinen Fall vergessen: die Sprachausgabe. Der Amiga dürfte so ziemlich der erste Computer sein, in dessen standardmäßiger Systemsoftware Sprachsynthese sofort mitgeliefert wird. Sie kann in jedem Programm eingesetzt werden und ist sehr problemlos zu programmieren. In der simpelsten Form übergibt man einfach einen ganz normalen Text an eine bestimmte Routine, die ihn daraufhin in akustischer Form ausgibt.

Die Aussprache kann man zusätzlich in verschiedenster Form beeinflussen. Die Stimmlage kann geändert werden; der Sprecher kann entweder im tiefen Baß oder als helle Frauenstimme sprechen. Die Geschwindigkeit der Sprache kann innerhalb weiter Grenzen geändert werden (in weiteren Grenzen, als sinnvoll ist). Und schließlich kann die Stimme entweder sehr ausdrucksvoll, mit betonten Worten und Satzzeichen oder monoton, computerhaft sprechen.

Wir können also schon bald Programme erwarten, die uns unsere Fehler nicht nur am Bildschirm melden oder durch einen nervösen Summton mitteilen, sondern auch rufen: "Heh du Dummkopf, was hast du da schon wieder angestellt?"

1.5.4 Sound- und Grafik-Hardware

Nachdem Sie nun einen ersten Eindruck von den Grafik- und Sound-Fähigkeiten des Amiga bekommen haben, wollen wir auch noch kurz darauf eingehen, wie sie zustandekommen — vor allem, wieso die Amiga-Grafik so schnell ist. Eine allzu genaue Kenntnis der Hardware-Zusammenhänge ist nicht (nicht einmal für die Programmierung) nötig, da einem die Amiga-Systemsoftware das meiste von der Bit-Fummelei abnimmt. Falls Sie allerdings ein Videospiel schreiben wollen, wie es die Welt noch nicht gesehen hat, werden Sie kaum um eine direkte Manipulation der Grafik-Chips

herumkommen. Dann sind allerdings wirklich Sachen möglich, die "die Welt noch nicht gesehen hat".

Die Qualität und Geschwindigkeit der Amiga-Grafik hat viele Gründe. Sie liegen aber im wesentlichen in den drei *Custom-VLSI-Chips* begründet, die wir schon mehrfach erwähnten. Z.B. hat jeder der drei Zusatzchips im Amiga ungefähr die Komplexität der CPU M68 000, des Hauptprozessors im Amiga. Der Amiga hat also, jedenfalls für bestimmte Aufgaben, ungefähr drei- bis viermal soviel Rechenleistung als ein Computer, der "nur" einen 68 000 enthält. Die drei Chips tragen sinnigerweise die Spitznamen "Agnus", "Denise" und "Paula" (obwohl gelegentlich auch andere Namen in der Computer-Presse zu lesen waren).

In diesen Chips befinden sich Zusatz-Prozessoren oder sog. CoProzessoren, die dem eigentlichen Mikroprozessor des Amiga viel Arbeit abnehmen. Sie arbeiten gleichzeitig "neben" diesem her, während er mit anderen Dingen beschäftigt ist. Einer dieser CoProzessoren ist z.B. der sogenannte *Blitter*, der unter anderem sehr schnell Linien zeichnen, Flächen mit einer bestimmten Farbe füllen und beliebige Teilbilder von einem Ort am Bildschirm zu einer andern Stelle bewegen oder auch entfernen kann.

Auch die Sound-Hardware holt sich ganz selbstständig die Daten, die sie zur Tonerzeugung benötigt, aus dem Speicher des Amiga. Erst wenn diese Daten zu Ende sind, meldet sie dies dem Programm und bittet um neue Daten. Zwischendurch aber wird das eigentliche Programm nicht durch die Erzeugung von Musik beeinträchtigt oder gebremst.

Der Amiga wird durch seine drei Zusatz-Chips ein echter Multi-Prozessor-Rechner, in dem zu jedem Zeitpunkt mehrere Mikroprozessoren gleichzeitig mit der Erfüllung der Wünsche des Benutzers beschäftigt sind. Er ist deshalb jedem Computer der gleichen Leistungsstufe, der alle Arbeit nur dem einen zentralen Mikroprozessor überläßt, haushoch überlegen!

1.6 Amiga-Software

Wie schon mehrfach erwähnt, kann die Hardware eines Computers noch so toll und technisch hochentwickelt sein — ohne Software, die diese Hardware entsprechend nutzt, bleibt sie unwichtig und nutzlos. Wir wollen nun ein wenig auf diese Software eingehen. Dabei geht die Sprache aber noch nicht auf die echten Anwendungsprogramme, die ihnen z.B. die Textverarbeitung oder Buchhaltung erleichtern. Wir wollen statt dessen einen kurzen Überblick

über die sogenannte System-Software geben, mit der Sie immer zu tun haben, wenn Sie einen Computer bedienen, ohne daß es Ihnen die meiste Zeit bewußt wird.

Diese System-Software erlaubt erst die Kommunikation der Computer-Hardware mit der Außenwelt. (Diese Außenwelt sind Sie selbst, aber auch Drucker und andere Geräte, die Sie an Ihren Amiga anschließen.) Sie stellt z.B. auch die Vorgänge im Innern des Computers oder den Inhalt des Speichers am Bildschirm dar.

1.6.1 Die Workbench

Nach dem Einschaltvorgang, der im folgenden Kapitel noch in aller Ausführlichkeit erläutert wird, erscheint normalerweise die *Workbench*. Die Workbench (zu deutsch *Werkbank*) ist die "Oberfläche" (der Computer-Fachmann sagt auch *Shell*, zu deutsch *Schale*) des Amiga. Von dieser Werkbank aus können Sie Programme starten, Dateien und Disketten kopieren, löschen und umbenennen.

Die Disketten, Dateien und Programme werden als Symbole auf dem Bildschirm dargestellt. Alle wichtigen Operationen können mühelos durchgeführt werden, indem man einfach mit einem kleinen Zeiger, den man mit der Maus bewegen kann, auf bestimmte Objekte zeigt und einen der beiden Knöpfe auf der Maus drückt. Bestimmte Bewegungen dieses Zeigers zu bestimmten Punkten können ebenfalls Befehle bedeuten.

Ihre Dateien und Programme können Sie in Schubladen ablegen und so etwas Ordnung auf Ihre Disketten bringen. Diese Schubladen werden Ihnen am Bildschirm auch wirklich als kleine Schubladen gezeigt. Sie können diese öffnen und nachschauen, welche Dateien darin sind. Sie können sie aber auch schließen und auf diese Weise Platz auf dem Bildschirm schaffen, weil Sie nicht mehr alle Dateien sehen, sondern nur noch die Schublade.

Wir wollen nun aber nicht mehr weiter auf die Workbench eingehen, da Sie in großer Ausführlichkeit in den folgenden Kapiteln des ersten Buchteils beschrieben wird.

Wie Sie aber recht bald bemerken werden, ist diese grafische Art der Computer-Bedienung sehr leicht zu erlernen und auch zu behalten. Sie brauchen sich nur sehr wenige Dinge zu merken. Die Möglichkeiten, die

Ihnen in jedem Augenblick zur Verfügung stehen, werden Ihnen immer gezeigt. Kurz: die Bedienung ist sehr intuitiv.⁹

1.6.2 Das CLI

Eine zweite, völlig verschiedene Methode der Bedienung des Amiga ist das CLI. CLI bedeutet *command line interface*; zu deutsch in etwa: *Kommandozeilen-Schnittstelle*. Diese Bedienungsweise ist der herkömmlichen Bedienungsweise, wie sie auch auf anderen Computern üblich ist, sehr ähnlich. Sie müssen dabei einen Befehl, ein kurzes englisches Wort, eintippen und einige Parameter, die diesem Befehl sagen, was er genau tun soll. Die Ergebnisse des Befehls werden Ihnen dann üblicherweise auch in Textform am Bildschirm aufgelistet. Der Amiga wird beim CLI quasi als *alphanumerisches Terminal* "mißbraucht".

Für viele Anwender gibt es wenig Gründe, sich mit dem CLI zu beschäftigen. Befehle in Textform sind wesentlich langwieriger einzugeben, schwerer zu behalten, und die teilweise langen Listen notwendiger Parameter ergeben viele Möglichkeiten, Fehler zu machen. Die Workbench ist eine viel "freundlichere" Umgebung.

Einige Dinge gehen aber mit einer textuellen Bedienungsweise, also dem CLI, wesentlich einfacher. Die Programmierung z.B. ist ein solches Anwendungsgebiet, in dem man mit dem CLI viel zügiger arbeiten kann. Das CLI als zusätzliche Möglichkeit, den Amiga zu bedienen, hat also durchaus seine Daseinsberechtigung.

Über das CLI werden Sie im zweiten Teil dieses Buches noch mehr erfahren; das Amiga-Handbuch schweigt sich darüber leider ziemlich aus.

1.6.3 Multitasking

Eine der Software-Eigenschaften, die es vor dem Amiga bei Personal Computern fast überhaupt nicht gab, ist die Fähigkeit, mehrere Programme gleichzeitig laufen zu lassen. Programme nennt man in diesem Zusammenhang oft auch *Tasks* oder *Prozesse* und die Eigenschaft, mehrere

⁹ Das Paket von Routinen in der Amiga-Systemsoftware, das für diese Dinge zuständig ist, heißt deshalb auch *Intuition*.

Programme gleichzeitig verarbeiten zu können, heißt dementsprechend *Multitasking*.

Wenn man von der Workbench aus ein Programm startet, übernimmt es nicht den ganzen Computer, wie bei Rechnern anderer Firmen üblich. Es öffnet ein neues Fenster und erledigt seine Ein- und Ausgaben in diesem Fenster. Falls es mehrere Fenster braucht, so kann es diese nacheinander öffnen oder sogar einen ganzen (virtuellen) Bildschirm anlegen, der einen eigenen Speicher und einen anderen Grafikmodus besitzt. Die anderen Programme aber, die zu diesem Zeitpunkt schon aktiv waren, z.B. auch die Workbench, laufen einfach weiter. Das hört sich im ersten Moment ganz toll an, insbesondere, weil dazu bis heute nur wesentlich größere und teurere Rechner fähig waren. Andererseits fragt man sich dann bald auch, wozu das Ganze? Schließlich kann man ja keine zwei oder drei Programme gleichzeitig bedienen, selbst wenn sie gleichzeitig laufen können.

Dieser Einwand stimmt, und man soll die Vorteile des Multitaskings auch nicht überschätzen. Andererseits ist Multitasking bei langwierigen Vorgängen aber auch eine ganz praktische Sache. Die meisten Druckvorgänge laufen z.B. auf dem Amiga standardmäßig im Hintergrund, also als separater Task. Während der Drucker rattert, kann man sich schon einmal anderen Dingen zuwenden. Genauso kann während des Formatierens einer Diskette bereits mit anderen Programmen gearbeitet werden. Auch langwierige Sortier- oder Rechenvorgänge können sehr gut ohne Ihren andauernden Eingriff ablaufen, und Sie können unterdessen schon einmal einen Brief schreiben. Der Wechsel von einem Programm zum anderen, ein Vorgang, den Sie häufiger benötigen werden, als Sie jetzt vielleicht meinen, ist wesentlich schneller, wenn jedes Programm nicht immer neu gestartet werden muß.

Alles in allem sind die Multitasking-Möglichkeiten des Amiga sehr vielversprechend. Wir sind uns sicher, daß Sie diese Möglichkeit nicht mehr missen wollen, wenn Sie sie erst einmal eine Weile genutzt haben.

1.7 Historie

Zum Abschluß dieser Einleitung noch ein paar Informationen zur Geschichte des Amiga. Es ist ja immer ganz interessant, auch das Vorleben einer neuen Freundin (oder eines neuen Freundes) kennenzulernen.

Wie kein anderer Computer vor ihm, wurde der Commodore Amiga schon lange vor seinem wirklichen Erscheinen von den wildesten Gerüchten

begleitet. Die ersten tauchten etwa im Winter '83 auf der COMDEX¹⁰ in den USA auf. "Amiga" war damals noch der Name einer Firma, die nur Joysticks, die kleinen Steuerknüppel für Videospiele — sehr gute allerdings — verkaufte. Auch heute noch kann man Amiga-Joysticks kaufen. Einen Personal Computer zu bauen und noch dazu einen, der alles bisher dagewesene in den Schatten stellen sollte, traute dieser Firma eigentlich kaum jemand zu.

Als man jedoch den Namen eines der Männer hörte, die am Design des Amiga mitarbeiten sollten, wurden die Zweifel etwas geringer. Für das Design der drei Spezial-Chips und ein großer Teil der grundsätzlichen Architektur des Amiga ist Jay Miner verantwortlich, der auch schon die Grafik-Chips für die Atari 400/ 800er Serie entwarf, die auch heute noch im 130 XE eingesetzt werden. Es sind im wesentlichen diese Chips, die für die relativ guten Grafik-Fähigkeiten der alten 8Bit-Ataris (und für den Ruhm Miners) verantwortlich sind. Die Philosophien hinter dem Atari-Chipsatz und dem des Amiga sind sehr ähnlich, nur sind die Amiga-Chips ungleich leistungsfähiger, da sie dem neuesten Stand der Microprozessor-Technik entsprechen. Der Amiga kann deshalb mit Fug und Recht als ein Nachfolger der alten Ataris bezeichnet werden — eher jedenfalls, als die neue Atari-ST-Serie. Die Chips von Jay Miner sollten diesem neuen Homecomputer Fähigkeiten geben, wie sie sonst nur in speziellen Grafikcomputern für einige Zehntausend Mark zu finden sind.

Die Firma Amiga machte in der Folge eine, vor allem auch finanziell, schwierige Zeit durch. Die Entwicklung der Grafik-Chips dauerte unerwartet lange und wurde dadurch natürlich auch unerwartet teuer.

In der Zwischenzeit brach der Homecomputer-Markt mehr oder weniger in sich zusammen. Mit neuen Homecomputern glaubte niemand mehr Geld verdienen zu können. Der Amiga wurde (allerdings nicht nur deshalb) immer größer und teurer und mauserte sich langsam aber sicher zum vollständigen Personal Computer. Die Firma Amiga benötigte immer mehr Geld für die immer länger dauernde Entwicklung eines sich kontinuierlich ändernden Computer-Projekts. Amiga hatte auch Geld von der Firma Atari für die Entwicklung neuer Grafik-Chips bekommen. Als sich das Geschäft mit Atari zerschlagen hatte, mußten diese Summen zusammen mit Vertragsstrafen zurückgezahlt werden. Kurz: Amiga hat eine sehr bewegte Geschichte hinter sich.

¹⁰ Eine große und wichtige Computermesse, die jedes Jahr zweimal (im Frühjahr und im Herbst) in den USA stattfindet.

Im Sommer '85 wurde der Amiga dann endlich in Rahmen einer millionen - schweren Multimedia-Show in New York vorgestellt. (Seine deutsche Premiere wird er auf der Hannovermesse (CEBIT) im März '86 haben.) Kaum einen der Anwesenden dürfte die Vorstellung kalt gelassen haben. Ganz abgesehen von reichlich Prominenz, mit der sich Amiga schmückte, so z.B dem schon etwas betagten Pop-Künstler Andy Warhol und der Rocksängerin Debby Harrie (Blondie), wurden vor allem die Grafik-Fähigkeiten des Amiga groß herausgestellt. Die Firma Amiga allerdings war inzwischen, nach einem kurzen gerichtlichen Nachspiel der Affäre mit Atari, von Commodore aufgekauft worden. Die Werbekampagne für die Markteinführung eines neuen Computers war für eine kleine Firma wie Amiga einfach zu kostspielig geworden.

Aus dem preiswerten Homecomputer für etwa 500 Dollar, den Amiga zunächst entwickeln wollte, war inzwischen ein leistungsfähiger PC geworden, der auch in der Geschäftswelt Erfolg haben sollte (aber auch ca. das dreifache kostet).

In diesen fast zwei Jahren zwischen den beiden Terminen kursierten die wildesten Gerüchte über eine wahre Supermaschine durch die Presse und die Computernetzwerke. Zu einem Zeitpunkt wurden die Gerüchte sogar Amiga selbst zu unglaublich, worauf ein Papier veröffentlicht wurde, in dem eine ungefähre Beschreibung an die Presse gegeben wurde; was die Maschine kann, aber auch, was sie nicht kann. Nun, da der Amiga endlich auch hier das Licht der Welt erblickt hat, stellt sich die Frage, ob sich das Warten gelohnt hat und die Gerüchteküche zu Recht so heftig gekocht hat. Das vorliegende Buch soll Ihnen — unter anderem — helfen, diese Frage für sich selbst beantworten zu können.

2

Auf der Werkbank

Das Programm, mit dem Sie einen Großteil Ihrer Arbeit am Amiga machen werden, ist die *Workbench* oder zu deutsch *Werkbank*.¹ Von dieser Werkbank aus werden Sie Ihre Disketten verwalten, Dokumente anlegen, kopieren, umbenennen, ablegen, löschen, Programme starten usw.. Prinzipiell laufen fast alle Dinge, die nicht unmittelbar etwas mit einem sogenannten Anwendungsprogramm zu tun haben, auf der Workbench ab.

2.1 Starten des Amiga

2.1.1 Der Einschaltvorgang

Um auf die Workbench zu gelangen, müssen Sie zunächst einmal natürlich Ihren Amiga und den zugehörigen Monitor einschalten. Nachdem Sie dies getan haben, blinkt eine ganze Weile ein rotes Lämpchen am linken Rand der

¹ Als dieses Buch entstand, lag uns leider nur die US-Version des Amiga und somit auch nur englischsprachige Software vor. Alle Begriffe, die sich auf diese Software beziehen, sind deshalb auf englisch. Deutsche Umschreibungen werden aber in den meisten Fällen zusätzlich gegeben, so daß Sie hoffentlich auch dann mit diesem Buch etwas anfangen können, wenn Sie bereits die deutsche Version des Amiga besitzen.

Haupteinheit. Der Amiga testet sich in dieser Zeit selbst. Er prüft z.B. seinen Speicher darauf, ob man aus diesem genau dieselben Zahlen lesen kann, die kurz zuvor hineingeschrieben wurden, schaut nach, ob die Speicher-erweiterung installiert ist, usw.. Wenn alles zu seiner Zufriedenheit ausgefallen ist, bittet er Sie, die Kickstart-Diskette einzulegen. Er zeigt Ihnen dazu das Bild einer Hand, die eine Diskette hält. Die Software auf dieser Diskette wird benötigt, um den Computer zu starten.

Eine kleine Sicherheitsvorkehrung vorweg: bevor Sie eine der Disketten, die Ihrem neuen Amiga beiliegen, in den Schlitz des Diskettenlaufwerks schieben, aktivieren Sie bitte deren Schreibschutz! Dies geht, indem Sie das kleine Plastikplättchen, das in einer der Diskettenecken zu finden ist, so verschieben, daß eine Öffnung entsteht, durch die Sie hindurchsehen können. Sie verhindern auf diese Weise, daß eine der relativ empfindlichen Disketten bei einem kleinen Fehler Ihrerseits, den Sie bei den folgenden Experimenten machen könnten, beschädigt werden kann. (Falls Ihnen diese kurze Erklärung des Schreibschutzes nicht genügt, können Sie eine etwas detailliertere am Ende dieses Kapitels finden.)

Bild 2-1 zeigt Ihnen, wie die Aufforderung, die Kickstart-Diskette einzulegen, im wesentlichen aussieht. In diesem Fall wird allerdings die Workbench-Diskette angefordert. Die Anforderung der Kickstart-Diskette sieht aber fast genauso aus. Achten Sie aber auf den Text, der — verkehrt herum — auf der Diskette steht.

Mit der Kickstart-Diskette hat es eine besondere Bewandnis, die wir ja bereits in der Einleitung erklärt haben. Die Software auf ihr wird nur beim Einschalten des Amiga einmal von der Diskette gelesen. Danach brauchen Sie diese Diskette während Ihrer gesamten Arbeit nie wieder. Erst, wenn Sie den Amiga ausschalten, geht der Inhalt des Speichers, der die Kickstart-Software aufgenommen hat, verloren.

Nachdem die Kickstart-Diskette verdaut wurde, bittet Sie der Amiga mit dem folgenden Bild, die Workbench-Diskette einzulegen. Sie müssen dazu zunächst die Kickstart-Diskette herausnehmen, denn auch die Workbench-Diskette muß beim Start in das in die Haupteinheit eingebaute Diskettenlaufwerk (das *interne Diskettenlaufwerk*) des Amiga eingelegt werden. Selbst wenn Sie stolzer Besitzer eines zweiten (*externen*) Diskettenlaufwerks sind, können Sie die Workbench-Diskette nicht in dieses schieben und die Kickstart-Diskette im internen Laufwerk belassen. **Der Amiga findet die Kickstart-Diskette und die Workbench-Diskette nur im internen Laufwerk!**

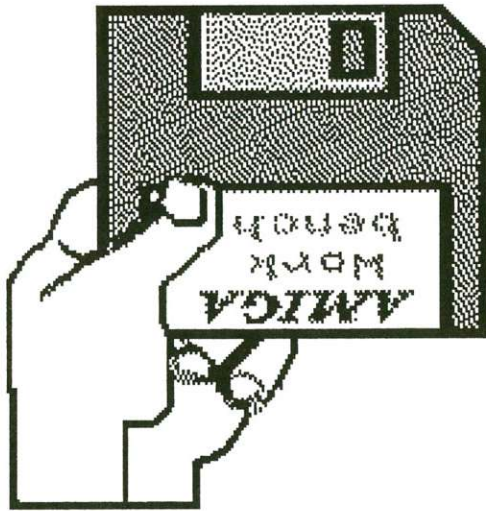


Bild 2 - 1: Die Aufforderung, die Workbench-Diskette einzulegen

Erscheint das Bild 2 - 1 nicht an Ihrem Bildschirm, so ist Ihre Kickstart-Diskette wahrscheinlich defekt. Wenn Sie noch eine Kopie davon besitzen, versuchen Sie es mit dieser. Falls das auch nichts fruchtet, schalten Sie Ihren Amiga noch einmal ganz aus und nach ca. 10 Sekunden wieder ein. Klappt es auch jetzt nicht, bleibt Ihnen nichts anderes übrig, als sich an Ihren Amiga-Händler zu wenden.

Ist aber alles gut gegangen und der Amiga hat auch Ihre Workbench-Diskette akzeptiert, so verschwindet Bild 2-1 nach wenigen Sekunden und das interne Laufwerk beginnt wieder hörbar zu arbeiten. Sie können übrigens immer an zwei Dingen feststellen, daß der Amiga auf die Disketten in einem der Diskettenlaufwerke zugreift. Zum einen hören Sie ein (hoffentlich) leises, schabendes, schnarrendes oder surrendes Geräusch und zweitens leuchtet das kleine rote Lämpchen unter dem Diskettenschlitz (*Diskettenlampe*) auf. **Während Sie eine dieser beiden Tatsachen registrieren, also die Diskettenlampe an ist oder Sie das Diskettengeräusch hören, sollten Sie niemals auf den Auswurfknopf drücken!** Diese Warnung werden Sie noch mehrmals lesen. Sie ist aber so wichtig, daß sie einige Wiederholungen verträgt. Gerade die 3,5-Zoll-Disketten des Amiga sind recht robust. Es bekommt aber keiner Diskette — und auch keinem Laufwerk — gut, wenn die Diskette aus dem Laufwerk entfernt wird, während der Computer auf die Daten darauf zugreift.

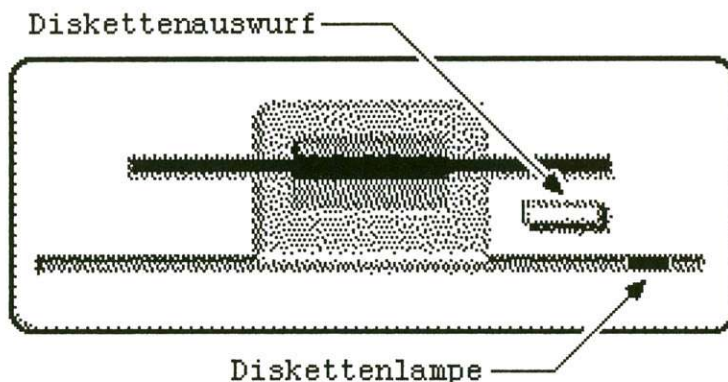


Bild 2 - 2: Ein Amiga-Diskettenlaufwerk

Nach einem kurzen Moment belebt sich der Bildschirm des Amiga wieder, und es erscheinen hintereinander eine Copyright-Meldung, die Erinnerung, die Uhr zu stellen und schließlich die Workbench am Bildschirm. Der Bildschirm färbt sich blau und am oberen Bildschirmrand sehen Sie einen weißen Streifen, in dem am linken Rand steht: **Workbench release 1.1: 368248 free memory**. (Die Zahl kann bei Ihnen allerdings eine andere sein.) Dieser Streifen heißt *Titelzeile* des Bildschirms.



Bild 2 - 3: Die Titelzeile des Workbench-Bildschirms

Am rechten Rand der Titelzeile stehen zwei kleine Symbole, über die in einem späteren Kapitel noch mehr zu erfahren ist. Falls Sie ein zweites Diskettenlaufwerk besitzen und sich in diesem schon eine Diskette befand, als Sie den Amiga gestartet haben, so erscheinen am rechten oberen Rand der blauen Fläche zwei kleine Bilder, die Disketten zumindest nicht unähnlich sind. Dies sind *Disketten-Abbilder* oder *Disketten-Icons*. (Die kleinen Bildchen, die beim Amiga die verschiedenen Objekte symbolisieren, mit denen wir zu tun haben, nennen wir generell *Icons*.) Besitzen Sie nur ein Laufwerk oder hatten Sie keine Diskette in Ihrem zweiten Laufwerk, so sehen Sie dort nur eines dieser Icons. Sie befinden sich nun auf der Workbench!

Wollen Sie die Workbench wieder verlassen, also den Amiga ausschalten, nehmen Sie bitte zuvor die Workbench-Diskette und/oder sämtliche anderen

Disketten aus dem (den) angeschlossenen(n) Diskettenlaufwerk(en). Ein Druck auf den Auswurfknopf an jedem Diskettenlaufwerk macht Ihren Disketten in diesem Fall das Leben leichter. **Sie sollten niemals "radikale" Aktionen, wie das Ein- oder Ausschalten des Amiga, durchführen, solange sich eine Diskette in einem Laufwerk befindet!**

Bevor wir nun in die Tiefen der Workbench eindringen, noch ein Wort zur Terminologie. Der Vorgang, den der Amiga soeben durchlaufen hat, war der *Einschaltvorgang*. Er beinhaltet das Einschalten des Amiga, das Einlesen der Kickstart-Software und das Starten der Workbench mit der Workbench-Diskette.

Der zweite Teil, das Starten der Workbench, heißt *Neustart* des Amiga. Der Computerfachmann sagt dazu auch oft *Rebooten*². Zum *Neustart* gehört nicht das Einschalten und nicht das Einlesen der Kickstart-Disketten. Selbst nach einem schwerwiegenden Programmfehler, der Ihren Amiga total "in Unordnung" gebracht hat, genügt immer ein Neustart, um wieder "sauber" von vorne anzufangen. Für ein Programm ist es technisch unmöglich, etwas an der Kickstart-Software zu zerstören, wenn sie erst einmal eingelesen ist.

2.1.2 Gezielter Neustart

Sollte Ihnen einmal das schreckliche Ereignis eines solchen Programmabsturzes widerfahren, oder sollte es Ihnen passieren, daß Ihr Amiga einmal auf keinen Ihrer Versuche, seine Aufmerksamkeit zu erregen, reagiert, dann können Sie einen Neustart erzwingen. Drücken Sie dazu die beiden Tasten mit dem A-Symbol links und rechts neben der Leer-Taste auf der Tastatur und dann gleichzeitig auf die Taste <CTRL>. Sie können die Tasten auch nacheinander drücken; es muß nur einen kurzen Augenblick geben, in dem alle drei zugleich niedergedrückt sind. (Sie benötigen dazu beide Hände.) **Bevor Sie dies machen, vergewissern Sie sich, daß keines der Diskettenlaufwerke mehr aktiv ist und nehmen Sie am**

² Der Begriff *booten* oder *rebooten* kommt vom englischen Wort *bootstraps* (Schnürsenkel) und hat etwas mit der Vorstellung zu tun, daß man sich selbst an seinen Schnürsenkeln aus einem Sumpf zieht. (Im deutschsprachigen Raum bevorzugt man allerdings das Ziehen an den eigenen Haaren — siehe Münchhausen.) Wenn man genau versteht, was beim Start eines Computers über Software, die er erst von der Diskette lesen muß, vor sich geht, liegt dieser absurde Vergleich gar nicht einmal so fern.

besten auch alle Disketten aus den Laufwerken (durch Drücken auf die entsprechenden Auswurfasten).

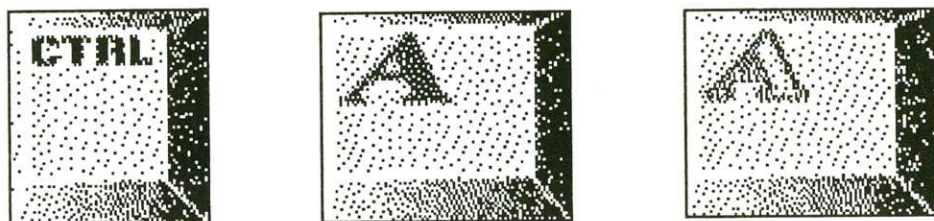


Bild 2 - 4: *Tastenkombination für den Neustart des Amiga*

Nachdem Sie die drei Tasten losgelassen haben, blinkt wieder die Betriebsleuchte links an der Haupteinheit auf, und nach kurzer Zeit erscheint die Aufforderung, die Workbench-Diskette einzulegen (Bild 2-1). Verfahren Sie dann wie oben beschrieben.

Die Tastenkombination für den Neustart sollte nur in absoluten Notfällen angewandt werden! Alle Daten, die sich zu dem Zeitpunkt, zu dem Sie den Amiga neu starten, noch nicht auf Disketten befinden, gehen dabei verloren. Aus Versehen kann diese Tastenkombination glücklicherweise aber kaum vorkommen — außer Sie besitzen 20 Zentimeter lange Finger oder lassen Gegenstände auf die Tastatur fallen und halten dabei gleichzeitig Tasten fest.

Wenn Sie gerade auf der Workbench sind und keine Programme mehr aktiv sind, probieren Sie den "gewaltsamen" Neustart doch einfach einmal aus! Warten Sie dazu, bis sämtliche Disketten-Aktivitäten beendet und alle entsprechenden Lampen erloschen sind, und nehmen Sie dann die Diskette(n) aus dem (den) Laufwerk(en). Dann drücken Sie die drei Tasten nieder und lassen Sie wieder los.

2.2 Die Maus

Das wichtigste Bedienungselement Ihres Amiga ist die Maus. Sie kann für einen großen Teil aller anfallenden Arbeiten auf der Workbench und auch innerhalb der meisten Programme, die Sie für den Amiga kaufen können, eingesetzt werden. Gerade auf der Workbench ist die Tastatur nahezu überflüssig — ohne Maus sind Sie hingegen (fast) rettungslos verloren. Dies

wird Sie wahrscheinlich verblüffen, wenn Sie schon andere Computer kennen. Bei vielen anderen Computern (mit ein oder zwei Ausnahmen) ist die Tastatur nämlich elementar für die ganze Bedienung des Gerätes, und ohne Tastatur wären Sie völlig aufgeschmissen. Auf dem Amiga ist das anders — und deshalb einfacher, wie Sie bald merken werden. Die Maus und vor allem die zugehörige Software hat einige Vorteile gegenüber der üblichen Computer-Bedienung mit über die Tastatur einzugebenden Befehlen.

Dies soll nicht heißen, daß Sie auf dem Amiga Textverarbeitung ohne Tastatur betreiben können. Für viele Anwendungen ist die Tastatur immer noch unentbehrlich. Wenn man aber von den Vorgängen auf der Workbench zum normalen (nicht computerisierten) Leben eine Analogie zieht, dann kann man sagen, daß Sie immer dann ohne Tastatur auskommen werden, wenn Sie das beim entsprechenden Vorgang ohne Computer auch würden. Die Tastatur wird beim Amiga wirklich nur für die Texteingabe benutzt, und sonst nicht.

2.2.1 Funktionsweise der Maus

Bild 2–5 zeigt Ihnen die Maus des Amiga von oben. Die Ähnlichkeit mit einer lebenden Maus ist nicht gerade frappierend, aber mit etwas Mühe zu erkennen. Ich habe allerdings schon Computer-Mäuse gesehen, die von ihrem Besitzer einen Pelz verpaßt bekommen haben und dann doch recht putzig aussahen. Wenn Sie also ihren Computer gern warm und flauschig bedienen wollen ... (geschickt gemacht, schadet eine solche Pelzhülle der korrekten Funktion überhaupt nicht).

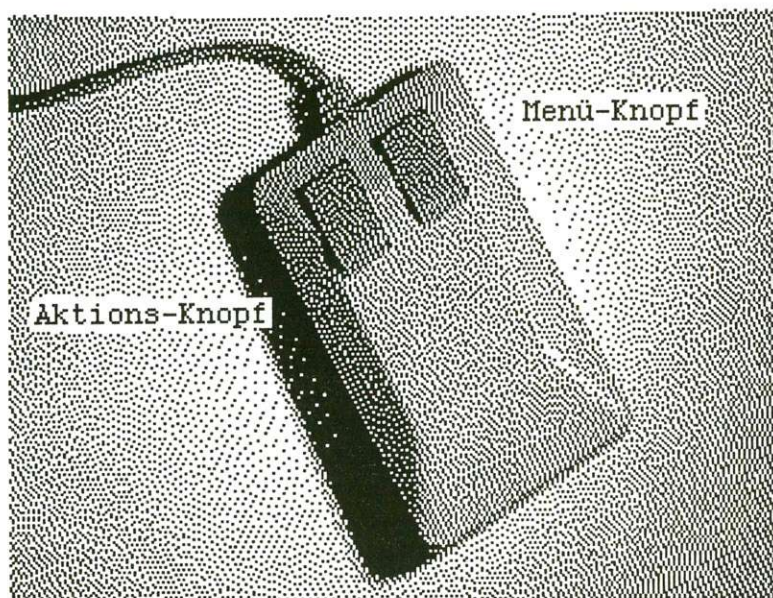


Bild 2 – 5: *Die Amiga-Maus*

Die Oberseite der Maus ist glatt und sollte gut in die locker darauf gelegte Handfläche passen. Ob sie das auch tut, darüber kann man geteilter Meinung sein — Geschmacksache. (Ihre Form ist aber wohl unbestreitbar schnittig und modern.) An der Oberseite trägt sie zwei Knöpfe. Wird einer dieser Knöpfe gedrückt, so bewirkt dies bei den meisten Programmen eine bestimmte Aktion — abhängig vom gerade laufenden Programm können es allerdings verschiedene Aktionen sein.

Der linke der beiden Knöpfe heißt *Aktions-* oder *Selektions-Knopf*, der rechte *Menü-Knopf*. (Keine Angst, Sie haben sich nicht in ein Kochbuch verirrt. Die Namen der Knöpfe werden weiter unten noch erläutert.)

Die Unterseite der Maus ist (zumindest für ihre Funktion) noch etwas interessanter. Falls Sie dieses Buch nicht gekauft haben, weil Sie erst noch Besitzer eines Amiga werden wollen, sondern es schon sind, drehen Sie die Maus einmal auf ihre Rückseite. In der Mitte des Mausbodens sehen Sie ein kleines Loch und darin eine kleine Gummikugel. Immer wenn Sie die Maus — richtig herum — auf eine glatte Oberfläche legen und sie dann bewegen, dreht sich die Kugel und überträgt diese Drehung auf zwei kleine Rädchen in ihrem Innern. Diese Rädchen senden bei Drehung elektrische Impulse zum Amiga, die ihm genau sagen, in welche Richtung und wie weit Sie die Maus bewegt haben.

Wie diese Impulse genutzt werden, ist prinzipiell natürlich Sache des gerade laufenden Programms. Meist werden Sie aber dazu verwendet, einen kleinen Zeiger auf dem Bildschirm synchron zur Maus zu bewegen.

2.2.2 Erste Maus-Spiele

Falls Sie noch nicht mit der Maus vertraut sind, sollten Sie jetzt versuchen, es zu werden. Begeben Sie sich dazu auf die Workbench (s.o.) und nehmen Sie die Maus in die Hand. Legen Sie sie dann auf eine glatte Unterlage, die wenigstens die Größe eines DIN-A4-Blattes haben sollte. Später werden Sie lernen, auch mit weniger Platz auszukommen; zunächst kann es aber gar nicht genug Platz für sie geben. In welche Hand Sie die Maus nehmen, ist im wesentlichen Geschmacksfrage. Zu Beginn ist es aber wahrscheinlich einfacher, wenn Sie sie als Linkshänder in die linke und als Rechtshänder in die rechte Hand nehmen. Mit ein wenig Übung wird diese Unterscheidung aber bald gegenstandslos werden und Sie werden die Maus mit der Hand bewegen, in deren Nähe Platz auf dem Schreibtisch ist.

Wenn Sie die Maus nun auf dieser Oberfläche bewegen, können Sie sehen, wie ein kleiner roter Pfeil auf dem Bildschirm diesen Bewegungen folgt. Dies ist der Mauszeiger — das wichtigste Objekt auf dem Bildschirm. Die folgende Abbildung zeigt Ihnen diesen Zeiger in vergrößerter Form — leider nur in schwarzweiß. Auf dem Bildschirm erscheint er vor allem rot, besitzt in Wirklichkeit aber mehrere Farbtöne, damit er auch über den unterschiedlichsten Hintergrundfarben sichtbar bleibt.

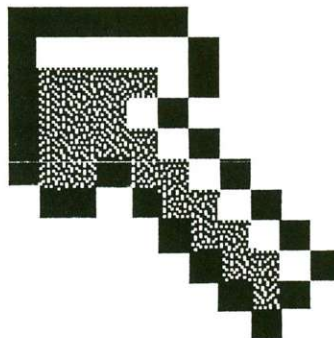


Bild 2 – 6: *Der Mauszeiger des Amiga*

Versuchen Sie nun einmal, die vier Ecken des Bildschirms mit diesem Zeiger gezielt anzufahren. Dies ist unter Umständen gar nicht so einfach, wenn Sie die Maus nicht richtig halten. **Richtig gehalten wird die Maus immer**

so, daß ihr Schwanz (der Kabelanschluss) auf den Bildschirm weist. Schieben Sie die Maus auf der Unterlage nach vorn, so geht der Zeiger auf dem Bildschirm nach oben. Ziehen Sie sie an ihren Körper, geht der Zeiger nach unten. Bewegen Sie die Maus nach rechts, und der Zeiger geht nach rechts. Bewegen Sie sie nach links, und der Zeiger folgt.

Drehen Sie die Maus testweise einmal so, daß der Schwanz nach links zeigt, und bewegen Sie sie dann nacheinander in die vier Bildschirmecken. Wie Sie sehen, ist dies viel schwieriger, da die Entsprechung von Hand- bzw. Mausbewegung zu Zeigerbewegung nicht mehr stimmt!

Versuchen Sie auch einmal, die Spitze des Mauszeigers genau auf die obere linke Ecke des Diskettensymbols in der rechten oberen Bildschirmecke zu bringen. Sie werden merken, daß es gar nicht so einfach ist, die Handbewegungen mit der Bildschirmdarstellung zu koordinieren und der Versuchung zu widerstehen, permanent auf die Maus zu schauen. Sie werden sich aber bald daran gewöhnen. Die Erfahrung zeigt, daß die Maus nach etwa ein bis zwei Stunden keinem Anwender mehr Probleme macht.

Die Kopplung zwischen Maus und Mauszeiger wird auch ihnen bald so natürlich erscheinen wie ein direktes Zeigen mit dem Finger.³ Sie ist sogar so selbstverständlich, daß in diesem Buch manchmal der Fehler gemacht wird, von der Maus zu sprechen, wenn eigentlich der Mauszeiger gemeint ist. Da die beiden unzertrennlich sind, ist dieser Fehler aber hoffentlich zu verzeihen.

2.2.3 Die Knöpfe der Maus

Wie die Maus bewegt wird, und was diese Bewegung bewirkt, haben wir ja jetzt gesehen. Auf der Maus-Oberseite befinden sich aber auch noch zwei Knöpfe. Da beide Knöpfe häufig gebraucht werden, sollten Sie sich eine Handhaltung angewöhnen, mit der Sie beide Knöpfe unabhängig voneinander niederdrücken können. Zeige- und Mittelfinger eignen sich dazu gut. Sie können die Maus dann mit Daumen und Ringfinger sehr gut umfassen und diese beiden Finger immer locker auf den Knöpfen liegen lassen. Ganz so

³ Es gibt Computer, die auf einen Fingerdruck auf den Bildschirm reagieren. Dieses System ist der Maus aber bei weitem unterlegen. Sie können mit der Maus nach einiger Übung jeden Bildschirmpunkt genau bezeichnen. Ihr Finger ist dazu wahrscheinlich doch etwas zu dick. Zudem brauchen Sie ihre Hand nicht die ganze Zeit vor den Bildschirm zu halten, sondern können Sie bequem auf der Maus liegen lassen.

locker wird Ihre Handhaltung am Anfang bestimmt nicht sein, aber keine Angst: das legt sich mit der Zeit (und Übung).

Bewegen Sie den Mauszeiger nun einmal auf die freie blaue Fläche auf dem Bildschirm, drücken Sie dann kurz auf den linken Mausknopf und lassen ihn sofort wieder los. Es passiert nichts! Wenn Sie dabei aber auf dem Bildschirm schauen, werden Sie feststellen, daß der Amiga Ihren Knopfdruck bemerkt hat. Die Titelzeile des Bildschirms flackert jedesmal leicht, wenn Sie den linken Knopf drücken. Dieser Vorgang des Drückens und Loslassens der Maustaste heißt *Mausklick* und das entsprechende Verb heißt *klicken* (mit der Maus).

Gehen Sie nun mit dem Mauszeiger über das Disketten-Icon in der rechten oberen Bildschirmcke. Die Spitze des roten Pfeils sollte sich über der weißen Fläche des Symbols befinden. Drücken Sie nun wieder kurz die linke Maustaste und lassen sie sofort wieder los. Das Symbol wechselt dadurch seine Farbe. Wer nennen diesen Vorgang *Selektieren* oder *Auswählen* eines Objekts. Der linke Knopf heißt dementsprechend ja auch Selektions-Knopf oder Aktions-Knopf. Bewegen Sie den Zeiger nun wieder auf die blaue Bildschirmfläche und klicken Sie erneut mit dem linken Knopf. Das Diskettensymbol erhält daraufhin wieder seine alten Farben. Sie haben das Symbol *deselektiert*. Versuchen Sie es zur Übung gleich noch einmal: Anklicken des Disketten-Icons und dann Anklicken der blauen Fläche.

Sie haben damit schon eine wesentliche Operation der Workbench erlernt: das Auswählen eines Objektes zur Bearbeitung. Wenn Sie später irgendeinen Befehl geben, z.B. *kopiere*, müssen Sie dem Amiga zuvor immer sagen, mit welchem Objekt er diesen Befehl durchführen soll (z.B. *was* er kopieren soll). Das vorherige Anklicken eines oder mehrerer Objekte erfüllt genau diesen Zweck. Sie suchen sich z.B. eine Diskette aus und sagen dem Amiga dann *kopiere* (diese Diskette).

Mit dem linken Knopf können Sie aber nicht nur Objekte auswählen und wieder deselektieren, sondern z.B. auch Objekte bewegen. Dazu drücken Sie die linke Maustaste und halten sie dann gedrückt. Bewegen Sie nun den Mauszeiger über das Diskettensymbol, drücken Sie die linke Taste und halten sie gedrückt. Sie sehen, wie sich der Mauszeiger (und natürlich auch die Farben der Diskette) ändert, sobald die Taste niedergedrückt wird. Er ist jetzt zu einem roten Kreis geworden, in dem ein durchsichtiges Kreuz steht.

Bewegen Sie nun diesen Zeiger in die Mitte des Bildschirms **und halten dabei die Maustaste immer unten**. Dieser Vorgang heißt *Ziehen* der Maus. Beim *Ziehen* bleibt im Gegensatz zum einfachen *Bewegen* die Maustaste gedrückt, während die Maus bzw. der Mauszeiger bewegt wird. Sobald Sie die Maustaste nun loslassen, erscheint das Diskettensymbol an der

Stelle, an der sich auch der Mauszeiger in diesem Moment befand. Sie haben das Disketten-Icon auf diese Weise verlegt. Versuchen Sie, es nun wieder zurückzulegen. Dazu "ergreifen" Sie das Icon wieder mit der Maus (drücken die Maustaste nieder, während der Zeiger sich über dem Icon befindet) und bewegen den roten Kreis in die rechte obere Bildschirmecke. Dort lassen Sie die Maustaste los, und die Diskette liegt wieder (fast) an ihrer alten Stelle.

Wir werden später noch eine Reihe anderer Icons kennenlernen, die auf der Workbench des Amiga erscheinen können. Alle diese Icons haben gemeinsam, daß man sie mit der linken Maustaste zur Bearbeitung auswählen (selektieren) und sie bei gedrückter linker Maustaste auch bewegen kann.

Die Maus besitzt aber noch einen zweiten Knopf. Diesen rechten Mausknopf, den sogenannten Menü-Knopf, haben wir überhaupt noch nicht verwendet. Falls Sie ihn schon aus Versehen gedrückt haben, werden Sie gesehen haben, daß er nicht dieselbe Funktion wie der linke hat — die obigen Beispiele sind "mit rechts" nicht durchführbar. Vielleicht haben Sie aber schon eine Veränderung auf dem Bildschirm bemerkt, als Sie die Menü-Taste herunterdrückten. Falls nicht, tun Sie es jetzt noch einmal.

Sobald Sie mit der rechten Taste klicken (und solange Sie sie festhalten), ändert sich die Titelzeile. Anstelle des Textes **Workbench release 1.1.....** erscheinen darin die drei Worte **Workbench**, **Disk** und **Special**. Dies sind die *Titel* der drei *Menüs*, aus denen Sie die Befehle der Workbench auswählen können. Solange die Menü-Titel sichtbar sind, nennen wir die Titelzeile des Bildschirms manchmal übrigens auch *Menü-Zeile*, um ihre neue Funktion dadurch etwas zu betonen. Menüs sind eine sehr bequeme und kaum fehleranfällige Möglichkeit, einem Computer Befehle zu erteilen. In den nächsten Abschnitten erfahren Sie noch mehr darüber. Fast alle Amiga-Programme verwenden diese Möglichkeit. Natürlich gibt es in der Workbench mehr als drei Befehle. Die Befehle der Workbench sind aber in drei Gruppen oder auf drei Menüs aufgeteilt. Die Titel dieser drei Gruppen sind es, die in der Titelzeile erscheinen, wenn der Menü-Knopf gedrückt wird.

2.3 Menüs

2.3.1 Aufrufen von Menüs

Um die Menüs des Amiga näher kennenzulernen, halten Sie zunächst einmal die (rechte) Menü-Taste der Maus nieder und bewegen Sie dann den Mauszeiger in die Titelzeile über das erste Wort **Workbench**.. Das Bild, das erscheint, sobald der Zeiger sich über dem Wort Workbench befindet, dürfte ungefähr so aussehen wie die folgende Abbildung.

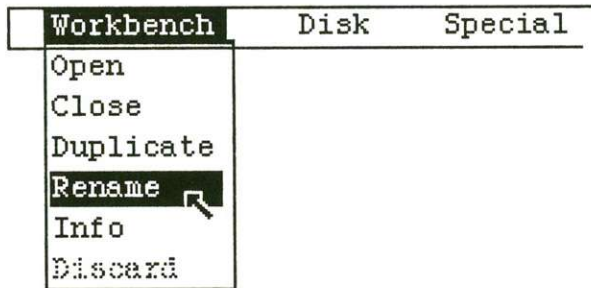


Bild 2 - 7: *Das Workbench-Menü*

Ein weißes Rechteck "klappt" herunter und zeigt Ihnen einige der Befehle, die die Workbench anbietet. Es kann durchaus sein, daß alle Befehls-Namen, die Sie sehen, so schlecht lesbar sind wie im obigen Bild das Wort **Discard**. Die Gründe dafür werden Sie weiter unten erfahren. Zunächst soll uns das einmal nicht zu sehr kümmern. Bewegen Sie den Mauszeiger (bei gedrückter Menü-Taste) nun nach rechts über das nächste Wort **Disk**. **Belassen Sie den Mauszeiger unbedingt in der Titelzeile und halten Sie die Maustaste gedrückt.** (Falls Ihr Zeigefinger erlahmen sollte, bewegen Sie den Zeiger über die freie blaue Fläche, lassen dann die Menü-Taste los und beginnen nach einer Wartezeit wieder von vorne.) Sobald sich der Mauszeiger über **Disk** befindet, verschwindet das erste Menü und ein neues taucht auf. Bewegen Sie ihn weiter nach rechts und der Vorgang wiederholt sich: das zweite Menü verschwindet und ein drittes erscheint unter dem Wort **Special**.. Bewegen Sie die Maus in der Titelzeile hin und her und sie können alle Befehle, die Ihnen in einem Programm — in diesem Fall dem Workbench-Programm — zur Verfügung stehen, schnell überblicken.

Auch für die Zukunft ist es eine gute Angewohnheit, bei gedrücktem Menü-Knopf über die Titelzeile zu gehen, wenn man ein neues Programm erlernen muß oder einmal nicht mehr recht weiter weiß. Dies ist eine unschätzbare Orientierungshilfe in jedem Programm. Das Schöne an den Menü-Befehlen des Amiga ist auch, daß man jederzeit sieht, welche Befehle gerade sinnvoll möglich sind und welche nicht. Die in einem Moment nicht möglichen oder sinnlosen Befehle erscheinen stets grau oder "geisterhaft", während die Befehle, die sie wirklich auswählen können, fett und gut lesbar sind.

Solange Sie keine Objekte in der Workbench ausgewählt haben, sind allerdings fast alle Einträge in den drei Menüs grau.

2.3.2 Das Erteilen eines Menü-Befehls

Bis jetzt haben wir uns ja nur den Inhalt der Menüs angeschaut, jedoch noch nichts damit gemacht. Nun aber geht es "zur Sache". In jedem Menü erscheint eine Reihe von kurzen Worten. Dies sind die Namen der verschiedenen *Menü-Befehle*, die die Workbench versteht. Allgemein nennt man diese Worte auch *Menü-Punkte*. Klappen Sie nun einmal das Menü **Special** herunter (wie das geht, wurde ja bereits im letzten Absatz erläutert) und belassen Sie den Mauszeiger über dem Namen **Special**.

Ihnen stehen jetzt die Befehle **Clean Up** (engl. für *aufräumen*), **Last Error** (engl. für *letzter Fehler*), **Redraw** (engl. für *neu zeichnen*) und **Version** (engl. für *Versionsnummer zeigen*) zur Verfügung. Einige davon sind wahrscheinlich grau. Niemals grau aber wird der Befehl **Version**, der uns die Versionsnummer der Software zeigt, und genau diesen wollen wir jetzt aufrufen.

Bei gedrücktem Menü-Knopf wird dazu die Maus innerhalb des weißen Kastens, der die Menü-Punkte enthält, nach unten bewegt. Jeder Menü-Punkt, der unter den Mauszeiger gerät und nicht grau gezeichnet war, wird dabei orange auf schwarz statt blau auf weiß gezeichnet. Probieren Sie es einmal aus, indem Sie den Mauszeiger im Menü **Spezial** auf und ab bewegen. Lassen Sie dabei die Menütaste aber bitte nicht los! Sobald der Zeiger das Menü verläßt, werden alle Befehle wieder normal gezeichnet; das Menü bleibt aber sichtbar, solange die Menütaste gedrückt bleibt.

Bewegen Sie nun den Zeiger über das Wort **Version**, so daß es orange auf schwarz geschrieben wird. Jetzt lassen Sie die Menü-Taste los. Stellen Sie sich bitte einen kleinen Tusch vor, denn soeben haben Sie dem Amiga den ersten Befehl erteilt. Er führt ihn auch sofort aus. Der Bildschirm flackert kurz auf und zeigt ihnen dann oben links in der Titelzeile die genaue

Versionsnummer der Workbench (z.B. **Workbench Version 31.334**). Zur Eingewöhnung versuchen Sie es am besten gleich noch einmal.

Jedes Mal, wenn ein Menü-Punkt orange auf schwarz hervorgehoben wird,⁴ und Sie in diesem Augenblick die Menü-Taste loslassen, erteilen Sie dem Amiga einen Befehl. Sie brauchen diesen Befehl also nicht wie auf anderen Computern einzutippen, sondern wählen ihn einfach aus dem Menü aus, wie Sie auch ein Gericht aus einem Restaurant-Menü auswählen würden — daher der Name.

Der **Version**-Befehl ist ein Menü-Befehl, der keine zusätzlichen Informationen über die Objekte, mit denen er arbeiten soll, benötigt. Dies stellt unter den Menü-Befehlen aber eher eine Ausnahme dar. Fast alle anderen Befehle benötigen solche zuvor ausgewählten Objekte, die der Computerfachmann *Argumente* bzw. *Parameter* des Befehls nennt. Wir wollen uns nun einmal einen solchen Befehl mit Argumenten anschauen.

Selektieren Sie dazu das einzige Objekt, das bisher auf der Werkbank zu sehen ist: das Disketten-Icon oben rechts am Bildschirm (durch einen Klick mit der linken Taste auf dieses Icon). Dann drücken Sie wieder die Menü-Taste und gehen nun über das Wort **Workbench** in der Menü-Zeile. Der erste Befehl in diesem Menü heißt **Open** (engl. für *öffnen*). Bewegen Sie die Maus über das Wort **Open** und lassen Sie die Taste los. Sie haben dem Amiga dadurch sinngemäß den Befehl *Öffne diese Diskette* gegeben.

Das erste, was geschieht, ist, daß ein weißer Rahmen auf dem Bildschirm erscheint. Dieser Rahmen trägt am oberen linken Rand den Titel Workbench und es tauchen nun langsam unter hörbarem Arbeiten des Diskettenlaufwerks nacheinander eine Reihe von Symbolen darin auf: einige Schubladen, eine Uhr, ein Mülleimer und ein Symbol, das der Systemeinheit des Amiga verdächtig ähnelt. Dieser weiße Rahmen und die Fläche in seinem Innern ist ein sogenanntes Fenster, in dem uns der Inhalt der Diskette namens Workbench gezeigt wird.

Diese Eigenschaft des **Open**-Befehls sollten Sie sich unbedingt merken. **Obwohl Sie den Open-Befehl auf die verschiedensten Objekte anwenden können, zeigt er Ihnen stets (meist in einem Fenster), was sich in diesem Objekt verbirgt.** Das Pendant zum Befehl **Open** (*öffnen*) ist natürlich der Befehl **Close** (engl. für *schließen*). Dieser Befehl

⁴ Es gibt auch noch andere Methoden, einen Menü-Punkt hervorzuheben, über die in den folgenden Kapiteln noch mehr berichtet wird. Auch können die Farben auf ihrem Bildschirm durchaus anders sein. Ein Menü-Punkt gilt aber immer dann als hervorgehoben, wenn er seine Färbung oder Form wechselt.

(*Close*) findet sich im Menü *Workbench* direkt unter *Open*. Wählen Sie ihn nun aus! Wie Sie sehen, verschwindet das Fenster und Sie blicken wieder auf die blaue Fläche der Workbench.

Klicken Sie nun wieder das Disketten-Icon an und wählen *Open* aus, und das Fenster erscheint wieder. Wenn Sie diesmal etwas auf den Mauszeiger achten, so werden Sie bemerken, daß er seine Form ändert, während sich das Fenster, das aus dem Disketten-Icon hervorgegangen ist, langsam mit Objekten füllt. Er wird zu einer Art Sprechblase (wie in einem Comic-Heftchen) in der zwei Z's stehen. Der Amiga sagt Ihnen auf diese Weise, daß er beschäftigt ist (bzw. ein kleines Nickerchen macht) und nicht mehr sofort auf alle Ihre Eingaben reagieren kann. Immer, wenn Sie diese kleine Sprechblase sehen, ist ein Programm mit einer etwas länger andauernden Tätigkeit beschäftigt. Falls Sie zur Ungeduld neigen, wird Sie diese Sprechblase bestimmt bald etwas nerven. Manche Dinge dauern nun aber einmal etwas länger und leider können Sie auch recht wenig dagegen tun.⁵

Hier wollen wir nun die Besprechung der Menüs erst einmal abbrechen und uns den Fenstern des Amiga zuwenden — neben der Maus und den Menüs wohl dem dritten wichtigen Konzept der Amiga-Software.

2.4 Fenster

Die Fenster des Amiga besitzen einige interessante Eigenschaften, die die Arbeit mit diesem Computer wesentlich erleichtern. Unter anderem ermöglichen sie es, "mehr Informationen gleichzeitig auf dem Bildschirm erscheinen zu lassen, als eigentlich Platz haben". Sollten Sie Zweifel daran haben, legen Sie dieses Buch nicht als Unsinn beiseite. Die Behauptung steht ja in Gänsefüßchen und ist also mit Vorsicht zu genießen. Ein Quentchen Wahrheit steckt aber darin.

2.4.1 Bewegen von Fenstern

Das einfachste, was wir mit dem Fenster machen können, das auf den *Open*-Befehl hin erschien, ist, es zu bewegen. Die Bewegung eines Fensters geht

⁵ Der Kauf einer — bedauerlicherweise meist recht teuren — Festplatte, lindert solche Beschwerden allerdings erheblich.

ganz ähnlich wie das Bewegen des Disketten-Symbols in einem der vorangegangenen Abschnitte. Während man das Disketten-Symbol aber einfach irgendwo mit der Maus ergreifen konnte, sind Fenster etwas wählerischer.

Wichtig für viele Operationen mit einem Fenster ist die Titel-Leiste. Dies ist der breite Streifen am oberen Fensterrand, der unter anderem den Namen des Fensters am linken Rand trägt. Diese Titelzeile samt Fensternamen ist zunächst etwas undeutlich. Klicken Sie auf die blaue Fläche im Innern des Fensters (nicht auf eines der Symbole darin). Das Fenster wird dadurch "zur weiteren Bearbeitung" selektiert. Die vorher schlecht leserliche Schrift des Fenster-Titels wird klar und die kleinen blauen Pünktchen neben dem Namen werden zu vier blauen Linien. Dies bedeutet, daß das Fenster nun selektiert ist.

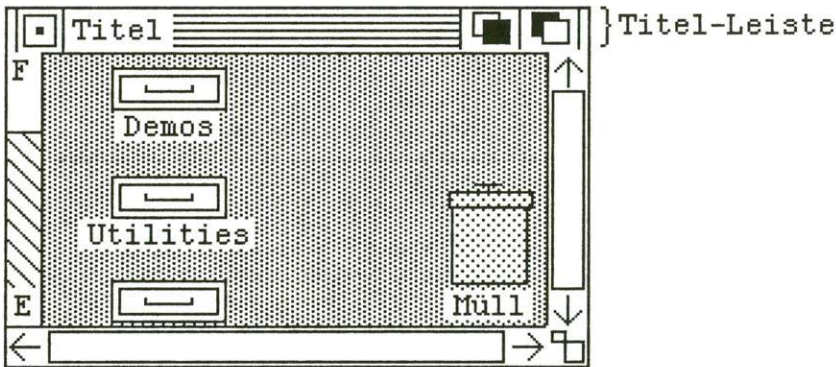


Bild 2 – 8: *Ein Fenster*

Wenn Sie jetzt mit der linken Maustaste auf den Namen oder die vier blauen Linien daneben klicken, ergreifen Sie das Fenster — genauso, wie Sie ein Disketten-Icon ergreifen können, indem Sie darauf klicken. Halten Sie die Maustaste fest und bewegen Sie die Maus. Den Mausbewegungen folgt jetzt ein orangefarbenes Rechteck, welches den Rahmen des Fensters symbolisiert. Lassen Sie die Maustaste in irgendeinem Augenblick los, so springt auch der Rest des Fensters samt Inhalt an die Stelle, an der sich der orange Rahmen in diesem Augenblick befand.

Experimentieren Sie etwas mit dem Bewegen von Fenstern. Es ist eine Sache, die in fast jedem Programm häufig gebraucht wird. Achten Sie dabei z.B. darauf, daß die Umrißlinie des Fensters immer komplett auf dem Bildschirm bleibt. Sie stößt bei der Bewegung am Bildschirmrand an und auch der Mauszeiger kann dann in dieser Richtung nicht weiter bewegt werden. Fenster müssen immer komplett auf dem Bildschirm sichtbar sein.

2.4.2 Überlappende Fenster

Eine der wichtigsten Eigenschaften der Fenster ist die Fähigkeit, sich zu überlappen. Ein Fenster kann ein anderes ganz oder teilweise verdecken, wie ein Blatt Papier auf dem Schreibtisch ein anderes Blatt teilweise verdecken kann. Wendet sich das Interesse wieder einem (teilweise) verdeckten Fenster zu, so zieht man es auf dem Bildschirm einfach zur Seite, damit wieder mehr oder alles davon sichtbar wird.

Um das ausprobieren zu können, brauchen wir natürlich ein zweites Fenster. Dazu klicken wir im Fenster *Workbench* das Schubladen-Symbol an, unter dem der Name **Demos** steht (oben links), und wählen dann wieder den Befehl **Open** aus dem Menü **Workbench** aus. Daraufhin erscheint ein neues kleines Fenster am unteren Bildschirmrand. Schubladen enthalten — wie es sich gehört — andere Dinge, die sichtbar werden, wenn man sie öffnet. Was das Fenster enthält, welches aus der Schublade **Demos** hervorgegangen ist, soll uns zunächst nicht interessieren. Wir benötigen nur das Fenster selbst. Um noch ein Fenster zu erhalten, öffnen Sie jetzt bitte auch die Schublade **Utilities**.

Nehmen Sie nun abwechselnd die beiden Fenster und schieben Sie sie auf dem Bildschirm hin und her. Sie sehen dabei sehr deutlich, daß jedes dieser Fenster ein "Blatt für sich" ist. Die Symbole des Fensterinhalts, die verdeckt wurden, erscheinen sofort wieder, sobald der entsprechende Teil des Fensters wieder freigelegt wird. Spinnen wir den Vergleich von Fenstern mit Blättern Papier auf einem Schreibtisch etwas weiter, so müßte es auch bei Fenstern eine Möglichkeit geben, eines ganz oben auf "den Stapel" oder "nach hinten" zu legen. Bis jetzt können Sie Fenster ja nur gegeneinander verschieben. Aber auch die Reihenfolge der Fenster "im Stapel" kann auf dem Amiga geändert werden.

2.4.3 Fenster-Gadgets

Zum Ändern der Reihenfolge der Fenster "im Stapel" dienen die beiden kleinen Symbole am rechten Rand der Titel-Leiste. Es sind sogenannte **Gadgets** (zu deutsch etwa *Geräte* oder *Instrumente*), die, ähnlich wie eine Menü-Auswahl, einen Befehl an ein Programm schicken, wenn sie angeklickt werden. Das linke der beiden Symbole heißt *Back-Gadget* und legt ein Fenster "ganz nach hinten", das rechte *Front-Gadget* und kann ein Fenster "ganz nach oben" legen.

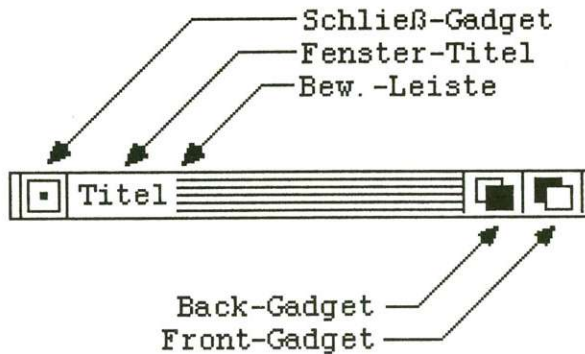


Bild 2 - 9: Die Titel-Leiste eines Fensters

Verteilen Sie die drei Fenster nun so auf dem Bildschirm, daß sie sich gegenseitig überdecken, aber von jedem wenigstens ein "Zipfelchen" der Titel-Leiste zu sehen ist. Nun können Sie die Gadgets in Ruhe ausprobieren. Klicken Sie in des Back-Gadget und das Fenster fällt nach hinten. Alle Teile anderer Fenster, die von diesem verdeckt wurden, werden wieder sichtbar. Ist das Front-Gadget eines teilweise verdeckten Fensters sichtbar, so klicken Sie darauf und können beobachten, wie es ganz nach oben kommt und sein kompletter Inhalt sichtbar wird.

Interessiert Sie der Inhalt eines Fensters überhaupt nicht mehr, so können Sie das Fenster ganz schließen. Hierzu können Sie natürlich den Befehl **Close** verwenden, der jedes Fenster wieder schließen kann, das mit **Open** geöffnet wurde. Es gibt allerdings auch eine "Abkürzung", die Ihnen den Weg zur Menü-Zeile spart. Klicken Sie in das kleine quadratische Gadget am linken Rand der Titel-Zeile eines Fensters, so wirkt dies, als hätten Sie das Fenster selektiert und dann den Befehl **Close** ausgewählt. Dieses Gadget heißt **Schließ-Gadget (Close Gadget)**.

Allen Gadgets, die Sie bis jetzt kennengelernt haben, ist eine Eigenschaft gemeinsam, die Sie sofort austesten können. Diese Gadgets bewirken nämlich nur dann etwas, wenn Sie die linke Maustaste niederdrücken, während sich der Mauszeiger über dem Gadget befindet, und die Taste dann aber auch wieder loslassen. Probieren Sie es einfach aus. Klicken Sie z.B. in ein Front-Gadget und halten die Aktionstaste fest, können Sie sehen, wie sich die Färbung des Gadgets ändert. Bewegen Sie den Mauszeiger bei gedrückter Taste vom Gadget fort, erhält es die ursprüngliche Färbung wieder. Wenn Sie jetzt die Taste loslassen, geschieht gar nichts. Bewegen Sie aber vor dem Lösen der Taste die Maus wieder über das Gadget, so schlägt dessen Farbe wieder um und wenn Sie jetzt die Taste loslassen, "wirkt das Gadget" und das Fenster kommt nach oben.

Somit haben Sie jetzt mehrere übereinanderliegende Blätter auf dem Bildschirm, von denen jedes fast so groß werden kann, wie ein ganzer Bildschirm und von denen Sie beliebig eines nach oben holen oder auch ganz beiseite legen (schließen) können. Die Behauptung, mit Fenstern könne man mehr Informationen auf dem Bildschirm zeigen, als eigentlich Platz haben, klingt damit also weniger absurd.

Fenster können noch eine ganze Menge anderer Eigenschaften haben. Für dieses Kapitel haben wir uns aber lange genug mit Fenstern beschäftigt. Zum Abschluß des Kapitels sollen Sie jetzt noch eine ganz elementare Operation kennenlernen, die jeder Computerbesitzer gar nicht oft genug durchführen kann.

2.5 Das Anlegen von Sicherungskopien

Eine Sache, die jeder Anwender eines Computers möglichst zu Anfang seines Kontakts mit einem neuen Modell erlernen sollte, ist das Anfertigen von *Sicherungskopien*. Die 3,5-Zoll-Disketten des Amiga sind zwar ein relativ robustes Speichermedium — aber bekanntlich hält nichts ewig. Bereits eine kleine Unachtsamkeit Ihrerseits oder ein kleiner Materialfehler können ihr schaden. Unter den Begriff Unachtsamkeit würde z.B. fallen, wenn Sie eine Diskette auf einen Lautsprecher, ein Gerät mit einem starken Elektromotor, unter ein Telefon oder auf die Heizung legen. Am gefährlichsten sind alle Ablagen, bei denen die Diskette einem starken magnetischen Feld ausgesetzt wird (wie es Elektromotoren und Telefonklingeln z.B. erzeugen). Aber auch die Alterung kann eine Rolle spielen.

Die möglichen Schäden, die dabei auftreten, können unter Umständen in kleinem Rahmen bleiben. Sie könnten z.B. nur ein Dokument betreffen, das Sie auf einer Diskette gespeichert haben, es könnte aber auch eine ganze Diskette unbrauchbar werden, so daß alle Daten darauf verloren gehen. Wenn Sie dann keine Sicherungskopie dieser Diskette haben, ist möglicherweise die Arbeit von Wochen verloren. Deshalb gilt die Grundregel Nummer 1:

Man kann gar nicht oft genug Kopien seiner wichtigen Disketten anlegen!

Dies gilt in dieser Form natürlich nur für Disketten, bei denen sich die darauf gespeicherten Daten andauernd ändern. Eine Diskette, wie die Kickstart- oder Workbench-Diskette, die Sie nie oder nur sehr selten ändern, braucht nur einmal bzw. nur einmal nach jeder schwerwiegenden Änderung kopiert zu

werden. Üblicherweise verfährt man dabei so, daß man nach dem Kopieren das Original an einer sicheren Stelle unterbringt (am besten so, daß auch neugierige Kinder nicht daran kommen können) und mit der Kopie arbeitet. Man spricht in diesem Zusammenhang oft auch von einer Arbeitskopie.

Von besonders wichtigen oder schwer wiederzubeschaffenden Disketten sollten Sie sogar mindesten 2 Kopien anlegen.

2.5.1 Vorbereitungen

Bevor Sie mit dem eigentlichen Kopieren anfangen, müssen Sie zunächst einige Vorbereitungen treffen. Legen Sie die Diskette bereit, die Sie kopieren wollen, die *Original-Diskette*, und sehen Sie nach, ob sie eindeutig beschriftet ist. *Eindeutig* heißt in diesem Fall, daß die Beschriftung den Inhalt deutlich beschreibt und sie nicht mit einer anderen Diskette verwechselt werden kann. Falls dies nicht der Fall ist, holen Sie die eindeutige Beschriftung nach — z.B. mit einem permanenten Faserschreiber.

Aktivieren Sie den Schreibschutz der Original-Diskette! Schieben Sie dazu den kleinen roten Schreibschutz-Schieber zum Rand der Diskette, so daß Sie durch das entstehende Loch hindurch schauen können. (Der Schreibschutz-Schieber befindet sich in der rechten, oberen Ecke der Diskette, wenn Sie auf die Diskettenoberseite schauen und der Metallschieber nach unten zeigt.) Dieser Schreibschutz macht es physikalisch unmöglich, daß bei einem Fehler beim Kopieren irgend etwas am Original zerstört wird.



Bild 2 - 10: Eine 3,5-Zoll-Diskette

Legen Sie dann die Diskette bereit, die die Kopie aufnehmen soll. Es kann eine neue Diskette frisch aus der Packung sein oder eine schon gebrauchte, deren Inhalt nicht mehr wichtig ist. Versehen Sie sie mit exakt derselben Beschriftung, wie die Original-Diskette und setzen sie noch groß **KOPIE** dahinter oder **BACKUP**, wenn Sie professionell wirken wollen. **Achten Sie darauf, daß der Schreibschutz der Kopie deaktiviert ist!** Der Schreibschutz-Schieber muß in einer Position stehen, die das kleine Loch in der entsprechenden Diskettenecke verdeckt.

Nun sind Sie bestens gerüstet, für "die Dinge die da kommen mögen". Sind sie den Anweisungen exakt gefolgt, so können Sie keinen Fehler machen, der ihre wertvolle Original-Diskette beschädigt.

Die Anfertigung einer Sicherungskopie läuft auf der Workbench ab. Starten Sie deshalb nun die Workbench, wenn Sie sich noch nicht auf ihr befinden. Falls Sie sich bereits auf der Workbench befinden, schließen Sie alle Fenster (und beenden Sie alle Programme, sofern Sie welche gestartet haben). Der genaue Ablauf des Kopiervorgang hängt nun stark davon ab, ob Sie ein oder zwei Diskettenlaufwerke besitzen. Naturgemäß ist es mit einem zweiten Laufwerk einfacher. Vielleicht ist das ein Anreiz für Sie, sich ein solches anzuschaffen.

2.5.2 Kopieren mit einem Diskettenlaufwerk

Besitzen Sie nur eine Diskettenlaufwerk, so steht Ihnen eine kleine Tortur bevor. Das Kopieren mit einem Diskettenlaufwerk ist kein reines Vergnügen — besonders wenn Ihr Amiga nur 256 KByte Speicher besitzt.

Falls Sie nicht die Workbench-Diskette selbst kopieren wollen, werfen Sie sie jetzt erst einmal aus. Halten Sie sie aber bereit; sie wird gleich wieder gebraucht. Schieben Sie die Diskette ein, die Sie kopieren wollen und warten Sie, bis das entsprechende Symbol am Bildschirm erscheint.

Selektieren Sie nun die Original-Diskette (mit der rechten Maustaste) und drücken Sie dann die Menütaste der Maus. Gehen Sie mit dem Mauszeiger über das Wort **Workbench** in der Menüleiste und fahren Sie dann — bei immer noch gedrückter Taste — über den Befehl **Duplicate**. (Falls dieser grau ist und nicht aktiviert werden kann, ist die Diskette nicht mehr selektiert. Klicken Sie erneut darauf.) Lassen Sie die Maustaste los, wenn das Wort **Duplicate** orange auf schwarz erscheint.

Falls Sie nicht die Workbench-Diskette kopieren, erscheint jetzt ein Fenster am oberen linken Bildschirmrand mit der Aufforderung, die Workbench-

Diskette wieder einzulegen. Werfen Sie in diesem Fall die zu kopierende Diskette aus und schieben Sie statt dessen die Workbench-Diskette ein. Das Fenster verschwindet daraufhin und das Diskettenlaufwerk arbeitet einen Moment lang.

Es erscheint nun ein neues Fenster namens **Disk Copy** mit der Aufforderung, die *SOURCE-Disk*, das ist das Original, ins Diskettenlaufwerk einzulegen. Tun Sie dies jetzt! (Falls die Workbench-Diskette noch im Laufwerk war, müssen Sie diese natürlich zuvor entfernen.) Am unteren Rand des Fensters **Disk Copy** befinden sich zwei Druckknöpfe — sogenannte Gadgets — in Form von schwarz und orange umrahmten Worten. Diese Gadgets verhalten sich genauso wie Tasten an einem Radio oder Kassettenrecorder. Das eine lautet **Continue** (engl. für *Fortfahren*), das ander **Cancel** (engl. für *Abbrechen*). Wenn Sie sich sicher sind, daß die richtige Diskette im Laufwerk liegt, klicken Sie auf **Continue**. Falls nicht, so klicken Sie bitte auf **Cancel**, was den gesamten Kopiervorgang abbricht.

Haben Sie **Continue** geklickt, so startet der Kopiervorgang. Im Fenster **Disk Copy** wird der Fortschritt des Vorgangs angezeigt. Der Amiga liest nun vom Inhalt der Original-Diskette so viel wie möglich in seinen RAM-Speicher. Ist dieser voll, so werden Sie gebeten, die *DESTINATION disk*, die Kopie, einzulegen. Warten Sie einen Moment und achten Sie darauf, daß das Diskettenlicht erloschen ist. Entfernen Sie dann das Original aus dem Laufwerk und schieben die Diskette ein, die die Kopie aufnehmen soll. Wenn Sie damit fertig sind, klicken Sie wieder auf **Continue**!

Der Amiga schreibt nun die Daten von der Original-Diskette, die er in seinem RAM gespeichert hat, auf die Kopie. Wenn er damit fertig ist, fordert er Sie wieder auf, die Original-Diskette (*SOURCE disk*) einzulegen, damit er das nächste Häppchen davon lesen kann. Folgen Sie dieser Aufforderung und klicken dann wieder auf **Continue**!

Dieser Diskettentausch wird noch einige Male nötig werden. Beim Amiga mit 256KByte ca. 9 oder 10-mal beim Amiga mit 512KByte 3-mal. Achten Sie dabei immer auf den Text im Fenster. Wird die *SOURCE disk* verlangt, müssen Sie das Original einlegen. Wird die *DESTINATION disk* verlangt, müssen Sie die Kopie einlegen. Achten Sie beim Herausnehmen der Disketten immer darauf, daß das rote Diskettenlämpchen aus ist und das Laufwerk keine hörbaren Geräusche von sich gibt. Wenn Sie mit dem Diskettenwechsel fertig sind, müssen Sie jedesmal auf **Continue** klicken (oder auf **Cancel**, wenn Sie die Tortur abbrechen wollen).

2.5.3 Kopieren mit zwei Diskettenlaufwerken

Diskettenkopieren mit zwei Laufwerken ist wesentlich angenehmer. Legen Sie dazu die Original-Diskette ins interne Laufwerk und die Diskette, die die Kopie aufnehmen soll, ins externe Laufwerk. Klicken Sie dann mit der linken Maustaste auf das Icon der Original-Diskette und ziehen den Zeiger dann über das Icon der Kopie. Nun lassen Sie die Maustaste los. Dieses Legen der einen Diskette auf die andere sagt der Workbench, daß Sie eine Kopie mit zwei Laufwerken machen wollen.

Falls Sie nicht die Workbench-Diskette selbst kopieren, erscheint jetzt ein Fenster mit der Aufforderung, die Workbench einzulegen. Werfen Sie dazu die Original-Diskette aus und legen Sie die Workbench-Diskette ein. Nach wenigen Augenblicken verschwindet die Aufforderung und ein neues Fenster namens **Disk Copy** erscheint.

In diesem Fenster erscheint unter anderem die Aufforderung, das Original (*SOURCE disk*) ins interne Laufwerk (*drive 0*) und die Kopie (*DESTINATION disk*) ins externe Laufwerk (*drive 1*) einzulegen. Tun Sie das jetzt und vergewissern Sie sich auf alle Fälle noch einmal davon, daß in jedem Laufwerk die richtige Diskette liegt! (Falls die Workbench-Diskette noch im Laufwerk war, müssen Sie diese natürlich zuvor entfernen.) Am unteren Rand des Fensters **Disk Copy** befinden sich zwei schwarz und orange umrahmte Worte, die Druckknöpfe — sogenannte Gadgets — darstellen, und die sich genauso verhalten, wie eine Taste an einem Radio oder Kassettenrecorder. Das eine Wort lautet **Continue** (engl. für *Fortfahren*), das andere **Cancel** (engl. für *Abbrechen*). Wenn Sie sicher sind, daß in jedem Laufwerk die richtige Diskette liegt, klicken Sie auf **Continue**. Falls nicht, so klicken Sie bitte auf **Cancel**, was den gesamten Kopiervorgang abbricht.

Nachdem Sie auf **Continue** geklickt haben, beginnt der Kopiervorgang. Er dauert insgesamt etwa 100 Sekunden. Sie können den Fortschritt im Fenster **Disk Copy** beobachten, um ungefähr zu wissen, wie lange es noch dauert.

2.5.4 Abschluss des Kopiervorgangs

Am Ende des Ganzen steht dann, egal ob Sie mit einem oder zwei Laufwerken, mit 256KByte RAM-Speicher oder 512KByte-RAM-Speicher arbeiten, eine komplette Kopie Ihres kostbaren Originals. Der alte Inhalt der Diskette, die die Kopie enthält, wurde dabei allerdings vollständig zerstört! Die Kopie erscheint übrigens auch unter dem Namen **copy of**

<Originalname> auf der Workbench. Sie können diesen Namen mit einem Befehl, den Sie in einem der folgenden Kapitel noch kennenlernen werden, auch ändern, wenn Sie es wünschen.

Falls es sich beim Original um die Kickstart-, Workbench- oder die Original-Diskette eines Programms handelt, legen Sie die Diskette beiseite und arbeiten Sie mit der Kopie weiter. Handelt es sich um eine Daten-Diskette, deren Inhalt häufig geändert wird, legen Sie die Kopie beiseite und arbeiten Sie mit dem Original weiter.

Es gibt allerdings inzwischen bereits einige kommerzielle Programme, die Sie auf diese Weise nicht kopieren können. Diese Disketten haben einen Kopierschutz, der sie vor unauthorisiertem (und leider auch vor rechtmäßigem) Kopieren schützt. Es ist traurig, daß Software-Hersteller meinen, zu solchen Methoden greifen zu müssen, aber man kann vorläufig wenig dagegen tun — außer solche Programme zu boykottieren! In Zukunft ist jedoch damit zu rechnen, daß auch Spezial-Kopierprogramme auf den Markt kommen, denen solche Kopierschutzverfahren nichts ausmachen. Ein gutes Kopierprogramm dieser Art gehört zur Grundausstattung jeder Softwarebibliothek, denn auch nur vorübergehender erzwungener Verzicht auf ein wichtiges Programm kann unter Umständen große Umstände (und Kosten) bereiten.

3

Objekte der Workbench

Nachdem wir im Kapitel 2 einen Überblick über die Funktionsweise der Amiga-Werkbank (Workbench) bekommen haben, wollen wir nun "zur Sache" kommen und die Objekte der Workbench näher beschreiben. Dies sind die verschiedenen Symbole, die auf dem blauen Bildschirmhintergrund und in den Fenstern der Workbench zu sehen sind. Wir nennen Sie *Icons*, und jedes von ihnen repräsentiert ein Objekt, das von der Workbench aus auf irgendeine Weise manipuliert werden kann. Diese Icons entsprechen den Disketten, Programmen und Dateien, mit denen Sie beim Umgang mit dem Amiga täglich zu tun haben werden. Besonders wichtig sind auch noch die Schubladen, die Ihnen helfen, Ordnung auf ihren Disketten zu halten.

Die im folgenden beschriebenen Befehle und Operationen sollten Sie bei einem ersten Lesen dieses Buches am besten immer sofort ausprobieren. Dazu sollten Sie am besten eine Kopie und nicht die Workbench-Diskette verwenden. Wie Sie eine solche Kopie anlegen, haben Sie ja im letzten Kapitel erfahren.

3.1 Disketten

Das erste Icon, welches Sie auf der Workbench kennengelernt haben, war das einer Diskette. Die folgende Abbildung zeigt es Ihnen noch einmal in etwas vergrößerter Form.



Bild 3 - 1: *Ein Disketten-Icon*

3.1.1 Auswählen und Verschieben von Disketten

"Normalerweise" erscheint ein Disketten-Icon weiß auf blauem Grund mit einem orangen Klecks, der den Metallschieber einer 3,5-Zoll-Diskette andeuten soll, am oberen Rand. Klicken Sie es aber mit dem linken Maustaste an, so wird es schwarz und der orange Klecks wird blau. Die Workbench zeigt auf diese Weise an, daß das Icon ausgewählt wurde und eventuell von dem nächsten Menü-Befehl beeinflusst wird, den Sie auswählen. Alle Icons der Workbench reagieren in dieser Weise mit einer Farbänderung auf eine Selektion mit der Maus.

Halten Sie die Maustaste fest, nachdem Sie das Icon angeklickt haben, so können Sie es bewegen. Dazu bewegen sie einfach den Mauszeiger zu der gewünschten Stelle und lassen die Taste wieder los. Aber dies kennen Sie ja schon aus Kapitel 2. Diesen Vorgang des "Ergreifens" und Verschiebens eines Icons mit der Maus nennen wir in Zukunft *Legen eines Icons* (an eine bestimmte Stelle). Die Mausbewegungen und Maustastendrucke, die dazu nötig sind, werden wir nicht jedesmal beschreiben.

Das Disketten-Icon bleibt so lange an der Stelle liegen, an der Sie es zuletzt hingelegt haben, bis Sie die Workbench verlassen. Das Verlassen der Workbench ist normalerweise nur durch ein Abschalten oder Neustarten (siehe Kapitel 2) des Amiga zu erreichen. Nach dem Abschalten "vergißt" die Diskette ihre neue Position. Beim nächsten Start der Workbench liegt sie wieder an ihrer alten Position (üblicherweise oben rechts am Bildschirm). Möchten Sie aber, daß eine bestimmte Diskette stets an einer anderen Position erscheint, so können Sie auch dies erreichen.

Legen Sie die Diskette dazu mit der Maus an die gewünschte Position und wählen dann den Befehl *Snapshot* (engl. für *Schnappschuß*) aus dem Menü *Special*. Dieser Befehl speichert die aktuelle Position des Disketten-Icons auf der Diskette ab — sofern sie nicht schreibgeschützt ist. Auch beim nächsten Start des Amiga erscheint das Disketten-Symbol dann an dieser neuen, von Ihnen gewählten, Position.

Falls die Diskette schreibgeschützt ist, wenn **Snapshot** aufgerufen wird, erscheint ein Fenster, in dem Ihnen dieser Fehler mitgeteilt wird. Dieses Fenster hat am unteren Rand zwei Knöpfe, die Sie mit der Maus anklicken können. Klicken Sie auf den Knopf **Cancel**, so wird der **Snapshot**-Befehl in der Annahme, daß Sie ihn versehentlich erteilt haben, abgebrochen. Sie haben aber auch die Möglichkeit, die Diskette aus dem Laufwerk zu nehmen, den Schreibschutz zu entfernen, die Diskette wieder einzulegen und dann auf den Knopf **Retry** (engl. für *versuche es noch einmal*) zu klicken. Der Schnappschuß wird dann auf der jetzt ungesicherten Diskette gespeichert. Im allgemeinen hat es aber guten Grund, eine Diskette mit Schreibschutz zu versehen. Sie sollten sich das Entfernen des Schreibschutzes deshalb gut überlegen!

Neben dem Selektieren und Bewegen eines Disketten-Icons sind natürlich noch eine ganze Menge anderer Operationen mit Disketten möglich. Fast alle davon werden dadurch ausgelöst, daß die Diskette zunächst mit der Maus selektiert wird und ihr dann ein Befehl geschickt wird, indem man einen der möglichen Befehle aus einem Menü auswählt. Einen wichtigen Befehl, der unter Umständen auch durch ein Bewegen der Diskette (auf eine andere) ausgelöst werden kann, haben wir ja bereits im Abschnitt 2.5 kennengelernt.

3.1.2 Das Diskettenfenster

Einer der wichtigsten Befehle, die eine Diskette "versteht", ist der Befehl **Open** (*öffnen*), den wir ja ebenfalls schon im letzten Kapitel kennengelernt haben. Auch andere Icons kennen den Befehl **Open** und zeigen daraufhin immer "was sich in ihnen verbirgt". Auf den Befehl **Open** hin öffnet sich am Bildschirm ein neues Fenster, in dem der Disketteninhalt dargestellt wird. Er besteht immer aus anderen Objekten (Icons), die wir weiter unten in diesem Kapitel noch näher kennenlernen werden. **Open** hat übrigens auch eine Abkürzung, die so praktisch ist, daß ich die Prophezeiung wage, daß Sie später fast nur noch diese Abkürzung wählen werden und nie mehr den Menü-Befehl **Open**. Klicken Sie einfach zweimal schnell hintereinander mit der linken Maustaste auf ein Disketten-Icon und Sie erzielen dieselbe Wirkung wie beim Auswählen der Diskette und der folgenden Wahl des Befehls **Open** aus dem **Workbench**-Menü. Diesen Vorgang werden wir im folgenden immer *Doppelklick* nennen.

Jedes Icon (also auch Disketten-Icons) kann entweder mit dem Menü-Befehl **Open** oder durch einen Doppelklick mit der linken Maustaste darauf geöffnet werden!

Den Aufbau und die möglichen Operationen mit einem Fenster haben wir zum großen Teil ebenfalls schon im letzten Kapitel kennengelernt. Am linken Rand eines jeden Disketten-Fensters ist eine Füllanzeige angebracht. Sie besteht aus einem schmalen senkrechten Balken, deren unterer Teil orange und deren oberer schwarz ist. Am unteren Ende des Balkens steht ein *E* für *empty* (leer), am oberen Ende ein *F* für *full* (voll). Der orange Teil kennzeichnet den Füllungsgrad der Diskette. Nimmt er die Hälfte des Balkens ein, so ist die Diskette halb voll; ist der ganze Balken orange, kann die Diskette keine neuen Daten mehr aufnehmen. Sie haben mit diesem Balken immer einen ungefähren Überblick über den "Füllgrad" einer Diskette. Benötigen Sie genauere Daten, so können Sie dies mit einem anderen Befehl herausfinden, der weiter unten noch erläutert wird.

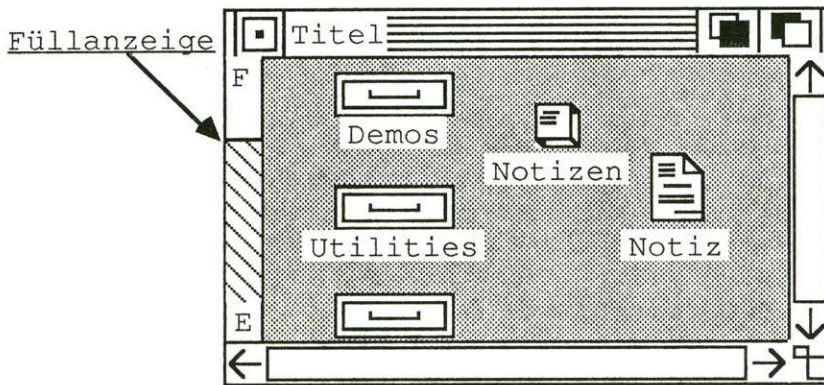


Bild 3 - 2: *Die Disketten-Füllanzeige*

Das Fenster einer Diskette können Sie jederzeit schließen, indem Sie das Disketten-Icon anklicken und dann den Befehl *Close* aus dem Menü *Workbench* auswählen. Falls Sie sich den "Weg ins Menü" sparen wollen, können Sie auch einfach ins Schließ-Gadget, links oben in der Titel-Leiste des Fensters klicken. Auch dadurch wird es wieder geschlossen.

Wir kommen nun zu zwei weiteren sehr wichtigen Eigenschaft von Fenstern, die auch im vorigen Kapitel noch nicht erwähnt wurden. Diese Eigenschaften werden vor allem dann wichtig, wenn Disketten-Fenster sehr viele andere Objekte enthalten.

Genauer gesagt handelt es sich um zwei neue Arten von Gadgets, die nun beschrieben werden sollen. Das erste davon ist das *Größen-Gadget*. Es ist das kleine Quadrat unten rechts im Rahmen eines Fensters mit einem kleinen und einem größeren Rechteck darin. Wie der Name schon sagt, können Sie damit die Größe eines Fensters ändern.

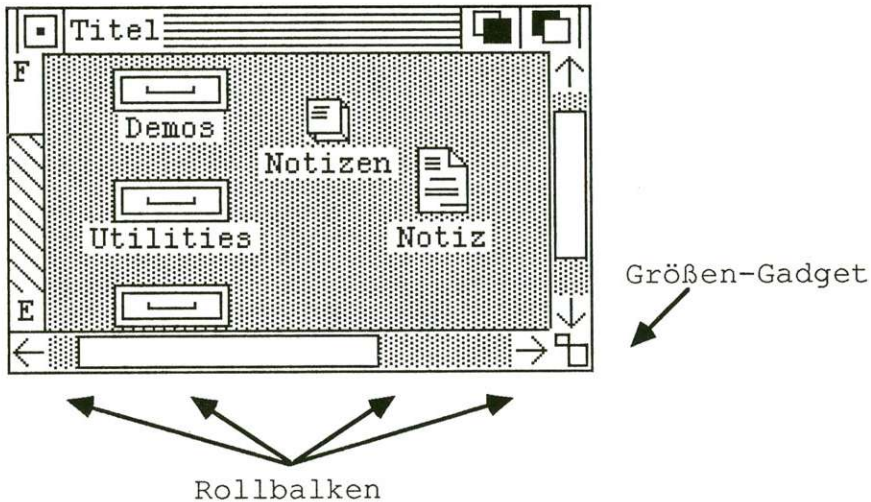


Bild 3 - 3: Rollbalken und das Größen-Gadget

Wenn Sie das Größen-Gadget mit der Maus ergreifen (d.h. mit der linken Maustaste darauf klicken und die Taste festhalten), so folgt eine rechteckige Umrißlinie den Mausbewegungen. Das sieht ähnlich aus wie beim Verschieben eines Fensters. Diesmal bleibt aber die obere linke Ecke unverändert und nur die untere rechte Ecke folgt (innerhalb gewisser Grenzen) den Mausbewegungen. Sobald die Maustaste losgelassen wird, erhält das Fenster seine neue Größe. Seine neuer Umriß paßt danach genau in die Umrißlinie hinein, die sichtbar war, kurz bevor die Maustaste losgelassen wurde.

Enthält ein Fenster nur sehr wenige Objekte, kann man es auf diese Art und Weise wesentlich platzsparender auf dem Bildschirm unterbringen. Liegen andererseits aber sehr viele Objekte darin, so kann man es entsprechend vergrößern. Was aber, wenn die Objekte darin mehr Platz beanspruchen, als das Fenster groß ist? In diesem Fall kann innerhalb des Fensters ja nur ein Teil der Fläche gezeigt werden, die eigentlich nötig wäre, um den ganzen Inhalt zu zeigen. Für solche Fälle gibt es die *Rollbalken*.

Um die Rollbalken auszutesten, öffnen Sie einfach ein Fenster und machen es dann mit dem Größen-Gadget so klein, daß einige der zuvor sichtbaren Objekte verschwinden.

Diese Objekte existieren natürlich nach wie vor. Sie sind nur nicht sichtbar. Man kann sich dies veranschaulichen, indem man sich vorstellt, daß alle Objekte, die innerhalb eines Fensters liegen, sich auf einer potentiell sehr großen Fläche verteilen, die Platz für alle Objekte bietet. Im Fenster selbst

kann aber nur ein (eventuell kleiner) Ausschnitt dieser Fläche gezeigt werden. Mit den Rollbalken kann man diesen Ausschnitt, der im Fenster sichtbar ist, auf verschiedene Arten ändern.

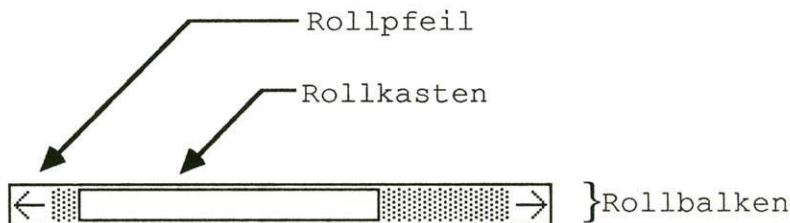


Bild 3 - 4: *Anatomie eines Rollbalkens*

Was zuerst auffällt, wenn man ein Fenster kleiner macht "als nötig wäre", ist, daß die beiden Streifen am rechten und unteren Rand des Fensters, dies sind die Rollbalken, ihr Aussehen ändern. Sie tragen an den beiden Enden zwei Pfeile, die in entgegengesetzte Richtungen zeigen. Die Fläche zwischen den Pfeilen ist leer und weiß, solange das Fenster "groß genug" ist. Sobald es aber zu klein wird, wird dieses weiße Rechteck ebenfalls kleiner und gibt eine dahinterliegende blaue Fläche frei. Die blaue Fläche symbolisiert dabei das Gesamtbild, von dem ein Teil innerhalb des Fensters gezeigt wird, das weiße Rechteck entspricht dem gerade sichtbaren Anteil. Dies ist der sogenannte *Rollkasten* (übrigens ebenfalls ein Gadget). Wenn der sichtbare Ausschnitt kleiner wird, verkleinert sich der Rollkasten entsprechend. Dies ist auch der Grund dafür, daß der Rollkasten bei einem Fenster, in dem das gesamte Bild sichtbar ist, kaum auffällt. Er füllt dann nämlich den gesamten Raum zwischen den beiden Pfeilen an, weil ja auch das gesamte Bild zu sehen ist. Aber auch, wenn sich der sichtbare Teil verschiebt, verschiebt sich der Rollkasten.

Die beiden Pfeile an den Enden jedes Rollbalkens sind die *Roll-Pfeile*. Klickt man auf einen davon, so bewirkt dies, daß der im Fenster sichtbare Ausschnitt des Gesamtbildes ein Stück in die angegebene Richtung "rutscht". Hält man bei diesem Mausklick die Shift-Taste¹ fest, so rückt der Ausschnitt nur um einen Punkt weiter, ansonsten ungefähr um eine halbe Fenster-Breite bzw. -Höhe. Bei diesem Verrücken rückt auch der Rollkasten in die angegebene Richtung, um zu zeigen, daß jetzt ein anderer Ausschnitt sichtbar ist.

¹ Diese Taste schaltet normalerweise zwischen Klein- und Großschreibung von Buchstaben um.

Der Rollkasten in einem Rollbalken kann aber nicht nur zur Anzeige, sondern auch zur direkten Manipulation des sichtbaren Ausschnitts verwendet werden. Er läßt sich wie ein Icon mit der Maus ergreifen und innerhalb des Rollbalkens (also nur horizontal oder nur vertikal) verschieben. Läßt man ihn (bzw. die Maustaste) los, "springt" der Ausschnitt zur angegebenen Stelle im Gesamt-Bild.

Die Rollbalken am rechten und unteren Rand eines Fensters verschieben den Ausschnitt vertikal und horizontal, verhalten sich aber ansonsten vollkommen gleich. Zusammen erlauben sie es, den sichtbaren Ausschnitt über einen beliebigen Bereich der "darunterliegenden" Fläche zu verschieben.

3.1.3 Aufräumen einer Diskette

Die Objekte, die im Disketten-Fenster gezeigt werden, können Sie mit der Maus an jede beliebige Stelle schieben. Dadurch kann der Fensterinhalt manchmal etwas "unaufgeräumt" erscheinen. Falls Sie ein bißchen Ordnung im Disketten-Fenster schaffen wollen, können Sie den Befehl **Clean up** (engl. für Aufräumen) verwenden. Wählen Sie dazu das Disketten-Icon aus (es reicht nicht, das Disketten-Fenster anzuklicken!) und wählen Sie dann **Clean up** aus dem Menü Special. Die Icons werden dann vom Amiga selbst in "Reih und Glied" gelegt.

Die Ergebnisse des Clean up-Befehls sind allerdings nicht immer so, wie man sich die Ordnung auf seiner Diskette vorstellt. Dies liegt unter anderem daran, daß die Icons verschieden groß sein können. Sind Sie mit der Ergebnis des Aufräumens nicht zufrieden, bleibt Ihnen nichts anderes übrig, als die Objekte selbst "von Hand" so zu legen, wie Sie es für richtig halten.

3.1.4 Umbenennen einer Diskette

Unter jedem Icon, das Sie am Bildschirm sehen, steht auch ein Name. Dies dient vor allem dazu, Icons mit dem selben Aussehen unterscheiden zu können. Den Namen, bzw. den Text, der unter dem Icon erscheint, können Sie selbst bestimmen. Selektieren Sie dazu z.B. ein Disketten-Icon und wählen dann den Befehl **Rename** (Umbenennen) aus dem Menü **Workbench**. Mitten auf dem Bildschirm erscheint daraufhin ein schmaler, weiß umrandeter Streifen, in dem der Name der Diskette steht (z.B. "Workbench"). Klicken Sie mit der linken Maustaste in diesen Streifen und Sie können den Text ändern.

Der erste Buchstabe des Namens erscheint zunächst einmal schwarz auf orange, während alle anderen Buchstaben weiß auf blau gedruckt sind. Der schwarze Buchstabe markiert die aktuelle Schreibmarke (*Cursor*). Wenn Sie auf der Tastatur nun zu tippen beginnen, erscheint der neue Text immer vor dieser Schreibmarke und die dahinterstehenden Buchstaben werden automatisch nach rechts geschoben. Sie können den Buchstaben, auf dem die Schreibmarke steht, auch löschen. Drücken Sie hierzu die -Taste oben rechts auf der Tastatur. Der Buchstabe bei der Schreibmarke verschwindet und die Buchstaben rechts davon rutschen dadurch nach links. Wollen Sie einen völlig neuen Namen eingeben, so tippen Sie einfach ein paarmal auf bis der alte Name völlig verschwunden ist und geben dann den neuen ein.

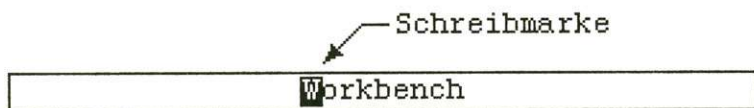


Bild 3 - 5: *Ein Rahmen für die Texteingabe*

Wenn Sie einen Tippfehler machen, können Sie immer den letzten eingegebenen Buchstaben mit der <BACKSPACE>-Taste (unter der -Taste) löschen. Die <BACKSPACE>-Taste löscht aber nicht nur den zuletzt eingetippten Buchstaben, sondern immer den Buchstaben links von der Schreibmarke, so daß Sie also auch mehrere Buchstaben auf diese Weise wieder löschen können.

Rechts unten auf der Tastatur befinden sich weiterhin vier Tasten, die einen Pfeil tragen. Dies sind die Cursortasten, die Sie auch zum Verschieben der Schreibmarke verwenden können. Die beiden nach links und rechts weisenden Pfeile bewegen z.B. die Schreibmarke jeweils einen Buchstaben nach links bzw. rechts, sofern dies möglich ist. Die beiden Pfeile nach oben und unten haben im Moment keine Wirkung.

Sind Sie mit dem Namen in dem weißen Rahmen zufrieden, so tippen sie kurz auf die <RETURN>-Taste, und das zuvor ausgewählte Disketten-Icon

erhält einen neuen Namen.² Die <RETURN>-Taste schließt typischerweise immer die Eingabe von Text in einen solchen Rahmen ab! Sie sollten sich das für später merken. Auch andere Befehle oder Programme verlangen manchmal Texteingaben, die immer ähnlich ablaufen.

3.1.5 Detaillierte Informationen über eine Diskette

Die Füllanzeige und der Name einer Diskette geben Ihnen ja bereits erste Informationen über ihren Zustand. Detaillierte Informationen darüber können Sie mit dem Befehl **Info** im Menü **Workbench** erhalten. Selektieren Sie dazu die Diskette, über die Sie mehr Informationen erhalten wollen, und wählen Sie dann den Befehl **Info** aus.

Daraufhin erscheint ein neues Fenster am Bildschirm, das den Namen **Info release 1.1** trägt.

² Falls die Diskette schreibgeschützt war, erhält das Icon keinen neuen Namen, und ein Fenster mit einer Fehlermeldung erscheint. Da der Name auf der Diskette gespeichert wird, verhindert ein Schreibschutz auch das Umbenennen. Das Fehlermeldungs-Fenster zeigt Ihnen zwei Knöpfe, auf die Sie mit dem Mauszeiger drücken können. Bei **Cancel** wird der **Rename**-Befehl abgebrochen und die Diskette behält den alten Namen. Sie können aber auch die Disketten aus dem Laufwerk nehmen, den Schreibschutz entfernen und dann auf **Retry** (engl. für *versuche es noch einmal*) klicken. Dann erhält die Diskette den neuen Namen.

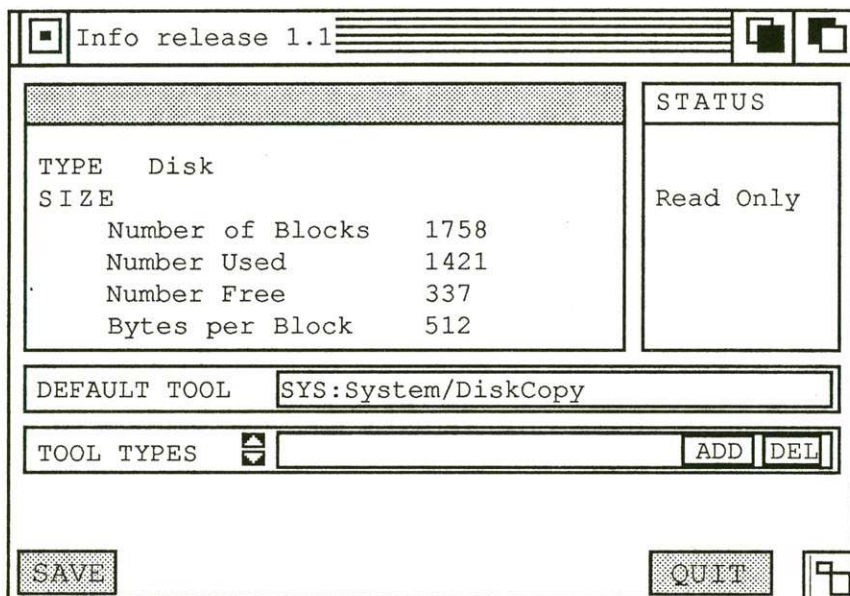


Bild 3 - 6: *Das Info-Fenster einer Diskette*

In diesem Fenster erfahren Sie unter anderem den Namen der Diskette (er wird auch auf der Workbench gezeigt), den Typ (**TYPE Disk**) und — unter der Typangabe — einen Überblick über die exakte Nutzung der Diskette. Hinter **Number of Blocks** erfahren Sie, in wieviele physikalische Blocks die Diskette aufgeteilt ist (bei 3,5-Zoll-Disketten immer 1758). Darunter finden Sie die Angaben darüber (hinter **Number Used**) wieviele davon belegt und (hinter **Number Free**) noch frei sind. Die letzte Angabe schließlich (**Bytes per Block**) sagt Ihnen, wieviele Buchstaben bzw. Bytes in jedem der Blöcke Platz haben, von denen oben die Rede war. Bei einer Diskette sind es üblicherweise 512; falls Sie Besitzer einer Festplatte sind, kann hier aber durchaus eine andere Zahl stehen. (Die Diskette kann also maximal etwas über 900 000 Buchstaben aufnehmen.)

Neben diesem Feld mit den vielen Zahlen befindet sich das Feld **Status**. Falls die Diskette schreibgeschützt ist, steht darin **Read Only** (*nur lesen*) ansonsten immer **Read/Write** (*lesen und schreiben*). Die beiden Felder darunter (**Default Tool** und **Tool Types**) sollen uns vorläufig nicht interessieren. Um diese zu modifizieren, benötigen Sie ein tieferes Wissen über das System, als Ihnen dieses Buch bis jetzt vermitteln konnte.

Das Info-Fenster können Sie wieder verschwinden lassen, indem Sie sein Schließ-Gadget in der oberen linken Ecke oder den Knopf **Quit** in der rechten unteren Ecke des Fensters mit der Maus anklicken. Den Knopf **Save** sollten

Sie vorläufig nicht betätigen. Er ist nur sinnvoll, wenn Sie **Default Tool** oder **Tool Type** modifiziert haben (was Sie nicht tun sollten).

3.1.6 Initialisieren und Löschen von Disketten

Bevor eine neue Diskette verwendet werden kann, muß sie zunächst einmal *initialisiert* werden. Dieser Vorgang teilt die zunächst unstrukturierte magnetische Oberfläche in einzelnen Sektoren bzw. Blöcke zu 512 Byte auf, über die ja z.B. auch der Info-Befehl Auskunft gibt. Auch wird dabei Platz für das Dateiverzeichnis angelegt, eine Art Landkarte, die den Programmen sagt, wo welche Daten auf der Diskette zu finden sind.

Der Befehl **Initialize** im Menü **Disk** initialisiert eine Diskette, die Sie zuvor mit der Maus selektiert haben. Dies kann in zwei Fällen nötig werden — bei neuen Disketten, die Sie nicht als Kopie einer anderen Diskette verwenden wollen und bei bereits verwendeten Disketten, deren Inhalt Sie (vielleicht aus Geheimhaltungsgründen) löschen wollen.

Legen Sie eine neue, noch nicht initialisierte Diskette in ein Laufwerk ein, so erscheint sie auf der Workbench als ein Icon mit dem Zusatz **BAD** (engl. für *schlecht/unbrauchbar*). Sie können ein solches Icon auswählen und mit dem Befehl **Initialize** eine brauchbare Diskette erhalten. Genauso können Sie eine bereits verwendete Diskette löschen. **Das Initialisieren zerstört alle Daten, die sich eventuell vorher auf einer Diskette befanden!**

Bevor der Vorgang beginnt, werden Sie in einem speziell zu diesem Zweck erscheinenden Fenster sicherheitshalber gefragt, ob Sie sich Ihrer Sache sicher sind. Nur wenn Sie in diesem Fenster auf den Knopf **Continue** drücken, wird die selektierte Diskette initialisiert und damit gelöscht. Klicken Sie aber auf den Knopf **Cancel**, wird der Vorgang abgebrochen. Sie können sich weiterhin vor Fehlern schützen, indem Sie den Schreibschutz wichtiger Disketten aktivieren. Schreibgeschützte Disketten können nicht initialisiert werden!

Wollen Sie eine komplette Kopie einer anderen Diskette auf einer zweiten Diskette ablegen, brauchen Sie sie vorher nicht zu initialisieren! Das geschieht beim Kopieren einer ganzen Diskette automatisch "nebenbei". Nur wenn Sie eine komplett leere Diskette benötigen, um darauf Daten abzulegen, müssen Sie diese vorher initialisieren.

3.2 Schubladen

Man kann sagen, die Schubladen sind die wichtigsten Objekte, die sich auf der Werkbank befinden. Wir haben die Schubladen ja schon kurz in Kapitel 2 kennengelernt, als wir Fenster zum Experimentieren brauchten. In Schubladen verbergen sich — ganz ähnlich wie in Disketten-Icons — immer andere Objekte. Die Schubladen sind an sich keine "richtigen" Objekte; sie enthalten nur andere.³

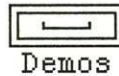


Bild 3 - 7: *Eine Schublade der Workbench*

Schubladen verhalten sich sehr ähnlich wie Disketten. Fast alle Befehle, die Sie von Disketten-Icons und Diskettenfenstern kennen, funktionieren in entsprechender Weise auch bei Schubladen. Sie werden deshalb hier nur sehr knapp beschrieben werden.

3.2.1 Aufgabe von Schubladen

Schubladen dienen der Gliederung der Informationen bzw. Objekte auf einer Diskette. Prinzipiell wäre es z.B. möglich, alle Objekte, die eine Diskette enthält, im Diskettenfenster selbst unterzubringen. Solange es 10 oder 20 Objekte sind, mag das völlig ausreichen. Wenn aber einige hundert oder (bei Verwendung einer Festplatte) gar einige tausend Objekte auf einer Diskette (bzw. Festplatte) liegen, wird man rasch die Übersicht verlieren, und nur sehr schwer das Objekt finden, welches man gerade sucht.

Auf einer richtigen Werkbank würden wir ja auch nicht alle Gegenstände einfach herumliegen lassen, sondern sie in Kisten, Kästchen und Schubladen legen, die jeweils eine zusammengehörige Gruppe von Gegenständen enthält. Eine Kiste für die Bohrer, eine für Nägel, Schrauben, Dübel, Schrauben - schlüssel, Schraubendreher, etc. Genauso funktionieren die Schubladen der

³ Schubladen entsprechen auf einer nicht ganz so bildlichen Ebene den sogenannten Dateiverzeichnissen oder auf neudeutsch *directories*. Dies sind Konzepte, die in Computerkreisen schon lange bekannt (und beliebt!) sind. Mehr darüber erfahren Sie im Kapitel *Dateien und Dateiverzeichnisse*.

Workbench. Wenn Sie alle persönlichen Briefe, die Sie mit einem Textverarbeitungssystem erstellt haben, in eine Schublade legen, alle Rechnungen in eine andere und die kleinen Hilfsprogramme, die Sie nicht täglich benötigen, in eine andere, so wird das Ihre Chancen, einen bestimmten Brief auf der Diskette zu finden, wesentlich erhöhen.

Aber wie Schubladen auf einer Werkbank ihrerseits wieder Kisten und Kästchen enthalten können, die vielleicht jeweils Nägel einer bestimmten Größe enthalten, können auch die Schubladen auf der elektronischen Workbench weitere Schubladen enthalten. Haben Sie z.B. eine Schublade mit Briefen, so könnten Sie in dieser wiederum Schubladen einrichten, die ältere, neue, und noch nicht abgesandte Briefe enthalten. Verwalten Sie sehr viele Briefe elektronisch auf einer Diskette, so dürfte dies wiederum Ihre Chancen, einen bestimmten zu finden, wesentlich verbessern — wie schon zuvor die Entscheidung, Brief und Rechnungen in getrennte Schubladen zu legen.

Während Sie auf einer richtigen Werkbank nicht unendlich viele Schubladen ineinander verschachteln können, geht dies im Prinzip jedoch bei den elektronischen Schubladen auf der Workbench. Sie können Schubladen in Schubladen in Schubladen in Schubladen haben. (Vielleicht kennen Sie ja diese russischen Püppchen, die man öffnen kann, und dann darin wieder ein Püppchen findet, das man öffnen kann usw. Genauso kann man die Schubladen der Amiga-Workbench anwenden.) Wie sinnvoll das in solch extremer Form ist, darüber läßt sich allerdings streiten. Müssen Sie, um etwa einen Brief bearbeiten zu können, zuerst 20 Schubladen öffnen, so ist die Arbeitserleichterung, die Ihnen Schubladen verschaffen können, vielleicht gar nicht mehr so groß. Wie immer, muß man auch hierbei das richtige Mittelmaß finden. Eine maximale Schachteltiefe von fünf bis sechs Schubladen scheint aber durchaus sinnvoll zu sein.

3.2.2 Umgang mit Schubladen

Wie bereits oben erwähnt, verhalten sich Schubladen "im wesentlichen" wie Disketten. Die Befehle aus dem Menü **Workbench** z.B. verhalten sich bei Schubladen alle genauso wie bei Disketten.

Durch den Befehl **Open** oder einen Doppelklick können Schubladen geöffnet, durch den Befehl **Close** oder einen Klick ins Schließ-Gadget ihres Fensters wieder geschlossen werden. Die Fenster, die sich öffnen, wenn man eine Schublade öffnet, sehen fast genauso aus und verhalten sich genauso, wie Disketten-Fenster. Ihnen fehlt nur die Füllanzeige am linken Rand. Schubladen kennen nämlich keinen Füllungsgrad. Sie können solange andere

Objekte aufnehmen, bis die Diskette oder Festplatte voll ist. Die Füllanzeige des Disketten-Fensters gilt deshalb gleichzeitig auch für alle Schubladen auf der Diskette.

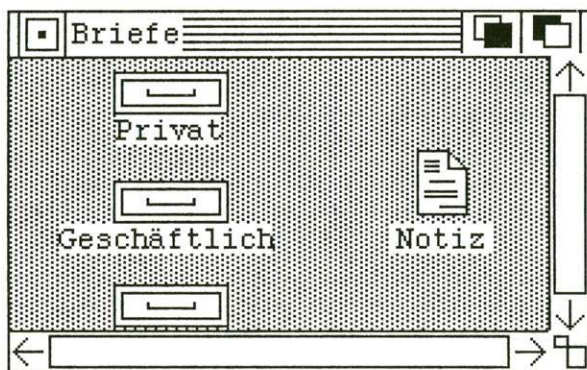


Bild 3 - 8: *Das Fenster einer Schublade*

Ein Tip übrigens noch, der ganz hilfreich sein kann, wenn Sie die Möglichkeit, Schubladen zu bilden, intensiv nutzen. Wenn allzu viele Fenster auf Ihrer Workbench liegen und Sie ein bestimmtes nicht mehr finden können, weil es irgendwo im Stapel versteckt ist, können Sie dieses zum Vorschein bringen, wenn Sie zufällig die Schublade sehen können, aus der das Fenster hervorgegangen ist. Öffnen Sie einfach die Schublade noch einmal (z.B. mit einem Doppelklick) und das entsprechenden Fenster erscheint oben auf dem Stapel. Ein kleiner Trick, mit dem man sich manchmal viel Arbeit erspart. (Wie Sie sehen hat die elektronische Werkbank in diesem Fall deutliche Vorteile gegenüber einer wirklichen. Bei einer echten Werkbank können Sie eine schon offene Schublade natürlich nicht noch einmal öffnen.)

Mit dem Befehl **Rename** können Schubladen, genau wie Disketten, umbenannt werden. Mit dem Befehl **Info** erhalten Sie detailliertere Informationen über eine Schublade. Das Fenster, in dem Ihnen diese Informationen dargeboten werden, sieht allerdings ein klein wenig anders aus als das einer Diskette.

Info release 1.1

NAME Brief-Schublade	STATUS
TYPE Drawer	DELETEABLE
COMMENT Ein Kommentar	
TOOL TYPES	ADD DEL
SAVE	QUIT

Bild 3 - 9: *Das Info-Fenster einer Schublade*

Das Info-Fenster zeigt für eine Schublade im Gegensatz zu einer Diskette nicht den Speicherbedarf der Objekte in der Schublade an. Dafür haben Sie jedoch die Möglichkeit einen zusätzlichen Kommentar zu vermerken, der Ihnen später hilft, wenn Sie einmal nicht mehr genau wissen, was Sie nun denn in die Schublade gelegt haben. Klicken Sie zum Anbringen eines Kommentars einfach in das Rechteck hinter **COMMENT** (engl. für *Kommentar*). Sie können in diesem Feld dann einen Namen eintippen und müssen die Eingabe mit der <RETURN>-Taste beenden. Zur Fehlerkorrektur innerhalb des Rechtecks stehen Ihnen übrigens genau dieselben Möglichkeiten zur Verfügung wie beim Befehl **Rename** (s.o.).

Sie können eine Schublade auch vor unbeabsichtigtem Löschen (s.u.) schützen. Klicken Sie dazu einfach im Feld **STATUS** das Rechteck an, in dem das Wort **DELETEABLE** (löscher) steht. An dieser Stelle erscheint nun **NOT DELETEABLE** (nicht löscher). Versuchen Sie nun, diese Schublade zu löschen, so erhalten Sie nur eine Fehlermeldung — die Schublade selbst bleibt stehen. Bevor sie wirklich gelöscht werden kann, muß mit dem **Info**-Befehl dann zunächst der "Schalter" unter **STATUS** wieder auf **DELETEABLE** umgelegt werden. Dieser Schutz für Schubladen ist allerdings nicht allzu sicher. Er verhindert z.B. nicht ein Löschen der gesamten Diskette (und damit natürlich auch dieser Schublade) mit dem

Initialize-Befehl. Wirklich wichtige Disketten sollten Sie deshalb immer mit dem Schreibschutz der Disketten sichern.

Beim Info-Befehl für Disketten war erwähnt worden, daß es sicherer ist, das Fenster mit dem Schließ-Gadget wieder zu schließen oder auf den Knopf **QUIT** unten rechts in der Ecke zu drücken. Für Schubladen gilt unter Umständen etwas anderes. Wenn Sie nämlich den Kommentar einer Schublade geändert oder den Schalter im Kasten **STATUS** umgelegt haben, so bleibt dies ohne dauerhafte Wirkung, wenn Sie das Schließ-Gadget oder den Knopf **QUIT** drücken. Nur wenn Sie den Knopf **SAVE** drücken, werden Ihre Änderungen auch wirklich gespeichert und berücksichtigt.⁴

Mit dem Befehl **Duplicate** aus dem **Workbench**-Menü können Sie ganze Schubladen mit allem, was sich darin befindet, duplizieren. Die neue Schublade erscheint danach neben der alten und erhält zunächst einmal den Namen *copy of <alter Name>* (den man natürlich mit **Rename** ändern kann).

Sie können Schubladen, genau wie Disketten-Icons auch, mit der Maus bewegen. Diese Bewegung — nicht nur von Schubladen — erhält durch Schubladen aber eine neue Qualität. Wenn Sie ein Objekt (Icon) bewegen und die Maustaste loslassen, während der Mauszeiger sich über dem Icon einer Schublade oder in einem anderen Fenster befindet, so wird das Objekt in diese Schublade gelegt. Auf diese Weise kann man Objekte aus einer Schublade (oder aus dem Disketten-Fenster) in eine andere Schublade legen.⁵

Dieses Umlegen von Objekten von einer Schublade in eine andere hat allerdings noch eine Besonderheit, je nachdem, ob die beteiligten Schubladen sich auf verschiedenen Disketten befinden oder nicht. Liegen beide an einem solchen Vorgang beteiligten Schubladen auf ein und derselben Diskette, läuft alles so ab, wie oben beschrieben. Das Objekt verschwindet aus der einen Schublade und erscheint in der anderen.

Befindet sich die zweite Schublade aber auf einer anderen Diskette, so bleibt das verlegte Objekt in der alten Schublade, und in der Schublade auf der zweiten Diskette erscheint eine **Kopie des Original-Objektes**. Das

⁴ Auch bei Schubladen sollten Sie, genau wie bei Disketten, den Inhalt des Kastens hinter **TOOL TYPES** nicht ändern. Um mit dieser Möglichkeit umzugehen, müßten Sie über weit mehr Informationen verfügen, als dieses Buch Ihnen im hier zur Verfügung stehenden Platz vermitteln kann.

⁵ Wegen eines Fehlers der Workbench-Software erscheint ein Objekt nicht im geöffneten Fenster einer Schublade, wenn man es in das entsprechende Schubladen-Icon legt. Damit es auch im Fenster sichtbar wird, muß man dieses Fenster einmal schließen und dann wieder öffnen.

Verlegen eines Objektes in eine Schublade auf einer zweiten Diskette ist also eine Möglichkeit, eine Kopie dieses Objektes auf einer anderen Diskette anzulegen. Dies ist meist sinnvoller als das Kopieren einer gesamten Diskette.

Beim Verlegen von Objekten gelten übrigens auch Disketten als "eine Art" Schubladen! Man kann deshalb auch Objekte in ein Disketten-Icon oder in ein Disketten-Fenster mit dem selben Effekt verlegen, als hätte man es in eine Schublade auf der selben Diskette gelegt. Mehr darüber und über die Unterschiede und Ähnlichkeiten zwischen Disketten und Schubladen im Abschnitt *Spezielle Schubladen*.

3.2.3 Erzeugen neuer Schubladen

Anders als bei einer wirklichen Werkbank können sie der Workbench des Amiga jederzeit neue Schubladen und Schubladen in Schubladen hinzufügen. Das Erzeugen neuer Schubladen läuft dabei üblicherweise immer über das Vervielfältigen bereits vorhandener Schubladen ab. Dies ist ganz einfach. Wie bereits oben erwähnt, können Sie Schubladen duplizieren, indem Sie sie auswählen und dann den Befehl **Duplicate** aus dem **Workbench**-Menü aufrufen. Bei Schubladen bewirkt dieser Befehl das Anlegen einer Kopie der ursprünglichen Schublade und ihres gesamten Inhalts. Auf diese Art und Weise entsteht natürlich eine neue Schublade — eben die Kopie.

Benötigen Sie aber eine neue leere Schublade, um darin z.B. einen Teil der Objekte aus einer anderen Schublade unterzubringen, so müssen Sie auch eine leere Schublade duplizieren. Auf der Standard-Workbench-Diskette ist deshalb eine leere Schublade untergebracht worden, die genau für diesen Zweck gedacht ist. (Sie trägt den Namen *Empty*; zu deutsch *leer*.)

Wenn Sie in irgendeiner Schublade eine neue Unterschublade unterbringen wollen, müssen Sie deshalb diese leere Schublade im Disketten-Fenster der Workbench-Diskette auswählen und duplizieren. Mit dem Befehl **Rename** (s.o.) können Sie der Kopie danach den gewünschten Namen geben. Zuletzt können Sie sie dann in die Schublade legen, in der Sie eine neue Schublade benötigen.

3.2.4 Löschen von Schubladen

Wenn Sie täglich mit dem Amiga arbeiten, wird es mit Sicherheit gelegentlich nötig werden, auf Ihren Disketten "aufzuräumen". Damit ist natürlich unter anderem auch gemeint, daß alle Objekte ordentlich in Schubladen

untergebracht werden, damit man sie später auch wiederfindet. Ab und zu werden Sie aber auch Objekte oder Schubladen, die durch eine Reorganisation überflüssig geworden sind, löschen müssen.

Für das Löschen von Schubladen (und anderen Objekten) gibt es zwei Möglichkeiten. Einmal können Sie sie in den Mülleimer werfen, was in einem folgenden Abschnitt noch ausführlich erläutert wird. Der Vorteil des Mülleimers ist, daß Sie es sich auch "noch einmal anders überlegen können". Objekte, die im Mülleimer sind, sind ja auch in Wirklichkeit noch nicht zerstört oder verloren, sondern können wieder hervorgeholt werden.

Wenn man sich aber ganz sicher ist, eine Schublade für immer entbehren zu können, so kann man sie auch auswählen und dann den Befehl **Discard** aus dem Menü **Workbench** auswählen. Tut man dies, so wird man in einem zu diesem Zweck erscheinenden Fenster noch einmal gefragt, ob man sich über die Konsequenzen (unwiederbringlicher Verlust der Schublade und aller Objekte darin) im klaren ist. Dieses Fenster besitzt am unteren Rand zwei Knöpfe, **Retry** (engl. für *versuch es noch einmal* oder *sinngemäßer Fortsetzen der begonnenen Aktion*) und **Cancel** (engl. für *abbrechen*). Drückt man (mit Mauszeiger und Maustaste) auf **Retry**, so wird der Löschvorgang begonnen, und danach gibt es wirklich keine Möglichkeit mehr, die so gelöschte Schublade oder die darin befindlichen Objekte wiederzubekommen. Drückt man aber auf den Knopf **Cancel**, wird der Vorgang abgebrochen und es geschieht nichts weiter.

Mit **Discard** kann man fast alle Objekte auf einer Diskette löschen. Das Löschen von Schubladen ist aber besonders gefährlich, da man oft nicht mehr genau weiß, was diese enthalten. **Vor dem Löschen einer Schublade (egal ob mit **Discard** oder über den Mülleimer) sollte man diese öffnen und sich vergewissern, daß nur Objekte in ihr liegen, die nicht mehr benötigt werden!**

Schubladen mit wichtigem Inhalt kann man auch (über den **Info**-Befehl) vor versehentlichem Löschen schützen.⁶ Dies ist aber kein sehr perfekter Schutz, da er nicht vor einem Öffnen der Schublade und dem Löschen der einzelnen Objekte darin schützt. Im allgemeinen ist es aber doch wohl so, daß nicht die Schublade, sondern ihr Inhalt wichtig ist.

⁶ Wie wir in dem Teil dieses Buches über den CLI, der zweiten Möglichkeit, den Amiga zu bedienen, noch sehen werden, können sogar Dateien in einer Schublade sein, die auf der Workbench gar nicht zu sehen sind. Auch diese würden gelöscht werden, wenn die ganze Schublade gelöscht wird!

3.2.5 Schnappschüsse und Schubladen

Wie Sie die Objekte in einem Schubladen-Fenster verteilen wollen, liegt ganz an Ihnen. Gefällt Ihnen eine Anordnung besonders gut, so können Sie die einzelnen Objekte in einem Fenster auch mit dem Befehl **Snapshot** (engl. für *Schnappschuß*) aus dem **Special**-Menü an diesen Stellen "festnageln", so daß sie auch nach einem Neustart wieder an dieser Stelle erscheinen. Diesen Befehl haben wir ja schon im Zusammenhang mit Disketten-Icons vorgestellt, wollen jetzt aber noch einmal näher auf seine "Feinheiten" eingehen.

Bei einem Snapshot wird z.B. nicht nur die Position eines Icons, sondern auch die Größe und Position der Fenster abgespeichert, die aus diesen Icons hervorgehen können. Immer wenn Sie Snapshot auf Schubladen anwenden, sollten Sie deshalb darauf achten, daß Ihnen auch die Position und Größe der Schubladen-Fenster gefallen.

Die Anwendung des **Snapshot**-Befehls ist aber sehr lästig, wenn Sie die Positionen mehrerer Schubladen oder anderer Objekte abspeichern wollen. Es nimmt immer einige Zeit in Anspruch, ein Objekt auszuwählen, den Befehl Snapshot aufzurufen und zu warten, bis sich das Diskettenlaufwerk wieder beruhigt hat. Aus diesem Grunde besteht auch die Möglichkeit, mehrere Objekte gleichzeitig für einen solchen Befehl auszuwählen. Der entsprechende Vorgang heißt *erweiterte Selektion*.

Klicken Sie hierzu zunächst das erste Icon an, von dem ein Snapshot gemacht werden soll. Es zeigt die Selektion durch eine Farbänderung an. Dann drücken Sie auf die Shift-Taste⁷ und klicken das nächste Icon an. Auch dieses zeigt nun durch Farbänderung an, daß es selektiert wurde, während das erste selektiert bleibt. (Normalerweise führt die Selektion eines Icons dazu, daß das zuletzt ausgewählte Icon deselektiert wird.) Fahren Sie mit dem Selektieren nun fort, bis alle Icons ausgewählt sind, von denen Sie einen Schnappschuß machen wollen. Dann wählen sie **Snapshot** aus, und die Positionen aller dieser Icons werden abgespeichert. Erweiterte Selektion ist übrigens nur mit Icons innerhalb eines Fensters möglich. Beim Anklicken eines Icons außerhalb des Fensters, in dem das zuerst angeklickte Icon lag, werden alle zuvor selektierten Icons deselektiert!

Bevor Sie einen Snapshot von einer Schublade und/oder ihrem Inhalt machen, ist es sinnvoll, erst einmal aufzuräumen. Hierzu können Sie den Befehl **Clean Up** aus dem Menü **Special** verwenden. Genau wie beim Aufräumen von Disketten-Fenstern legt dieser Befehl auch in

⁷ Diese Taste schaltet normalerweise zwischen Klein- und Großschreibung von Buchstaben um.

Schubladenfenstern die verschiedenen Objekte ordentlich neben- und untereinander. Sind alle Icons in einer Schublade von einer Sorte, so sieht das meist sehr gut aus und gibt ein wirklich ordentliches Bild. Bei einer inhomogenen Mischung von Icons werden Sie wahrscheinlich mit der Hand nachhelfen müssen.⁸

3.2.6 Spezielle Schubladen

Wie schon mehrfach erwähnt, gibt es neben den normalen Schubladen auch noch *spezielle* oder besondere *Schubladen*. Diese zerfallen in zwei Gruppen: Disketten und Mülleimer.

3.2.6.1 Disketten als Schubladen

In vieler Hinsicht können Disketten behandelt werden wie Schubladen. Sie können z.B. auf dieselbe Weise geöffnet und geschlossen, umbenannt und dupliziert werden. Noch wichtiger ist aber, daß Diskettenfenster und Disketten-Icons sich ähnlich verhalten wie Schubladenfenster und Schubladen-Icons.

Ergreift man z.B. ein Objekt mit der Maus und läßt es über einem Disketten-Icon fallen (indem man dort die Maustaste löst), so erscheint das Objekt im Fenster dieser Diskette (sobald dieses neu geöffnet wird). Man kann Objekte auch direkt in ein Diskettenfenster legen.

Befand sich das "alte Objekt" vor seiner Verlegung in einem Ordner, der sich seinerseits auf einer anderen Diskette befand, so findet auch beim Verlegen in ein Diskettenfenster oder -Icon ein Kopiervorgang statt. Auf der Diskette erscheint (falls genügend Platz darauf ist) eine Kopie des Objektes. Das Objekt selbst verbleibt in seiner "alten" Schublade.

In einer Hinsicht unterscheiden sich Disketten aber von Schubladen. Disketten können zwar Schubladen enthalten, umgekehrt gilt dies jedoch nicht. Versucht man, ein Disketten-Icon in ein anderes Fenster oder in ein Schubladen-Icon zu legen, so erscheint eine Fehlermeldung. Disketten können allerdings in andere Disketten-Icons gelegt werden, wodurch der

⁸ Falls der *Clean Up*-Befehl im Menü grau erscheint, wenn Sie ein Fenster aufräumen wollen, haben Sie wahrscheinlich vergessen, vorher das entsprechende Icon (entweder eine Diskette oder eine Schublade) zu diesem Fenster zu selektieren.

Inhalt dieser zweiten Diskette komplett durch den der bewegten Diskette ersetzt wird (vgl. Abschnitt 2.5).

3.2.6.1 Mülleimer

Ein Symbol, das Ihnen bestimmt sofort aufgefallen ist, nachdem Sie das erste Mal ein Disketten-Icon geöffnet haben, ist der Mülleimer (engl. *Trashcan*). Der Mülleimer dient zum Löschen anderer Objekte. Er verhält sich aber zum Teil wie eine "Schublade".

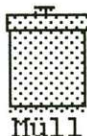


Bild 3 - 10: *Das Mülleimer-Icon*

Wollen Sie ein beliebiges Objekt auf der Workbench löschen, so nehmen Sie es einfach und legen es in den Mülleimer. Das geht genauso, wie das Verlegen eines Objektes in eine andere Schublade. Man ergreift das Objekt mit der Maus, bewegt dann den Mauszeiger über das Mülleimer-Icon und läßt die Maustaste los.

Wenn man ein Objekt in den Mülleimer gelegt hat, ist es noch nicht sofort verloren. Wie "in Wirklichkeit" liegt es zunächst einmal im Mülleimer und kann wieder daraus hervorgeholt werden — solange die Müllabfuhr noch nicht da war. Hierzu kann man den Mülleimer öffnen, wie jede andere Schublade auch. Daraufhin erscheint ein Schubladenfenster, in dem alle Objekte auftauchen, die man bisher in den Mülleimer geworfen hat. Wollen Sie eines davon nicht löschen, so können Sie es mit der Maus ergreifen und in eine andere Schublade legen.

Erst wenn die Müllabfuhr gekommen ist, sind die Objekte, die in den Mülleimer geworfen wurden, unwiederbringlich verloren. Auf dem Amiga können Sie selbst bestimmen, wann die Müllabfuhr kommt. Wenn Sie sich ganz sicher sind, die Objekte im Mülleimer nicht mehr zu benötigen, selektieren Sie ihn und wählen danach **Empty Trash** aus dem **Disk**-Menü. (Der Befehl ist nur dann möglich, wenn Sie wirklich zuvor ein Mülleimer-Icon selektiert haben.) Daraufhin werden die Objekte im Mülleimer wirklich von der Diskette gelöscht. Sie können dabei unter Umständen an der Füllanzeige des entsprechenden Disketten-Fensters beobachten, wie der

"Pegel" fällt, der noch zur Verfügung stehende Platz also wieder größer wird.⁹

Den Mülleimer sollten Sie der Verwendung des Befehls **Discard** im Zweifelsfall immer vorziehen. **Legen Sie alle Objekte, die sie löschen wollen, zunächst in den Mülleimer und leeren Sie diesen erst, wenn Sie dringend mehr Platz auf der entsprechenden Diskette benötigen!** Dies läßt Ihnen die Chance, die so "fortgeworfenen" Objekte bei einem Irrtum wieder zu holen.

Mülleimer sind auch insofern besondere Schubladen, weil man sie nicht aus dem Diskettenfenster entfernen kann. Sie können weder in eine andere Schublade noch in ein anderes Diskettenfenster gelegt werden. Auch können Mülleimer nicht mit dem Befehl **Discard** gelöscht werden. Besitzt eine Diskette erst einmal ein Mülleimer-Icon, so kann dieses nicht wieder entfernt werden!

Für Mülleimer gilt übrigens auch die oben erwähnte Regel, daß Disketten nicht in andere Ordner gelegt werden können. Disketten können also auch nicht dadurch gelöscht werden, daß man sie in den Mülleimer legt!¹⁰

3.3 Weitere Icons auf der Workbench

Alle Objekte, die wir bis jetzt kennengelernt haben, waren mehr oder weniger Ordnungshilfen für Disketten. Die Icons, die diese repräsentieren sind Disketten, Schubladen und Mülleimer. Es gibt aber auch noch andere Icons in den Fenstern z.B. der Workbench-Diskette und in ihren Schubladen.

⁹ Enthält der Mülleimer eines oder mehrere Objekte, die mit dem **Info**-Befehl vor Löschung geschützt wurden (s.o.), so ergibt **Empty Trash** eine Fehlermeldung und die geschützten Objekte werden auch nicht entfernt. Es kann aber durchaus sein, daß zu dem Zeitpunkt, wo der Amiga diesen Fehler bemerkt hat, bereits Objekte, die im Mülleimer lagen und nicht geschützt waren, gelöscht wurden!

¹⁰ Disketten können – sofern überhaupt nötig – gelöscht werden, indem man auf sie den Befehl **Initialize** aus dem Disk-Menü anwendet.

3.3.1 Werkzeuge und Projekte

Bis jetzt wurde ja immer sehr abstrakt von Objekten gesprochen, wenn alle möglichen Icons in einem Schubladenfenster gemeint waren. Disketten, Schubladen und Mülleimer haben wir bereits kennengelernt. Was aber steckt hinter den anderen Icons auf der Workbench? Damit wollen wir uns jetzt ein wenig beschäftigen.

Alle Objekte, die Sie in einem Disketten- oder Schubladen-Fenster sehen können, und bei denen es sich nicht um Schubladen oder Mülleimer handelt, sind *Werkzeuge* und *Projekte*.

Getreu der Analogie, daß die Bedienung des Amiga abläuft wie die Arbeit auf einer Werkbank (engl. *workbench*), heißen die Programme des Amiga *Werkzeuge* (engl. *tools*) und die Dateien, die diese Programme erzeugen oder für ihre Arbeit benötigen, heißen *Projekte* (engl. *projects*). Sie können diese Dateien auch als ein Dokument ansehen, z.B. als einen Brief oder eine Notiz. Jedes Werkzeug kann ein nahezu beliebiges Icon am Bildschirm zeigen¹¹, meist deutet das Icon aber in etwa die Funktion des Programms an. Eine Textverarbeitung wird also vielleicht wie eine kleine Schreibmaschine aussehen und ein Grafikprogramm wie eine Palette mit Pinsel.

Zwischen Werkzeugen und den Projekten, die man damit bearbeitet, besteht ein enger Zusammenhang. Meist kann eine Projekt nur mit einem ganz bestimmten Werkzeug bearbeitet werden. Eine Textverarbeitung kann mit Grafiken nichts anfangen und umgekehrt (Ausnahmen kommen, wo sinnvoll, natürlich vor). Dieser Zusammenhang wird meist dadurch angedeutet, daß die Icons von Projekten logisch zu den Werkzeug-Icons passen. Texte werden also z.B. wie ein Blatt Papier mit einem Eselsohr und ein paar angedeuteten Buchstaben aussehen und von der Aufmachung (z.B. den Farben) her dem Textverarbeitungsprogramm ähnlich sehen.

Wenn Sie ein solches Werkzeug jetzt gerne einmal sehen wollen, öffnen Sie einfach die Workbench-Diskette, und warten Sie einen Moment, bis das Diskettenlaufwerk ruhig wird. Oben im Fenster sind zwei Objekte erschienen, die weder wie Schubladen noch wie Disketten oder Mülleimer aussehen. Dies sind die beiden Programme (Werkzeuge) *Clock* und *Preferences*. *Clock* zeigt Ihnen z.B., wie der Name (und das Icon) schon sagt, am Bildschirm eine Uhr.

¹¹ Später werden Sie sogar erfahren, wie Sie selbst das Aussehen eines Icons verändern können.

Im Rest dieses Kapitels sollen Werkzeuge und Projekte aber nur aus der Sicht der Workbench betrachtet werden. Näher mit diesen Objekten beschäftigen wird sich das folgende Kapitel *Werkzeuge und Projekte*.

3.3.2 Operationen für Werkzeuge und Projekte

Auf der Workbench können Werkzeuge und Projekte im wesentlichen wie alle anderen Objekte behandelt werden. Die meisten Befehle funktionieren wie bei Schubladen.

Werkzeuge und Projekte können z.B. mit der Maus ergriffen und bewegt werden. Sie können sie in Schubladen legen, indem Sie sie in das Schubladen-Icon oder in das entsprechende Schubladenfenster legen (sofern eines offen ist). Werden sie in eine Schublade, die auf einer anderen Disketten ist, oder direkt in das entsprechende Diskettenfenster gelegt, so werden auch Werkzeuge und Projekte kopiert, wie es bei einer Schublade der Fall wäre.

Die Position eines Werkzeugs oder einer Schublade innerhalb "seines" Disketten- oder Schubladenfensters kann mit dem Befehl ***Snapshot*** im Menü ***Special*** abgespeichert werden.

Aus dem Menü ***Disk*** können keine Befehle angewendet werden, wenn zuvor ein Werkzeug oder Projekt selektiert wurde. Dafür sind fast alle Befehle des Menüs ***Workbench*** anwendbar. Der Befehl ***Open*** wird ausführlich im nächsten Kapitel erläutert (und sollte deshalb vorläufig nicht angewendet werden, wenn man ein Werkzeug oder Projekt selektiert hat).

Der Befehl ***Close*** kann nicht auf Werkzeuge oder Projekte angewendet werden. Mit dem Befehl ***Duplicate*** kann (auf derselben Diskette) eine Kopie eines Werkzeugs oder Projektes angelegt werden, die zunächst den Namen ***copy of <alter Name>*** trägt. Diesen Namen kann man danach mit dem Befehl ***Rename*** beliebig ändern.¹²

Mit dem Befehl ***Info*** bekommt man detailliertere Informationen über ein Werkzeug oder Projekt in einem neuen Fenster angezeigt, das ganz ähnlich aussieht wie das Info-Fenster von Schubladen. Näheres über diese Info-Fenster von Projekten und Werkzeugen erfahren Sie im nächsten Kapitel. Der Befehl ***Discard*** schließlich dient zum (unwiederruflichen) Löschen von Projekten oder Werkzeugen. Wie bei Löschen von Schubladen werden Sie vor der Ausführung des Befehls aber erst noch einmal in einem neuen Fenster

¹² Die Bedienung des ***Rename***-Befehls wurde relativ ausführlich bei der Erörterung von Disketten erläutert.

um Bestätigung gebeten. Klicken Sie in diesem Fenster auf den Knopf **Retry**, ist das Projekt oder das Werkzeug verloren. Sicherer ist es immer, ein Projekt oder Werkzeug, das gelöscht werden soll, erst einmal in den Mülleimer zu legen.

3.4 Alle Befehle der Workbench

Im folgenden finden Sie eine Liste aller Menü-Befehle, die Sie auf der Workbench aus den Menüs aussuchen können. Jeder Befehl wird zunächst auf englisch aufgelistet, dann eine ungefähre deutsche Übersetzung oder Umschreibung. Die Menü-Namen (fettgedruckt und unterstrichen) sind den einzelnen Befehlen (fettgedruckt) jeweils vorangestellt.

Workbench

Open	(<i>Öffnen</i> eines Objektes; z.B. einer Schublade)
Close	(<i>Schließen</i> eines Disketten- oder Schubladenfensters)
Duplicate	(<i>Kopieren</i> bzw. Duplizieren eines beliebigen Objektes)
Rename	(<i>Umbenennen</i> eines Objektes)
Info	(<i>Informations-Fenster</i> eines Objektes zeigen)
Discard	(Unwiderrufliches <i>Löschen</i> eines Objektes)

Disk

Empty Trash	Unwiderrufliches <i>Löschen</i> aller Objekte im Mülleimer)
Initialize	(<i>Vorbereiten</i> einer neuen Diskette zur Bearbeitung oder <i>Löschen</i> einer alten)

Special

Clean Up	(<i>Ordnen</i> der Objekte in einer Diskette oder Schublade)
Last Error	Die <i>letzte Fehlermeldung</i> noch einmal zeigen)
Redraw	(<i>Neuzeichnen</i> der Workbench)
Snapshot	(<i>Abspeichern</i> der aktuellen Position eines Objektes)
Version	(<i>Versions-Nummer</i> der Workbench-Software zeigen)

4

Werkzeuge und Projekte

Wenn Sie mit Ihrem Amiga wirklich etwas vollbringen wollen, so benötigen Sie dazu ein Programm. Dies ist bei keinem Computer anders und sollte inzwischen bereits ein recht vertrautes Prinzip für Sie sein. Auch bisher (in den ersten drei Kapiteln dieses Buches) haben wir uns ja bereits mit einem Programm beschäftigt, der Workbench. Die Workbench aber ist ein sogenanntes *Systemprogramm*, das von den meisten Anwendern als selbstverständliche Einrichtung mehr oder weniger übersehen wird. Spricht man von einem Programm, so meint man damit meist ein sogenanntes *Anwendungsprogramm*, wie z.B. eine Textverarbeitung oder ein Buchhaltungsprogramm. Erst die Anwendungsprogramme machen einen Computer zu einem wirklichen Hilfsmittel; ohne sie bleibt er ein teures Spielzeug.

4.1 Werkzeuge

Auf der Workbench des Amiga heißen Anwendungsprogramme *Werkzeuge* und genau das sollten gute Programme auch sein. Kein Programm leistet wirklich "von selbst" etwas. Es kann Ihnen aber, wie ein gutes Werkzeug, dabei helfen eine Aufgabe zu erledigen bzw. die Erledigung dieser Aufgabe erst ermöglichen.

4.1.1 Öffnen eines Werkzeugs

Im letzten Kapitel wurden die meisten Operationen mit Werkzeugen und Projekten bereits vorgestellt. Die eine Operation, die noch nicht erläutert wurde, war das Öffnen von Werkzeugen.

Bereits bei anderen Objekten der Workbench konnte man die Grundregel erkennen, daß das Öffnen eines Objektes immer zeigt, "was sich darin verbirgt". Um zu sehen, was sich in einem Programm verbirgt, muß man es natürlich starten – und genau so wirkt der Befehl **Open** auch, wenn Sie zuvor das Icon eines Werkzeugs selektiert haben.

Um ein Werkzeug (Programm) zu starten, selektieren Sie es einfach mit einem Mausklick und wählen dann den Befehl **Open** aus dem Menü **Workbench**. Alternativ können Sie dasselbe auch durch einen Doppelklick mit der Maus auf das Icon des Werkzeugs erreichen. Probieren Sie es aus!

Auf der Workbench-Diskette liegt direkt im Diskettenfenster ein Icon, das wie eine Uhr aussieht und auch den vielsagenden Namen *Clock (Uhr)* trägt. Dies ist ein Werkzeug. Starten Sie es nun mit einem Doppelklick auf dieses Icon oder über den Befehl **Open**. Das Diskettenlaufwerk beginnt nun sich zu drehen, liest das Programm von der Diskette in den RAM-Speicher des Amiga und aktiviert es dann.

Nach wenigen Sekunden erscheint ein neues Fenster links oben auf der Werkbank. Es trägt den Titel **Clock V1.93** und zeigt in seinem Innern ein wirklichkeitsgetreues Bild einer (Analog-)Uhr mit sich bewegendem Sekundenzeiger. Die Uhrzeit stimmt zwar wahrscheinlich nicht, ansonsten geht sie aber genau.¹ Sie können die Uhr einfach dort in der Ecke des Bildschirms liegen lassen und mit Ihrer Arbeit auf der Workbench fortfahren. Die Uhr läuft dabei weiter, wie es sich für eine ordentliche Uhr gehört, und zeigt Ihnen weiter die falsche Zeit genau an (wie es sich für eine ordentliche Uhr nicht gehört).

¹ Der Amiga hat standardmäßig zwar eine ziemlich genau gehende Uhr eingebaut aber keine Batterie, die dafür sorgt, daß diese Uhr auch dann noch weiterläuft, wenn Sie das Gerät abgeschaltet haben. Bei jedem Neustart ist es deshalb eigentlich nötig (und auch ratsam) die Uhr nachzustellen. Sie können dies mittels des Werkzeugs **Preferences** machen, das weiter unten noch erläutert wird.

4.1.2 Fenster eines Werkzeugs

Sobald Sie aber das Fenster der Uhr anklicken, z.B. um es zu bewegen, teilen Sie dem Programm mit, daß Sie damit arbeiten, bzw. direkt mit ihm kommunizieren wollen. Dies wird vor allem durch zwei Dinge auch am Bildschirm verdeutlicht. Zum einen wird das Fenster aktiv: der Titelstreifen besser lesbar (er ist gepunktelt und schlecht lesbar, solange das Fenster inaktiv ist). Zum zweiten ändert sich die Titel-Leiste des Bildschirms. Statt **Workbench 1.1** und der Angabe des freien Speicherplatzes steht dort nun auch **Clock 1.93**.

Sobald Sie Werkzeuge öffnen, wird es wichtig, ob ein Fenster aktiv ist oder nicht. Solange das Fenster eines Programms nämlich inaktiv ist, kann das Programm zwar tätig sein, erst wenn das Fenster aktiv wird, können Sie aber Einfluß auf das Programm nehmen. Da Sie sich nicht zerreißen können, können Sie ja immer nur mit einem Programm zur selben Zeit kommunizieren. Nur dieses eine Programm empfängt auch Ihre Aktionen (z.B. Mausklicks und Tastendrucke). Indem Sie auf das Fenster eines bestimmten Programms klicken (es also aktiv machen) schenken Sie diesem Programm Ihre Aufmerksamkeit. Das Programm, welches vorher aktiv war, geht dann in eine Art Wartezustand über, in dem es ohne Ihre Eingaben auskommen muß. Das neue Programm empfängt nun Ihre Aktionen und reagiert darauf. Wie es darauf reagiert, entscheidet es allein. Deshalb kann auch ein Mausklick oder eine Bewegung der Maus in einem Fenster etwas völlig anderes bewirken als in einem anderen Fenster.

Die Systemsoftware des Amiga speichert zu jedem Fenster, das am Bildschirm sichtbar ist, ab, zu welchem Programm es gehört. So kann sie jederzeit entscheiden, welches Programm die "Nachrichten", die vom Anwender kommen, empfangen und verarbeiten soll.

Experimentieren Sie doch ein wenig mit dem Uhr-Fenster. Sie werden sehen, daß es sich ganz ähnlich verhält, wie die Fenster der Workbench, die wir in den vorangegangenen Kapiteln schon kennengelernt haben. Das Uhr-Fenster enthält z.B. ein Größen-Gadget, Back- und Front-Gadgets und ein Schließ-Gadget, mit dem Sie das Fenster schließen und damit das Programm (Werkzeug) **Clock** wieder beenden können. Versuchen Sie vor allem einmal das Größen-Gadget. Die Uhr paßt sich jeder neuen Fenstergröße an. Wenn Sie wollen, können Sie eine Uhr in Bildschirmgröße haben. Wird das Fenster rechteckig, so wird das Ziffernblatt entsprechende oval. Unten links im Fenster liegt noch ein kleines Gadget, das ein G-Symbol zeigt. Klicken Sie darauf und Sie schalten den Sekundenzeiger der Uhr an und aus.

4.1.3 Menü-Leiste eines Werkzeugs

Wie oben bereits erwähnt, wechselt jedesmal, wenn man auf das Fenster eines anderen Programms klickt, der Inhalt der Titel-Leiste des Bildschirms. Diese Titel-Leiste enthält aber nicht einen kurzen Namen wie **Workbench** oder **Clock**, sondern eben auch die Menüs. Auch diese werden komplett ausgetauscht, wenn ein Fenster eines anderen Programms angeklickt wird.

Drücken Sie einmal die Menü-Taste, während in der Titel-Leiste **Clock** steht, und Sie werden drei neue Menü-Titel sehen: **Type**, **Mode** und **Alarm**.

Im Menü **Type** stehen Ihnen nur zwei Befehle zur Auswahl: **Digital** und **Analog**. Der Befehl **Analog** ist zusätzlich noch mit einem "Häkchen" versehen, der Ihnen sagt, daß die Uhr im Moment auf Analog-Anzeige geschaltet ist. Dadurch sehen Sie auch gleich eine weitere Möglichkeit in den Menüs: gibt es nämlich in einem Menü einen Befehl, der quasi ein Schalter ist, so kann die Stellung des Schalters durch ein Häkchen angezeigt werden. Ist der Schalter *an* (Analog-Anzeige sichtbar), wird der Befehl mit einem Haken versehen (*abgehakt*), ansonsten nicht. Indem Sie den anderen Befehl, **Digital**, im **Type**-Menü wählen, wechselt die Anzeige der Uhrzeit von Zeigern nach Ziffern. Danach ist **Digital** mit einem Häkchen versehen und **Analog** nicht mehr. Die Digitalanzeige hat allerdings ein Fenster von fester Größe, so daß es beim Wechsel von einer Anzeige-Art zur anderen meist auch zu einem Sprung in der Fenstergröße kommt, der etwas überraschend sein kann.

Im Menü **Mode** können Sie zwischen 24-Stunden-Anzeige und 12-Stunden-Anzeige für die Uhr wählen. Die Bedienung ist genauso wie bei **Type**. Der jeweils zuletzt gewählte — und damit gültige — Befehl wird mit einem Häkchen versehen.

Im letzten Menü **Alarm** können Sie die Uhr des Amiga schließlich auch noch als Wecker einsetzen. Unten in diesem Menü gibt es die beiden Punkte **Alarm On** und **Alarm Off**, von denen wieder der jeweils gültige mit einem Häkchen versehen ist. Wie sich das auch für einen richtigen Wecker gehört, klingelt auch der Wecker des Amiga auf keinen Fall, solange **Alarm Off** gültig (*abgehakt*) ist. Mit dem ersten Befehl **Set** in diesem Menü können Sie die Alarm- (bzw. Weck-) Zeit einstellen. Hierzu erscheint, nachdem Sie den Befehl aufgerufen haben, ein Fenster mitten auf dem Bildschirm.



Bild 4 - 1: *Einstellen der Alarm-Zeit*

In diesem Fenster sehen Sie unter anderem die Uhrzeit 12:00 Uhr in digitaler Form. Klicken Sie auf die angezeigten Stunden und Sie können mit den danebenliegenden Pfeilsymbolen die Stunden vor und zurücklaufen lassen. Klicken Sie auf die Minuten daneben und Sie können die Minuten der Weckzeit mit den beiden Pfeilen verstellen. Um das Fenster wieder verschwinden zu lassen, müssen Sie auf einen der Knöpfe *USE* (engl. für Verwenden) oder *CANCEL* (engl. für Abbrechen) klicken. Nur wenn Sie auf *USE* klicken, ist die eingestellte Weckzeit gültig, bei *CANCEL* vergißt der Amiga den ganzen Vorgang. Sie werden aber natürlich nur dann zur eingestellten Zeit "geweckt," wenn *Alarm On* im Menü *Alarm* eingestellt ist.

Die Amiga-Uhr ist nicht gerade ein überwältigendes, aber doch ein ganz nettes kleines Programm, für das Sie gelegentlich sicher Verwendung finden werden. Ihr einziger Haken (außer den vielen Häkchen in den Menüs) ist, daß Sie die richtige Zeit bei jeder Arbeitssitzung neu einstellen müssen. Wie das geht, erfahren Sie am Ende dieses Kapitels.

4.1.4 Info-Fenster eines Werkzeugs

Nachdem wir jetzt ja die Arbeitsweise eines Werkzeugs andeutungsweise kennengelernt haben, betrachten wir noch einmal kurz, was die Workbench über ein Werkzeug mitzuteilen hat. Die Funktion des *Info*-Befehls bei Werkzeugen verbleibt ja bis jetzt als einziger Workbench-Befehl noch ungeklärt.

Wählt man den Befehl *Info* aus dem Menü *Workbench*, wenn ein Werkzeug selektiert ist, so bekommt man ein Fenster gezeigt, wie es in etwa die folgende Abbildung zeigt.

The screenshot shows a window titled "Info release 1.1". Inside, there's a form for editing tool preferences. The main area is divided into two columns. The left column contains fields for "NAME" (set to "Preferences"), "TYPE" (set to "Tool"), "SIZE" (with sub-fields for "in bytes" at 51932 and "in blicks" at 107), and "STACK" (with a text input field containing "8000"). The right column has a "STATUS" field and a "DELETEABLE" checkbox. Below these columns is a "COMMENT" field containing the text "Ein Kommentar". At the bottom, there's a "TOOL TYPES" field with a dropdown arrow, followed by "ADD" and "DEL" buttons. At the very bottom are "SAVE" and "QUIT" buttons, along with a small window icon.

Bild 4 - 2: Das Info-Fenster eines Werkzeugs

Dieses Fenster dürfte im wesentlichen bereits von den anderen Workbench-Objekten her vertraut sein. Es enthält nur einige zusätzliche Komponenten. Links in diesem Fenster steht unter dem Namen der Objekt-Typ, in diesem Fall **Tool** (Werkzeug). Darunter steht die Größe in Byte (Buchstaben) und Blöcken (die jeweils 512 Byte enthalten). Ähnliche Angaben gibt es ja auch bei Disketten-Infos. Darunter befindet sich nun außerdem ein neues Feld namens Stack. Hinter dem Wort Stack können Sie eine Zahl eingeben, die bestimmt, wieviel von einer besonderen Sorte Speicherplatz das Werkzeug beim Start im RAM des Amiga reserviert bekommt. Diese Angabe ist für Sie erst dann interessant, wenn Sie ein wenig mehr von den Innereien des Amiga kennengelernt haben; sie sei aber der Vollständigkeit halber erwähnt.² Die anderen Felder im Info-Fenster sollten alle bereits vom Info-Fenster für Schubladen bekannt sein. Hinter **Comment** kann z.B. eine Bemerkung (Kommentar) zu diesem Werkzeug eingetragen werden und im Feld **Status** kann ein Werkzeug vor unbeabsichtigtem Löschen geschützt werden (indem

² Falls Sie bei der Anwendung eines Programms einmal eine Fehlermeldung bekommen, in der der Begriff *stack overflow* vorkommt, können Sie versuchen, diesen Wert zu ändern (also zu vergrößern). Normalerweise beträgt er 4000 (Byte), wird aber nicht angezeigt. Tragen Sie schrittweise größere Werte ein, und versuchen Sie den Vorgang, der zum Fehler führte, noch einmal.

Sie den "Schalter" `DELETEABLE` darin auf `NOT DELETEABLE` stellen). Das Feld `TOOL TYPES` sollten Sie zunächst ignorieren — es kann auch bei der augenblicklichen Version des Systems kaum sinnvoll eingesetzt werden.

Wie bei allen anderen Info-Fenstern gilt auch hier, daß die Änderungen nur dann gültig werden, wenn Sie das Fenster mit einem Klick auf den Knopf `SAVE` schließen. Schließen Sie es mit `QUIT` oder mit einem Klick ins Schließ-Gadget, werden alle Änderungen zwischen dem Öffnen und Schließen des Fensters verworfen.

4.2 Projekte

Die Objekte (Dateien), die von einem Programm erzeugt werden, heißen auf der Workbench *Projekte* (engl. *projects*). In der, inzwischen ja schon sattem bekannten, Werkbank-Analogie kann man sich das so vorstellen, daß mit einem bestimmten Werkzeug an einem bestimmten (oder mehreren) Projekt(en) gearbeitet wird. Diese Verbindung zwischen Werkzeugen wird meist auch durch die Icons der entsprechenden Workbench-Objekte verdeutlicht, wie wir unten auch an einem Beispiel sehen werden.

Um etwas über Projekte sagen zu können, benötigen wir natürlich erst einmal ein solches Objekt. Dazu muß erwähnt werden, daß nicht alle Werkzeuge auch für die Bearbeitung von Projekten eingesetzt werden. Die Uhr z.B., die in den vorangegangenen Abschnitten vorgestellt wurde, kann keine Projekte bearbeiten. (Welches Projekt wollte man auch "in Wirklichkeit" mit dem "Werkzeug Uhr" bearbeiten?)

Auf der Workbench-Diskette befindet sich glücklicherweise aber auch ein simples Textverarbeitungsprogramm, mit dem Projekte (Texte) erzeugt und verarbeitet werden können. Es heißt *Notepad* (engl. für *Notizblock*) und befindet sich in der Schublade *Utilities* (zu deutsch etwa *kleine Nützlichkeiten*) auf der Workbench-Diskette.

In der selben Schublade befindet sich auch noch ein Werkzeug namens *Calculator* (*Taschenrechner*). Dessen Bedienung ist so einfach, daß wir uns jegliche Erläuterungen dazu ersparen. Es handelt sich wirklich um einen am Bildschirm simulierten Taschenrechner, dessen Tasten allerdings auch mit der Maus bedient werden können.

4.2.1 Das Werkzeug Notepad

Nachdem Sie die Schublade *Utilities* geöffnet haben, können Sie das Werkzeug *Notepad* genauso einfach öffnen wie die Uhr. Ein Doppelklick mit der Maus auf das Icon genügt. Das Diskettenlaufwerk rotiert nun aber schon merklich länger, bis endlich ein neues Fenster mit dem Namen *Notepad V1.7* erscheint. Im Gegensatz zur Uhr wird dieses Fenster sofort aktiv und auch die Titel-Leiste des Bildschirms zeigt gleich durch den Inhalt *Notepad* an, daß dieses Programm nun Ihre Eingaben erhält.

Notepad ist eine sehr einfache Textverarbeitung, die für das Anlegen kleiner Notizen und zum Herumspielen mit den Möglichkeiten der Textverarbeitung auf dem Amiga aber völlig ausreicht. Ihnen stehen insgesamt zehn "Blätter" zur Verfügung, auf denen sie kurze Texte eintippen, wieder löschen, ausdrucken und abspeichern können.

Nach dem Start des Werkzeugs wird Ihnen zunächst ein leeres weißes Blatt gezeigt, dessen Größe Sie mit Hilfe des Größen-Gadgets Ihren Bedürfnissen anpassen können. Oben links auf diesem weißen Blatt befindet sich ein dünner schwarzer Strich. Tippen Sie nun auf der Tastatur ein paar Buchstaben, so erscheinen diese immer hinter der Stelle, an der der Strich gerade steht. Bei jedem Buchstaben, der neu eingetippt wird, wandert der Strich nach rechts weiter. Es handelt sich bei ihm natürlich um eine sogenannte Schreibmarke (auch Cursor), die Sie ja schon kennengelernt haben (z.B. beim Befehl *Rename*). Diese Schreibmarke steht — im Gegensatz zu der beim *Rename*-Befehl oder in einem *Info*-Fenster — immer **zwischen** zwei Buchstaben und nicht über einem Buchstaben. Neu eingetippter Text erscheint immer zwischen diesen beiden Buchstaben und schiebt den Text rechts davon zur Seite.

Wenn Sie sehen wollen, was der Amiga so alles an verschiedenen Buchstaben zu bieten hat, so halten Sie einmal eine der beiden <ALT>-Tasten fest und tippen dann auf eine der "normalen" Buchstaben-Tasten — genauso wie sie die <SHIFT>-Taste festhalten müssen, um Großbuchstaben zu bekommen. Sie können mit der <ALT>-Taste eine Vielzahl von speziellen Buchstaben und Sonderzeichen, wie ñ, œ, ç, µ und ð erzeugen, so daß Sie auch bei Briefen an Ihre französischen oder spanischen Freunde nie in die Verlegenheit kommen werden, einen Buchstaben der Landessprache nicht verwenden zu können.

Machen Sie Fehler bei der Texteingabe, so können Sie diese mit der <BACKSPACE>-Taste (oben rechts auf der Tastatur) wieder löschen. Die <BACKSPACE>-Taste löscht immer den ersten Buchstaben links von der

Schreibmarke. Die restlichen Buchstaben rechts davon rücken dabei nach links auf.

Die Schreibmarke wandert nicht nur "von selbst" weiter, während Sie Text eintippen oder löschen. Klicken Sie mit der Maus zwischen zwei Buchstaben, so springt die Schreibmarke zu dieser Stelle. Neu eingetippter Text erscheint dann dort. Während Sie gerade Ihre Hände auf der Tastatur haben, können Sie die Schreibmarke aber vielleicht noch bequemer mit den Pfeil-(Cursor-)Tasten bewegen. Jede dieser vier Tasten bewegt die Schreibmarke in die angegebene Richtung "solange es geht".

Wird das Fenster voll, so können Sie auf ein neues Blatt überwechseln. Hierzu dienen die beiden Gadgets oben rechts und unten links im Fenster, die wie Eselsohren aussehen. Das Gadget oben rechts blättert vor, bis die letzte (zehnte) Seite erreicht ist und das Gadget unten links blättert wieder zurück.³

Den Text, den Sie auf diese Weise erstellt haben, verlieren Sie aber normalerweise in dem Augenblick, wo Sie das Notepad-Fenster wieder schließen. Sie können ihn jedoch auch (relativ) dauerhaft auf einer Diskette abspeichern, womit wir zu dem Aspekt von Notepad kommen, der uns im Zusammenhang Werkzeuge und Projekte besonders interessiert.

4.2.2 Ein Project-Menü

Das erste Menü links in der Menü-Leiste von Notepad trägt den Titel **Project**. Es dient zur Auswahl von Operationen, die den Text als Ganzes, also als ein Projekt, betreffen. Dieses Menü ist in gleicher oder ähnlicher Form bei vielen anderen Programmen auch zu finden. Es lohnt sich deshalb, sich ein wenig näher damit zu beschäftigen.

Der erste Befehl, **New** (engl. für *Neu*), löscht das Notepad-Fenster und beginnt so ein neues Projekt. Aller Text, den Sie eingegeben haben, geht dadurch verloren. Versuchen Sie also, diesen Befehl nie aus Versehen auszuwählen.

Der Befehl **Save** (engl. für *Sichern*) dient zum Abspeichern des gerade erstellten Textes auf der Diskette. Wenn Sie den Text das erste Mal mit

³ Sie können auch mehr Text auf ein Blatt tippen, als ins Fenster paßt. Der Inhalt wird dann nach oben verschoben und geht dabei auch nicht verloren. Das "Navigieren" innerhalb eines solchen "übervollen" Blattes ist allerdings recht umständlich, weshalb man dies vermeiden sollte. Notepad ist eben **kein** vollständiges Textverarbeitungsprogramm!

diesem Befehl abspeichern, müssen Sie Ihm zunächst einen Namen geben, wofür extra ein neues Fenster geöffnet wird. Hatten Sie den Text schon einmal gesichert, so wird es einfach noch einmal unter dem alten Namen abgespeichert **und die letzte Version dabei zerstört!**

Das Fenster, in dem Sie dem Text einen Namen geben müssen, enthält nur einen kleinen Rahmen, in dem Sie den Namen eingeben können und zwei Tasten. Zur Eingabe des Namens stehen Ihnen dieselben Möglichkeiten zur Verfügung wie bei der Eingabe eines neuen Icon-Namens (bei **Rename**) oder eines Kommentars. (Denken Sie daran, den Text-Rahmen erst anzuklicken, bevor Sie den Namen eintippen!) Wenn Sie damit fertig sind, drücken Sie die <RETURN>-Taste und dann den Knopf **OK**. Ihr Text wird nun unter dem eingegebenen Namen auf die Diskette geschrieben und wird somit *gesichert*. (Im Gegensatz zu den Daten im RAM-Speicher des Amiga überstehen die Daten auf der Diskette z.B. das Ein- und Ausschalten des Geräts und sind somit verhältnismäßig *sicher* zu nennen.)

Der Befehl **Save As** (engl. für *Sichern Unter*) wirkt genau wie **Save**, nur daß Sie **immer** einen Namen für den abzulegenden Text eingeben müssen. Auch dann, wenn Sie ihn schon einmal gesichert hatten und ihm dabei einen Namen gegeben haben. **Save As** dient vor allem zum Ablegen schon anderweitig gespeicherter Texte unter einem neuen Namen.

Save und **Save As** sind (auch in den meisten anderen Werkzeugen) die Befehle, die neue Projekte erzeugen. Sie können das leicht überprüfen, indem Sie einen kurzen Text im Notepad eingeben und diesen dann unter einem beliebigen Namen sichern. Schließen Sie dann das Fenster **Utilities** (in dem ja das Werkzeug Notepad liegt) und öffnen es dann wieder. Darin erscheint auf einmal ein neues Icon, das wie ein Blatt Papier aussieht und unter dem der Name steht, den Sie soeben ihrem Text gegeben haben. Dies ist ein *Projekt*, wie es Notepad erzeugen und bearbeiten kann.

Mit dem Befehl **Open** können Sie ein solches Projekt wieder im Notepad-Fenster anzeigen lassen und bearbeiten, wenn Sie es einmal abgespeichert haben. Nachdem Sie **Open** ausgewählt haben, erscheint wieder das kleine Fenster, das Sie ja schon von **Save As** kennen, und Sie müssen einen Namen eingeben. Drücken Sie danach <RETURN> und dann mit der Maus auf **OK** und der Text mit diesem Namen wird eingelesen und angezeigt. **Open** zerstört dabei aber den alten Text, den Sie zuvor im Notepad bearbeitet hatten. Wenn Sie ihn nicht zuvor gesichert hatten, ist er danach verloren.

Print (engl. für *Drucken*) ist ein völlig "neuartiger" Menü-Punkt. Sobald Sie nämlich mit der Maus über diesen Punkt kommen, erscheint ein neues kleines Menü rechts daneben. Dies ist ein sogenanntes *Sub-Menu* und illustriert eine weitere Möglichkeit, die Amiga-Menüs bieten. Jeder Punkt in einem (Haupt-)

Menü kann nämlich ein weiteres (Sub-) Menü hervorbringen. Punkte, die dies tun, sind eigentlich keine "richtigen" Programm-Befehle, sondern nur Obertitel einer Gruppe von Befehlen. Wenn ein solches Sub-Menü erscheint, so genügt es nicht, diesen Oberpunkt auszuwählen und die Menü-Taste loszulassen. Um dem Programm einen Befehl zu erteilen, muß ein Punkt aus dem Sub-Menü ausgewählt werden.

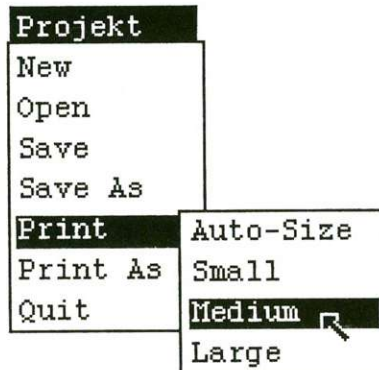


Bild 4 - 3: Eine Sub-Menü taucht auf

Im Falle von **Print** sind dies die Unterpunkte **Auto-Size** (engl. für *automatische Größe*), **Small** (engl. für *Klein*), **Medium** (engl. für *Mittel*) und **Large** (engl. für *Groß*). Alle diese Befehle bewirken einen Ausdruck des Fensterinhalts auf dem Drucker (sofern Sie einen besitzen und korrekt angeschlossen haben; s.u.). Wählen Sie **Auto-Size** überlassen Sie Notepad, in welcher Größe Ihr Text ausgegeben wird. Bei **Small**, **Medium** und **Large** bestimmen Sie selbst die Größe des Ausdrucks (klein, mittel oder groß). Experimentieren Sie am besten ein wenig mit den verschiedenen Möglichkeiten! Es kostet ein wenig Papier, ist aber die beste Methode, um zu wirklich (für Sie) brauchbaren Ergebnissen zu kommen.

Der Befehl **Print As** läßt ebenfalls ein Sub-Menü erscheinen. Sie haben hier die Auswahl zwischen **Graphic** (engl. für *Grafik*) und **Draft** (engl. für *Entwurf*). Diese beiden Punkte des Sub-Menüs sind Schalter, die sich gegenseitig ausschließen. (Solche Menü-Schalter sind ja bereits von der Uhr her bekannt.) Ist der Schalter **Graphic** an, so werden bei einem Ausdruck des Textes alle Besonderheiten und speziellen Effekte, die im nächsten Abschnitt noch vorgestellt werden, auf dem Papier ausgegeben. Sie erhalten dann einen Ausdruck, der weitgehend dem Bild am Bildschirm entspricht. Dies gilt allerdings nur dann, wenn Sie einen grafikfähigen Drucker besitzen. Schließen Sie einen nicht grafikfähigen Drucker an den Amiga an, oder der Schalter **Draft** im Sub-Menü **Print As** steht auf An, so werden zwar alle

Buchstaben des Textes korrekt ausgegeben, die verschiedenen Zeichensatz- und Zeichenstil-Formen (s.u.) werden ignoriert.

Der letzte Befehl im **Project**-Menü ist ein besonders schwerwiegender. Mit **Quit** verlassen Sie das Werkzeug Notepad, und sämtlicher Text, den Sie nicht gesichert haben, ist verloren. Denselben Effekt wie mit Quit können Sie aber auch erreichen, wenn Sie das Schließ-Gadget des Notepad-Fensters betätigen.

4.2.3 Möglichkeiten von Notepad

Wo Sie nun schon ein Menü kennengelernt haben, sollen Ihnen die anderen Möglichkeiten von Notepad auch nicht vorenthalten werden. Notepad ist nicht so sehr ein wirklich brauchbares Anwendungsprogramm als vielmehr eine Demonstration der Textausgabe auf dem Amiga. Sie können z.B. im Menü **Font** (engl. für *Zeichensatz*) den Zeichensatz ändern, mit dem der von Ihnen eingegebene Text im Fenster erscheint. Dies wirkt, als würden Sie bei einer elektrischen Schreibmaschine den Kugelpopf oder das Typenrad tauschen. Probieren Sie die verschiedenen Möglichkeiten einfach einmal aus.

Die Menü-Punkte im **Font**-Menü tragen alle Namen von Edelsteinen oder Halbedelsteinen — dies ist, bei all ihren vielen Unterschieden, die gemeinsame Eigenschaft der Zeichensatz auf dem Amiga. Alle Menü-Punkte enthalten aber noch einmal ein Unter-Menü, in dem Sie noch die Größe wählen müssen, in der die Buchstaben im Fenster erscheinen sollen. Einige Zeichensätze (z.B. *sapphire*) können in bis zu vier Größen ausgegeben werden. Der Name des gerade ausgewählten Zeichensatzes wird im Menü abgehakt, damit Sie auch immer wissen, wie der Zeichensatz heißt, den Sie gerade sehen.

Im Menü **Style** (engl. für *Zeichenstil*) können Sie wählen, ob der Text normal (**Plain**), kursiv (**Italic**), fett (**Bold**) oder unterstrichen (**Underline**) ausgegeben werden soll. Im Gegensatz zur Einstellung für den Zeichensatz, die immer für den ganzen Text gilt, können Sie im Fenster durchaus Text-Passagen haben, die einen unterschiedlichen Zeichenstil haben. Es können auch mehrere Stilmittel miteinander kombiniert werden. So sind Passagen möglich, die fett gedruckt sind, andere die kursiv und wieder andere, die unterstrichen und kursiv gedruckt sind.

Das letzte Menü **Format** schließlich bestimmt, in welcher Farbe das "Papier" angezeigt werden soll (**Paper Color**; zu deutsch *Papier-Farbe*) und in welcher Farbe die Buchstaben darauf erscheinen sollen (**Pen Color**; zu deutsch *Stift-Farbe*). Beides können Sie frei unter den vier Farben

auswählen, die Ihnen in der Workbench zu Verfügung stehen. Die Grundeinstellung ist — wie könnte es auch anders sein — schwarzer Text auf weißem Hintergrund.

4.3 Zusammenhänge

Projekte und Werkzeuge gehören stets eng zusammen. Üblicherweise gibt es zu jedem Projekt nur ein Werkzeug, mit dem es bearbeitet werden kann und alle Projekte, die mit einem bestimmten Werkzeug bearbeitet werden können, besitzen auch das selbe Icon.

4.3.1 Öffnen eines Projektes

Der Zusammenhang zwischen Projekten und Werkzeugen wird aber nicht nur durch solche Äußerlichkeiten bestimmt, sondern Sie haben beispielsweise auch die Möglichkeit, ein Projekt zu öffnen wie ein Werkzeug. D.h., wenn Sie durch einen Doppelklick oder den Befehl **Open** ein Projekt statt eines Werkzeuges öffnen, so wird Ihnen — wie von anderen Objekten bekannt — gezeigt, was es enthält. Im Falle eines Textes wird also dieser Text in einem Fenster gezeigt. Und die einfachste Methode, diesen Text zu zeigen, ist natürlich, das Programm, mit dem er erstellt wurde, zu öffnen!

Wenn Sie ein Projekt öffnen, prüft die Software des Amiga, zu welchem Werkzeug es gehört und startet dieses. Gleichzeitig wird dem Werkzeug mitgeteilt, welches Projekt der Anwender geöffnet hat. Ein gut programmiertes Werkzeug liest dieses Projekt daraufhin ein und zeigt es an.

Dies ist wesentlich praktischer, als immer zuerst ein Werkzeug starten zu müssen und dann das Projekt einzulesen, wenn man es bearbeiten will. Hier geschieht also beim Öffnen eines Projektes genau das, was Sie aus Ihrer bisherigen Erfahrung mit anderen Objekten erwarten.

4.3.2 Das Info-Fenster eines Projektes

Die Verbindung zwischen einem Projekt und dem zugehörigen Werkzeug wird auch dann sichtbar, wenn man das Info-Fenster des Projektes öffnet (also dem Projekt den Befehl *Info* erteilt).

The screenshot shows a window titled 'Info release 1.1'. It contains several fields and buttons:

- NAME:** Notiz
- TYPE:** Project
- SIZE:**
 - in bytes: 22
 - in blocks: 1
- STACK:** 8000
- STATUS:** DELETEDABLE
- COMMENT:** Ein Kommentar
- DEFAULT TOOL:** SYS:Utilities/Notepad
- TOOL TYPES:** (empty field) with **ADD** and **DEL** buttons.
- SAVE** and **QUIT** buttons at the bottom.

Bild 4 – 4: *Das Info-Fenster eines Projektes*

Das Info-Fenster eines Projektes sieht genauso aus, wie das eines Werkzeuges mit einem zusätzlichen Feld namens **DEFAULT TOOL**. Dahinter kann der Name eines Werkzeuges eingegeben werden, das aufgerufen werden soll, wenn das Projekt geöffnet wird. Beim ersten Ablegen eines neuen Projektes auf der Diskette hinterlegt das erzeugende Programm hier meist seinen eigenen Namen.

Betrachtet man das Info-Fenster eines Notepad-Projektes, so steht dort z.B. **SYS:Utilities/Notepad**. **SYS:** bedeutet, das Werkzeug ist auf der Start-Diskette zu finden, **Utilities**, es liegt in der Schublade **Utilities** und **Notepad** schließlich ist natürlich der Name des Werkzeuges selbst.⁴ Es ist theoretisch möglich, hier einen anderen Namen einzutragen. Wenn Sie z.B. einmal eine leistungsfähigere Textverarbeitung erwerben, können Sie unter

⁴ Mehr zu dieser ungewöhnlichen Schreibweise können Sie erfahren, wenn Sie das Kapitel *Dateien und Dateiverzeichnisse* lesen.

Umständen auf diese Weise erreichen, daß beim Öffnen von Texten, die mit **Notepad** erstellt wurden, dieses neue Werkzeug gestartet wird und nicht mehr **Notepad**.

4.4 Multitasking

Einen Aspekt von Werkzeugen haben wir bisher als völlig selbstverständlich dargestellt, obwohl er es eigentlich gar nicht ist. Nachdem ein Werkzeug geöffnet wurde, hält einen niemand davon ab, in ein anderes Fenster zu klicken und dort mit einem anderen Programm zu arbeiten. Das beste Beispiel für dieses Verhalten ist die Workbench selbst, die ja auch ein Programm ist. Nachdem von ihr aus ein Werkzeug geöffnet wurde, kann man problemlos weitere Schubladen öffnen, verschieben oder auch weitere Werkzeuge öffnen. Während der ganzen Zeit läuft ein erst einmal gestartetes Programm (Werkzeug) weiter, wie z.B. bei der Uhr sehr gut zu sehen ist, die ja auch ein Werkzeug ist.

Auf anderen Computern, von denen Sie möglicherweise auch den einen oder anderen kennen, ist ein solches Verhalten durchaus nicht selbstverständlich. Startet man dort ein Programm, so "übernimmt es" den ganzen Computer, beansprucht den ganzen Speicher für sich, löscht unter Umständen alles was sich darin befindet und beansprucht auch den gesamten Bildschirm für sich. Auf dem Amiga üben die Programme hingegen (weitgehend) friedliche Koexistenz, wozu unter anderem auch die Fenster beitragen, die dafür sorgen, daß den Programmen genau abgegrenzte Bereiche des Bildschirms zugeteilt werden können, ohne sie dadurch allzusehr einzuschränken.

Der Fachmann nennt diese Verhalten (gleichzeitige Aktivität mehrerer Programme) eines Computers *Multitasking*, weil der Computer gleichzeitig an mehreren (*Multi*-) Aufgaben (eng. *tasks*) arbeiten kann. Dies ist natürlich nur eine scheinbare Gleichzeitigkeit, da auch ein Computer immer nur ein Programm zur gleichen Zeit bewältigen kann. Der Amiga arbeitet aber immer nur ein kleines Stück eines Programms ununterbrochen ab, danach "springt" er zu einem anderen und bearbeitet dieses für einen kleinen Augenblick usw. Er wechselt aber so schnell zwischen den Programmen hin und her, daß er den Eindruck erweckt, er würde diese Programme gleichzeitig bearbeiten.

4.4.1 Vorteile des Multitaskings

Eventuell werden Sie sich nun fragen, wozu Multitasking denn überhaupt gut sein soll. Denn obwohl der Computer scheinbar schnell genug ist, mehrere Aufgaben gleichzeitig zu bewältigen, kann ja kaum vom einem Menschen verlangt werden, blitzschnell zwischen der Bedienung von zwei oder drei Programmen hin und her zu wechseln. Ganz davon abgesehen, daß für einen solchen ständigen Wechsel wohl kaum Bedarf besteht.

Dies ist auch völlig richtig, und auch das Multitasking des Amiga ist nicht dafür gedacht, daß sie simultan einen Text tippen und Ihre Buchhaltung erledigen. Es gibt aber bei vielen Programmen Vorgänge, die sehr viel Zeit in Anspruch nehmen — einige Minuten oder gar eine Stunde und mehr. Das Ausdrucken langer Dokumente und aufwendige Kalkulationsaufgaben sind z.B. solche Vorgänge. Wenn Ihr Computer immer nur ein Programm bearbeiten kann, müssen Sie darauf warten, daß das Programm fertig wird, bevor Sie mit der Arbeit fortfahren können — oder Sie müssen sich einen zweiten Computer kaufen. Beherrscht Ihr Computer aber das Multitasking, können Sie, während sich das eine Programm seiner langwierigen Tätigkeit hingibt, ein anderes Programm für eine andere Aufgabe nutzen.

Aber auch ein schnelles Wechseln zwischen verschiedenen Programmen kann hilfreich sein. Nehmen wir an, sie wollen einen Bericht schreiben und verwenden dazu ein Textverarbeitungsprogramm. Plötzlich merken Sie, daß Ihnen einige Daten fehlen. Diese Daten verwalten Sie normalerweise mit einem Datenbankprogramm, und um sie jetzt zu betrachten, müßten Sie dieses Programm nun starten. Auf der Workbench des Amiga können Sie die Daten nachsehen, ohne Ihre Arbeit in der Textverarbeitung erst abzuspeichern, und sind danach blitzschnell wieder zurück in der Textverarbeitung. Gerade das Starten und Verlassen von Programmen sind Vorgänge, die auf einem (bei aller Leistungsfähigkeit doch vergleichsweise) kleinen Computer wie dem Amiga recht zeitraubend sein können.

Die Möglichkeit des Multitaskings kann Ihnen also in vielerlei Hinsicht Zeit und Mühe ersparen und nutzt zugleich den Computer auch noch viel besser aus. Selbst ein kleiner Computer wie der Amiga ist nämlich immer noch so schnell, daß er einen Großteil seiner Zeit nichts anderes tut, als auf ihre Eingaben zu warten. In dieser Zeit könnte er genausogut schon sinnvollere Dinge tun — z.B. drucken oder rechnen.

4.4.2 Mögliche Probleme des Multitaskings

Multitasking ist aber auch nicht ganz ohne Probleme. Wenn zwei identische Programme gleichzeitig laufen, benötigen sie auch den doppelten Speicherplatz im Computer. Gerade bei großen und leistungsfähigen Programmen werden Sie rasch feststellen, daß selbst die ca. 260 000 oder 520 000 Zeichen, die der Amiga in seinem RAM-Speicher ablegen kann, nicht allzu viel sind. Während Sie auf der Workbench sind, können Sie sehr schön beobachten, wie das Öffnen eines jeden Objektes (aber vor allem eines Werkzeugs oder Projektes) den freien Speicherplatz, der oben in der Titel-Leiste des Bildschirms angezeigt wird, reduziert. Genau für diesen Zweck ist die Speicheranzeige nämlich auch gedacht. Sie gibt Ihnen die Möglichkeit, recht genau den Speicherbedarf eines Programms zu ermitteln.

Wollen Sie häufig mit mehreren Werkzeugen gleichzeitig arbeiten, also das Multitasking so richtig nutzen, so ist es sinnvoll, sich zu jedem Programm zu notieren, welchen Speicherbedarf es ungefähr hat. Zur Ermittlung dieses Wertes reicht meist ein Probelauf des Programms und die Kontrolle der Anzeige des freien Speichers in der Workbench-Titel-Leiste. Sie können dann jederzeit abschätzen, ob für ein weiteres Werkzeug, das sie gerade öffnen wollen, noch genügend Platz ist.

Ein weiteres Problem des Multitaskings ist natürlich, daß sich jedes Programm, welches gleichzeitig mit anderen Programmen läuft, die Leistung des Computers mit diesen anderen Programmen teilen muß. Bei zwei oder drei Programmen, die gleichzeitig laufen, kann dies noch ganz erträglich sein, insbesondere, wenn diese Programme nicht sehr "rechenintensiv" sind, d.h. den Computer nicht allzusehr beanspruchen. Je mehr Programme Sie dann aber darüber hinaus noch starten, desto langsamer werden die Reaktionen des Amiga auf Ihre Eingaben.

Wollen Sie einen ungefähren Eindruck dieses Effekts haben, so öffnen Sie einmal die Schublade **Demos** auf der Workbench-Diskette. Darin befinden sich drei kleine Programme namens **Dots** (engl. für *Punkte*), **Boxes** (engl. für *Kästen*) und **Lines** (engl. für *Linien*). Diese Programme leisten nichts großartiges, sondern malen nur ein wenig in ihren jeweiligen Fenstern herum. Sie eignen sich aber sehr gut als Demonstrationsobjekte für das Multitasking.

Öffnen Sie die drei Programme nacheinander. Verschieben und vergrößern Sie die dabei neu auftauchenden Fenster danach und betrachten Sie die grafischen Effekte ein wenig, die in den Fenstern zu sehen sind. Achten Sie dabei vor allem auf die Geschwindigkeit, mit der die Programme ablaufen. (Das **Lines**-Fenster sollten Sie recht groß machen, damit die Grafik gut sichtbar wird.)

Öffnen Sie die drei Programme jetzt nacheinander noch einmal und achten Sie dabei darauf, wie die Geschwindigkeit, mit der die Grafiken gezeichnet werden, bei jedem neuen Programm weiter abnimmt. Auch die Reaktion der Workbench auf das Verschieben, Vergrößern und Verkleinern eines Fensters wird immer "träger". Schließen Sie nun wieder einige der Fenster und achten Sie darauf, wie die Geschwindigkeit in den übrigbleibenden Fenstern dabei wieder größer wird. Lassen Sie das **Lines**-Fenster als letztes offen, dann erhalten Sie den spektakulärsten Effekt.

Sie haben nun "life" miterlebt, wie sich ein Computer aufteilt, und dabei gehörig "in die Knie geht". Denken Sie aber daran, daß nicht alle Programme immer so "beschäftigt" sind, wie die drei **Demos**-Programme. Viele Werkzeuge auf der Workbench eignen sich sehr gut für ein zu anderen Programmen parallel laufendes Arbeiten und sie werden dabei nicht unerträglich langsamer.

4.5 Screens

Mit der Verwendung eines Fensters hat ein Programm ja schon die Möglichkeit, recht unabhängig von dem zu werden, was die anderen Programme gerade tun. Das Programm kann z.B. "so tun", als hätte es den ganzen Bildschirm oder eine noch viel größere Fläche allein zu Verfügung und kann auf dieser ganzen Fläche Grafiken, Texte, usw. zeigen. Sichtbar wird davon aber nur ein Teil, der so groß wie das Fensterinnere des Fensters ist, das dem Programm beim Start zugeteilt wurde. Der Programm-Anwender, nicht der Programmierer, entscheidet dann darüber, wie groß dieser Ausschnitt ist, und welcher Teil des Gesamtbildes gezeigt wird.

Die Werkzeuge des Amiga, die wir bisher kennengelernt haben, nutzten seine grafischen Fähigkeiten kaum. Sie öffnen alle nur ein Fenster auf der Workbench und begnügen sich mit den vier Farben, die die Workbench standardmäßig zur Verfügung stellt (normalerweise schwarz, weiß, blau und orange).

Was aber, wenn einem Programm die *Bildschirm-Attribute*, wie sie die Workbench standardmäßig zur Verfügung stellt "nicht gefallen"? Ein Programm kann z.B. eine größere Anzahl oder eine andere Kombination von Farben bevorzugen (oder benötigen) als die Workbench. Der Amiga bietet ja noch wesentlich mehr Möglichkeiten als die vier Farben des Workbench-Bildschirms.

Ein solches Programm könnte nun natürlich einfach in den Interna des Amiga "herumpfuschen" und die gewünschten Einstellungen für die Bildschirm-Attribute vornehmen. Alle anderen Programme, die gerade aktiv sind, würden davon aber beeinträchtigt. Es könnte sein, daß deren Fenster auf einmal schwer oder gar nicht mehr lesbar wären. Es könnte aber auch vorkommen, daß diese Programme gar nicht mehr lauffähig sind und nur noch völligen Quatsch produzieren.

Glücklicherweise erlaubt es die Video-Hardware des Amiga aber, jedem Programm seinen eigenen *virtuellen Bildschirm* zu Verfügung zu stellen. (Statt *virtuellen Bildschirm* werden wir oft auch einfach den Fachausdruck *Screen* verwenden; er ist einfach kürzer.) Diese Möglichkeit heißt deshalb *virtueller Bildschirm*, weil es natürlich nur einen wirklichen, *physikalischen* Bildschirm gibt, den Sie ja vor sich sehen.

Einem Programm kann es egal sein, ob es einen virtuellen Bildschirm verwendet oder einen echten; die Programmierung bleibt nahezu gleich. Nur Sie als Anwender müssen sich entscheiden, welchen virtuellen Bildschirm Sie auf Ihrem wirklichen Bildschirm sehen wollen. Wollen Sie (Teile von) mehrere(n) virtuelle Bildschirme(n) sehen, müssen Sie den wirklichen Bildschirm entsprechend "aufteilen". Betrachten wir nun einmal, wie man so einen virtuellen Bildschirm verwendet:

Vielleicht ist es Ihnen in der Workbench schon öfter einmal passiert, daß Sie aus Versehen mit der linken Maustaste in die Titel-Leiste des Bildschirms geklickt haben, statt die rechte (Menü-) Taste zu nehmen. Wenn man bei einem solchen Fehler die Maus bewegt, so kann es passieren, daß sich das Bild zu bewegen beginnt.

Klickt man nämlich mit der linken Taste in die Titel-Leiste eines (virtuellen) Bildschirms, so kann man diesen ergreifen — wie ein Icon oder einen Schieberegler — und auf und ab bewegen.⁵ Liegt "hinter" dem Workbench-Bildschirm noch ein anderer Bildschirm, so kommt dieser zum Vorschein, wenn Sie den Workbench-Bildschirm nach unten ziehen.

Falls Sie nur die Grundausrüstung des Amiga besitzen, können Sie dies leider nicht ausprobieren. Keines der Programme, die in der Grundausrüstung mitgeliefert werden, benutzt einen eigenen virtuellen Bildschirm. Einige der Demo-Programme, die Ihr Händler für Kunden-Demonstrationen von Commodore/Amiga zur Verfügung gestellt bekommen hat, nutzen diese Fähigkeit aber. Genauso die Malprogramme Gaphicraft und Deluxe Paint und

⁵ Die derzeitige Hardware des Amigs ist noch nicht fähig, virtuelle Bildschirme auch seitlich zu bewegen.

viele andere Programme für den Amiga. Nur wenn Sie eines dieser Programme schon besitzen, können Sie Screens austesten.

Der Umgang mit Screens ist ganz einfach. Sie verhalten sich fast genauso wie Fenster. Sie können sie an ihrer Titel-Leiste ergreifen und auf- und abbewegen — jedoch nicht nach links oder rechts. Die Reihenfolge, in der Screens "übereinanderliegen", können Sie mit den Back- und Front-Gadgets am rechten Rand der Titel-Leiste eines Screens ändern.

Ansonsten funktionieren Screens genauso wie Fenster (ihre Größe ist aber ebenfalls konstant — eben ein voller Bildschirm) und dienen auch den selben Zwecken. Vor allem ermöglichen sie die gleichzeitige Bearbeitung von Programmen, die sehr unterschiedliche Bedürfnisse bezüglich der Bildschirm Eigenschaften haben.

4.6 Ein wichtiges Werkzeug: Preferences

Vor Abschluß dieses Kapitels sollen Sie noch eines der wichtigsten Werkzeuge der Workbench kennenlernen: **Preferences** (zu deutsch etwa: *Bevorzugtes*). Mit diesem Werkzeug können Sie z.B. Uhrzeit und Datum stellen, die Farben ändern, in denen Ihnen die Workbench am Bildschirm gezeigt wird und verschiedene Drucker an den Amiga anpassen.

4.6.1 Das Hauptfenster von Preferences

Nachdem Sie **Preferences** geöffnet haben, erscheint ein bildschirmfüllendes Fenster, das geradezu überläuft von einer Ansammlung der verschiedensten Gadgets. Dies ist das Haupt-Fenster von Preferences. Wenn Sie auf einige der verschiedenen Tasten in diesem Fenster drücken, so erscheinen noch drei andere Fenster, die aber nie gleichzeitig mit dem Hauptfenster zu sehen sind, sondern es immer (vorübergehend) verdecken. Sie finden Abbildungen einiger dieser Fenster auf den Farbseiten dieses Buches. Ausschnitte davon sehen Sie aber auch in den folgenden Abbildungen.

Nachdem Sie Preferences aufgerufen haben, können Sie es nur aus diesem Fenster heraus wieder verlassen. Unten rechts im Fenster sind fünf "Tasten" zu sehen: **Change Printer**, **Edit Pointer**, **Save**, **Use** und **Cancel**. Wenn Sie drei davon (mit einem Mausklick) betätigen, so wird Preferences

verlassen, die beiden anderen Knöpfe zeigen Ihnen zwei der anderen Fenster von Preferences.

Drücken Sie auf **Cancel** (engl. für *Abbrechen* oder *Vergiß es*), so wird das Werkzeug **Preferences** verlassen und alle Änderungen, die Sie durchgeführt haben, werden ignoriert. Drücken Sie auf **Save**, so werden die Änderungen gültig und zugleich auf Ihrer Workbench-Diskette gespeichert, so daß Sie beim nächsten Start genauso verwendet werden können. Betätigen Sie **Use**, werden Ihre Änderungen und Einstellungen zwar gültig, werden aber nicht gespeichert, d.h. nach dem nächsten Start gelten wieder die alten.

Zu den Einstellungen, die Sie in **Preferences** ändern können, gehören z.B. die vier Farben, die für das Zeichnen des Workbench-Bildschirms verwendet werden. Dazu dienen die vier "Schieberegler" unten links im Fenster.

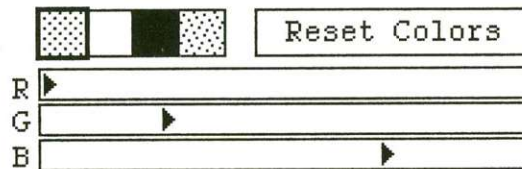


Bild 4 – 5: Einstellen der Workbench-Farben

Wollen Sie die Farben ändern, müssen Sie zunächst eine aussuchen. Dies geschieht, indem Sie eines der vier farbigen Quadrate, die über den Reglern zu sehen sind, anklicken. Die so selektierte Farbe wird dann mit einem Rahmen umgeben. Die kleinen Dreiecke in den Reglern kann man nun mit der Maus ergreifen und nach links und rechts verschieben, wobei sich die ausgewählte Farbe sofort ändert. Die Buchstaben **R**, **G** und **B** stehen für Rot, Grün und Blau. Aus diesen drei Grundfarben werden die am Bildschirm sichtbaren Farben nämlich "zusammengemischt". Sind alle Regler am rechten Anschlag, ergibt dies weiß. Sind alle am linken Anschlag, entsteht schwarz. (Auf die selbe Weise — durch additive Farbmischung aus Rot, Grün und Blau — entstehen übrigens auch die Farben auf Ihrem Farbfernseher.)

Es ist nicht ganz einfach, eine Farbe, die man sich vorstellt, auf diese Weise zusammenzumischen. Ein wenig Experimentieren wird Ihnen aber bestimmt bald ein Gefühl dafür vermitteln. Sie sollten ruhig ein wenig mit den drei Reglern herumspielen! Ähnliche Regler werden in fast jedem Mal- und Grafik-Programm für den Amiga verwendet.

Falls es Ihnen gelungen sein sollte, so scheußliche Farben zusammenzumischen, daß Ihre Augen es nicht mehr aushalten, können Sie übrigens mit der Taste **Reset Colors** die Standard-Kombination (Schwarz, Weiß, Orange und Blau) wiederherstellen.

Während die Verstellung der Workbench-Farben eher in die Kategorie *Spielerei* fällt, ist die folgende Funktion von Preferences durchaus ernsthaft. Oben rechts im Fenster sehen Sie die aktuelle Uhrzeit und das Datum — oder jedenfalls, was der Amiga dafür hält. Wenn Sie Wert darauf legen, daß seine Vorstellung von Zeit und Datum mit der Ihren übereinstimmt, so können Sie die Anzeige mit den beiden Pfeilen daneben korrigieren.

01/26/86 
10:31

Bild 4 – 6: *Einstellen der richtigen Uhrzeit*

Klicken Sie dazu auf die Ziffer, die Sie verstellen wollen. Danach können Sie diese mit einem Klick auf die danebenstehenden Pfeile verändern. Der Pfeil nach oben erhöht die zuvor selektierte Anzeige um eins, der Pfeil nach unten reduziert sie entsprechend. So können Sie nacheinander alle angezeigten Zahlen auf die korrekten Werte stellen.

Unter der Datums- und Zeit-Anzeige können Sie die Geschwindigkeit (*Baud Rate*) ändern, mit der die serielle Schnittstelle Daten empfangen oder senden kann. Dazu dienen wieder zwei Pfeile, mit denen dieser Wert erhöht und erniedrigt werden kann. Welcher Wert "der Richtige" ist, hängt vom Gerät ab, das Sie an diese Schnittstelle angeschlossen haben. Sehen Sie im Gerätehandbuch nach, bevor Sie hier etwas verstellen!

Die nächste Einstellungsmöglichkeit betrifft die Größe (und damit auch Lesbarkeit) von Text auf dem Bildschirm. Je nachdem, was für einen Bildschirm Sie an Ihren Amiga angeschlossen haben, wie nahe Sie dem Bildschirm sind und wie gut Ihre Augen sind, können Sie zwischen zwei Größen wählen: 60 Zeichen pro Zeile und 80 Zeichen pro Zeile. (Text mit 80 Zeichen pro Zeile wird auf einfacheren Bildschirmen oder gar auf einem Fernseher leicht unleserlich, da es dem Bildschirm an Schärfe mangelt. 60-spaltiger Text kann hingegen fast immer scharf dargestellt werden.) Klicken Sie einfach auf eine der beiden Zahlen neben dem Wort Text. Die so ausgewählte Zahl erscheint (sofern Sie nicht an den Farbreglern gespielt haben) schwarz auf orange, während die andere weiß auf blau zu lesen ist.

Der Schalter darunter (*CLI*) wird in einem späteren Kapitel noch genauer erläutert. Er bestimmt, ob das sogenannte CLI auf der Workbench sichtbar ist oder nicht.

Einen recht spektakulären Effekt können Sie mit dem großen blauen Rechteck in der Mitte des Preferences-Fensters erzielen. Dieses Rechteck dient zur Zentrierung des Bildes auf dem Bildschirm. Durch kleine Unterschiede bei den Bauteilen des Bildschirms kann es nämlich dazu kommen, daß das Bild,

das der Amiga erzeugt, nicht immer genau in der Mitte der Fläche zu sehen ist. Manchmal sind sogar Teile des Bildes durch den Rand verdeckt.

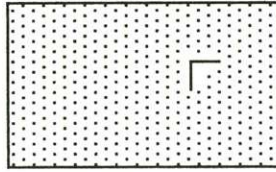


Bild 4 – 7: Zentrierung des Bildes mit der Maus

Ergreifen Sie aber den kleinen weißen Winkel in der blauen Fläche mit der Maus und bewegen dann die Maus, so folgt ihr nicht nur dieser Winkel, sondern das komplette Bild. Auf diese Weise ist es problemlos möglich, das Bild innerhalb des Bildschirms dahin zu legen, wo man es haben möchte.

Mit den beiden Schieberegler darüber können Sie die Geschwindigkeit der *automatischen Tasten-Wiederholung* und die Pause bis zu deren Beginn einstellen. Halten Sie nämlich eine der Tasten der Amiga-Tastatur fest, während Sie einen Text eingeben oder bearbeiten, so wirkt dies nach einer kleinen Pause so, als würden Sie sie schnell hintereinander immer wieder niederdrücken; die Taste wiederholt sich automatisch.

Die Geschwindigkeit, mit der die simulierten Tastendrucke ablaufen, können Sie mit dem Regler namens **Key Repeat Speed** einstellen. Je näher der Knopf in diesem Regler dem rechten Anschlag kommt, desto höher ist die Wiederholungsrate. Der Regler darüber (**Key Repeat Delay**) bestimmt, wie lange Sie eine Taste niederhalten müssen, bis die automatische Wiederholung beginnt. Je näher der Knopf in diesem Regler dem rechten Anschlag kommt, desto länger ist diese Pause.

Wie Sie diese beiden Regler einstellen, ist weitgehend persönlicher Geschmack. Falls Sie aber kein "Zehn-Finger-Tipper" sind, sollten Sie den Regler **Key Repeat Delay** nicht auf zu kleine Werte (nahe bei **Short**) einstellen, da sonst fast jede Taste gleich zwei Buchstaben erzeugt. Auch zu hohe Einstellungen für **Key Repeat Speed** führen in vielen Programmen zu verwirrenden Effekten und sind deshalb kaum zu empfehlen.

Zwei andere wichtige Einstellungen ähnlicher Art sind die "Übersetzung" der Maus-Bewegungen und die Zeitspanne, während der zwei Mausklicks als ein Doppelklick gelten. Beide Werte können Sie mit zwei Reglern am rechten Rand des Fensters einstellen.

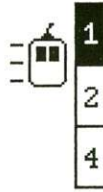


Bild 4 – 8: *Einstellung der Maus-Übersetzung*

Die Übersetzung der elektrischen Impulse, die von der Maus kommen, in die Bewegung des Zeigers am Bildschirm, geschieht ja in der Systemsoftware des Amiga. Diese Umsetzung kann deshalb auch in Software (durch Preferences) beeinflußt werden. Je höher der Wert ist, den Sie bei dem entsprechenden Schalter einstellen, desto weiter müssen Sie die Maus bewegen, um mit dem Mauszeiger eine bestimmte Strecke auf dem Bildschirm zurückzulegen.



Bild 4 – 9: *Einstellung der Doppelklick-Zeitspanne*

Für viele Operationen auf der Workbench (besonders für das Öffnen von Objekten) ist es sehr wichtig, ob zwei Mausklicks, die kurz hintereinander erfolgen, wirklich *zwei Mausklicks* oder *ein Doppelklick* sind. Ist die Zeitspanne, während der die Amiga-Software noch zwei Klicks zu einem Doppelklick zusammenfaßt, zu groß, kann es sehr leicht vorkommen, daß Objekte versehentlich geöffnet werden — was Zeit kostet. Ist diese Zeitspanne zu kurz, kann es besonders für Anfänger sehr schwierig werden, überhaupt einen Doppelklick zu erzeugen. Mit dem entsprechenden Regler im Hauptfenster von Preferences kann sich jeder Anwender diese Zeitspanne so einstellen, wie er es am angenehmsten empfindet. Je näher der Knopf in diesem Regler dem oberen Anschlag kommt, desto kürzer wird der bewußte Zeitraum. (Nur für geübte Benutzer ist eine solch kurze Zeitspanne üblicherweise geeignet.)

4.6.2 Ändern des Mauszeigers

Klickt man im Hauptfenster von Preferences auf die Taste mit der Bezeichnung **Edit Pointer** (nahe der unteren rechten Ecke), so erscheint ein neues Fenster, in dem man die Form und die Farben des normalerweise pfeilförmigen Mauszeigers ändern kann.

Auch für dieses Fenster gilt, daß wenn man es mit der Taste **Cancel** (*abbrechen*) abschließt, alle Modifikationen am Mauszeiger ignoriert werden. Verläßt man es aber mit einem Klick auf **OK**, so kommt man mit einem neuen Mauszeiger ins Hauptfenster zurück.

Die Änderung der Cursor-Farben erfolgt in ähnlicher Weise wie die Änderung der Farben für die Workbench. Drei Regler am unteren Rand des Fensters sind hierfür "zuständig". Bei einem Mauszeiger können Sie aber nur drei Farben modifizieren, die vierte ist "transparent", d.h. die darunterliegende Farbe der Workbench "scheint durch".

In einem großen Rechteck oben links im Fenster können Sie den Mauszeiger Punkt für Punkt ändern. Er wird dazu in ca. 8-facher Vergrößerung gezeigt. Wollen Sie einem Punkt im Mauszeiger eine andere Farbe geben, so klicken Sie diese neue Farbe in der Palette über den drei Schieberegler einfach an und klicken danach auf den entsprechenden Punkt in der Vergrößerung. Wollen Sie einen Punkt löschen, so klicken Sie einfach auf die "transparente" Farbe in der Palette (normalerweise ist sie blau) und klicken dann auf den zu löschenden Punkt. Wie dieser neue Mauszeiger aussieht, wenn er über den vier möglichen Farben der Workbench erscheint, können sie neben dem Vergrößerungsfeld dann auch gleich kontrollieren.

Jeder Mauszeiger besitzt auch einen *heißen Punkt*. Dies ist der Punkt, mit dem Sie "wirklich" zeigen, wenn Sie die Maustaste niederdrücken. Der ganze Zeiger ist ja für präzise Positionsbestimmungen zu vage. Beim normalen pfeilförmigen Zeiger ist es die Spitze. Auch diesen Punkt können Sie verschieben. Betätigen Sie dazu die Taste **Set Point** (engl. für *Punkt setzen*) und klicken den bewußten Punkt dann in der Vergrößerung einmal an.

4.6.3 Die Druckeranpassung

Preferences dient auch dazu, Ihren Amiga an die verschiedensten Drucker anzupassen, die Sie an ihn anschließen können. Nachdem Sie den Drucker mit einem passenden Kabel hinten an Ihren Amiga angeschlossen haben, betätigen Sie dazu im Hauptfenster von **Preferences** die Taste **Change Printer** (engl. für *Drucker anpassen*).

Auch für das Druckeranpassungsfenster gilt, daß man es mit der Taste **Cancel** (*abbrechen*) oder mit **OK** wieder schließen und die Anpassung damit beenden kann. **Cancel** bedeutet dabei *Ignorieren aller Änderungen* und **OK** *Akzeptieren der durchgeführten Anpassung*.

Welche Einstellungen Sie hier vornehmen müssen, hängt in starkem Maße von dem Drucker ab, den Sie verwenden. Konsultieren Sie deshalb das entsprechende Handbuch, das dem Drucker beilieg. Falls Ihnen die darin auftauchenden Fachausdrücke alle oder teilweise nichts sagen, können Sie die Druckeranpassung auch durch den Händler vornehmen lassen und ein für allemal auf Ihrer Workbench-Diskette abspeichern. Sie brauchen Sie im allgemeinen nie wieder zu verändern.

Vom Druckeranpassungs-Fenster können Sie allerdings noch in ein weiteres Fenster von **Preferences** gelangen, das Sie vielleicht häufiger benötigen. Wenn Sie nämlich einen grafikfähigen Drucker besitzen, können Sie dort einstellen, wie die Darstellungen am Bildschirm auf das Papier umgesetzt werden. Drücken Sie dazu im Druckeranpassungs-Fenster auf die Taste **Graphic Select** (engl. für *Grafik-Anpassung*).

In diesem Fenster können Sie insgesamt vier Einstellungen verändern: **Shade**, **Aspect**, **Image** und **Threshold**.

Bei **Shade** (engl. für *Schattierung* oder *Farbe*) haben Sie drei Möglichkeiten zur Wahl: **Black and White**, **Gray Scale** und **Color**. Die jeweils gültige davon wird in schwarz auf orange hervorgehoben. **Black and White** (engl. für *Schwarzweiß*) bedeutet, daß der Drucker beim Ausdruck jedes Bildschirm-Punktes die Entscheidung trifft, ob dieser schwarz oder weiß ausgedruckt wird. Welche Bildschirmfarben schwarz und welche weiß gedruckt werden, können Sie im Detail dann mit dem Schieberegler **Threshold** einstellen. Bei der Einstellung **Gray Scale** (engl. für *Graustufung*) versucht der Drucker, die verschiedenen Farben durch Grau-Schattierungen anzunähern. Hierbei werden aber mehrere Druckerpunkte pro Bildpunkt benötigt, weshalb bei einigen Druckern bzw. einigen sehr detaillierten Bildern Auflösung verloren geht und somit feine Details verschwimmen. Die Einstellung **Color** (Farbe) ist nur dann sinnvoll, wenn Sie einen farbfähigen Drucker angeschlossen und in der Druckeranpassung korrekt eingestellt haben. Die Farben auf dem Bildschirm erscheinen dann auch wirklich als verschiedene Farben auf dem Papier. Sie sollten jedoch selbst bei einem sehr guten und teuren Farbdrucker nicht erwarten, auch auf Papier ein ebenso brillantes Bild zu erhalten wie es auf dem Bildschirm zu sehen ist. Die heutige Drucker-Technologie ist dazu leider in keiner Weise fähig. Zudem leuchtet der Bildschirm selbst, während die Farbe auf dem Papier nur Licht reflektiert, was immer einen anderen Effekt ergibt.

Aspect (*Orientierung des Ausdrucks*) bestimmt, ob ein Bild so ausgedruckt wird, wie es im Bildschirm zu sehen ist (*horizontal*), oder ob es, um 90° gedreht, quer aus dem Drucker kommt (*vertical*) — was bei großen Bildern, die breiter als hoch sind, durchaus sinnvoll sein kann.

Bei **Image** können Sie zwischen positivem (**Positive**) und negativem (**Negative**) Ausdruck wählen. Wird positiver Ausdruck gewählt, so werden helle Flächen bei einem Ausdruck in schwarzweiß auch hell gedruckt (weiß oder hellgrau) und dunkle dunkel. Bei negativem Ausdruck wird schwarz und weiß (bzw. helles Grau und dunkles Grau) genau umgekehrt. Farbige Ausdrücke werden von **Image** nicht beeinflusst.

Der Schieberegler **Threshold** (*Schwarzweiß-Grenze*) bestimmt (bei reinem Schwarzweiß-Druck), welche Farben weiß und welche schwarz ausgedruckt werden. Je näher der Knopf in diesem Schieberegler dem linken Anschlag kommt, desto weniger Farben werden schwarz auf das Papier gedruckt, d.h., desto heller wird das Bild. Je weiter der Regler nach rechts geschoben wird, desto hellere Farben werden noch schwarz ausgedruckt und umso weniger werden weiß ausgegeben; das Bild wird dunkler. Dies gilt zumindest, wenn die Einstellung für **Image** positiv ist. Ist diese negativ, so bedeuten kleine Werte für **Threshold** ein dunkles Bild und große ein helles. Besitzen Sie einen grafikfähigen Schwarzweiß-Drucker, so müssen Sie wahrscheinlich eine Weile mit **Threshold** experimentieren, um zufriedenstellende Ergebnisse zu erreichen. Für unterschiedliche Bilder können sogar verschiedene **Threshold**-Werte nötig sein. Farbige Ausdrücke werden von **Threshold** aber nicht beeinflusst.

Damit haben Sie nun schon fast alle Programme kennengelernt, die standardmäßig mit Ihrem Amiga mitgeliefert werden. Sie sollten dadurch einen guten Eindruck davon bekommen haben, wie Amiga-Programme bedient werden. Mehr über die "allgemeinen" Eigenschaften von Amiga-Programmen (Werkzeugen) erfahren Sie im folgenden Kapitel.

5 Intuition

Wenn wir bis jetzt immer von Eigenschaften der Workbench wie Menüs und Fenstern gesprochen haben, so klang das oft, als wäre die Workbench-Software für alle diese Vorgänge verantwortlich. Dies ist nicht ganz richtig. In Wirklichkeit nutzt die Workbench — und auch die meisten anderen Programme auf dem Amiga — eine Programm-Bibliothek, die sich um alle wichtigen Aspekte der Kommunikation eines Programms mit dem Anwender (also Ihnen) kümmert. Man kann sich diese Programm-Bibliothek als eine Sammlung kleiner Programme vorstellen, die irgendwo im Speicher des Amiga liegen und bei Bedarf von anderen Programmen aufgerufen werden können. Die Programm-Bibliothek heißt *Intuition*, was nichts anderes bedeutet, als daß sie es ermöglicht, die Bedienung von Programmen auf dem Amiga möglichst intuitiv zu machen.

Intuition ist ein Teil der sogenannten *System-Software* des Amiga, die unter anderem auch die Verwaltung des RAM-Speichers, der Diskettenlaufwerke und des Druckers erledigt. Der Fachmann nennt diesen Teil *Benutzerschnittstelle* und deutet damit an, daß diese Software "an der Stelle sitzt", an der der Übergang vom Innenleben des Computers und der Programme darin zum Anwender stattfindet.

Intuition steht jedem Programm zur Benutzung zur Verfügung und erfüllt auf diese Art und Weise gleich zwei wichtige Aufgaben. Da sich die Programme nicht alle selbst um alle Details der Programm-Bedienung, wie das Verschieben von Fenstern und die korrekte Anzeige der Maus, zu kümmern brauchen, werden sie kleiner und können schneller geschrieben werden. Zugleich werden die Programme aber auch in gewissen Grenzen standardisiert, d.h. sie erledigen ähnliche Aufgaben auf ähnliche Weise, da sie gleiche Teile von Intuition dazu verwenden. Für Sie als Anwender bedeutet dies, daß Sie nicht jedesmal völlig umdenken müssen, wenn Sie ein

neues Programm bekommen. Ein Paket wie Intuition sollte deshalb nicht unterschätzt werden.

Im folgenden sollen die wichtigsten Komponenten und Eigenschaften von Intuition beschrieben werden. Vieles davon werden Sie schon aus den vorangegangenen Kapiteln kennen. Es soll an dieser Stelle aber noch einmal im kompakter Form zusammengetragen werden. Wenn Sie Schwierigkeiten mit einem der Aspekte haben, die im folgenden beschrieben sind, so können sie hier immer sehr gut nachschlagen um "Grundsätzliches" über Fenster, die Maus etc. zu erfahren, bevor Sie sich dem Handbuch des entsprechenden Programms oder anderen Kapiteln in diesem Buch zuwenden. **Sie können dieses Kapitel als eine Art Kurz-Handbuch zur Amiga-Software verwenden.** Alle grundlegenden Eigenschaften von Amiga-Programmen sind hier zu finden.

Alle folgenden Erläuterungen sind programmunabhängig, d.h. sie können Ihnen bei der Arbeit mit jedem Programm helfen, da sie beschreiben, wie sich Teile eines "typischen" Amiga-Programms verhalten. Es wird sehr genau getrennt zwischen den allgemeinen Eigenschaften, wie Intuition sie bietet, und den Beispielen in konkreten Programmen, ohne die man natürlich auch nicht auskommen kann. Bei allen Erläuterungen wird aber davon ausgegangen, daß Sie zumindest schon etwas mit der Bedienung des Amiga vertraut sind, wenn vielleicht auch nicht mit allen Details.

Denken Sie jedoch immer daran, daß kein Programm gezwungen wird, Intuition in der beschriebenen Weise zu verwenden. Prinzipiell gibt es immer viele verschiedenen Methoden, einen bestimmten Aspekt der Bedienung eines Programms zu lösen. Und es gibt immer auch Programme, die andere Wege gehen, als Intuition sie vorgibt.

5.1 Die Maus

Die Maus ist für die Bedienung des Amiga ein noch wichtigeres Werkzeug als die Tastatur. Sie wird innerhalb von Intuition (und damit innerhalb der Programme, die Intuition verwenden) für unzählige Aufgaben eingesetzt. Ohne die Maus sind Sie nahezu verloren — jedenfalls wenn Sie dann noch den Amiga bedienen wollen.

5.1.1 Funktionsprinzip der Maus

In dem kleinen Kästchen, das wir immer Maus nennen, obwohl die Ähnlichkeit mit dem entsprechenden Tier nicht gerade überwältigend ist, befindet sich eine kleine Kugel. Immer, wenn Sie die Maus auf eine glatte Oberfläche legen und sie dann bewegen, dreht sich diese Kugel und überträgt diese Drehung auf zwei kleine Rädchen im Innern der Maus. Diese Rädchen senden bei Drehung elektrische Impulse zum Amiga, aus denen ein Programm genau schließen kann, in welche Richtung und wie weit sich die Maus bewegt hat. Intuition verwendet diese Impulse dazu, einen kleinen Zeiger auf dem Bildschirm zu bewegen, den sogenannten *Mauszeiger*. Er sieht meist aus wie ein kleiner Pfeil. (Sie können ihn aber mit dem Programm Preferences nach Ihrem Geschmack ändern.)¹

Richtig gehalten wird die Maus immer so, daß ihr Schwanz (der Kabelanschluss) auf den Bildschirm weist.

Bewegt man die Maus dann nach rechts oder links, so bewegt sich auch dieser Zeiger nach rechts oder links. Bewegt man sie auf der Unterlage nach vorn, geht der Mauszeiger am Bildschirm nach oben. Und zieht man die Maus näher zu sich heran, bewegt sich der Zeiger nach unten.

Wie schnell sich der Mauszeiger über den Bildschirm bewegt, bestimmen Sie weitgehend mit der Maus. Sie können im Programm Preferences (oben rechts im Haupt-Fenster) aber auch die prinzipielle Übersetzung zwischen Mausbewegung und Zeigerbewegung wählen. Sie haben die Auswahl zwischen 1, 2 und 4, wobei größere Zahlen eine langsamere, trägere Maus bedeuten.²

5.1.2 Der Mauszeiger

Man verwendet diesen Zeiger normalerweise wie einen zusätzlichen Finger, um auf bestimmte Punkte am Bildschirm zu deuten. Bei genauen Arbeiten ist es oft wichtig zu wissen, auf welchen Punkt am Bildschirm er genau zeigt. Bei dem kleinen Pfeil, der normalerweise innerhalb der Workbench gezeigt

¹ Der Zeiger ist in Wirklichkeit ein sogenanntes *Sprite*, über das Sie im letzten Teil des Buches noch mehr erfahren können.

² Die Zahlen entsprechender Anzahl von elektrischen Impulsen, die von einem der beiden Rädchen in der Maus kommen müssen, bevor sich der Zeiger auf dem Bildschirm um einen Punkt in die angegebene Richtung bewegt.

wird, liegt dieser *heiße Punkt* an der Pfeilspitze. Ist der Mauszeiger ein kleines Kreuz — wie in einigen Grafik-Programmen — so liegt der heiße Punkt meist genau "auf der Kreuzung".

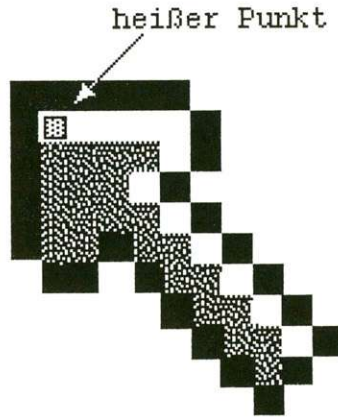


Bild 5 - 1: *Der heiße Punkt eines Mauszeigers*

Wenn Sie im Programm Preferences (natürlich mit der Maus) auf den Knopf **Edit Pointer** drücken, können Sie den Mauszeiger beliebig Punkt für Punkt ändern. Ihnen stehen dazu drei echte Farben und die "Farbe" Transparent zur Verfügung, die Sie nach Möglichkeit geschickt verwenden sollten, damit der Zeiger auch auf jedem Hintergrund immer gut zu sehen ist. (Wenn Sie mutig sind, können Sie den Zeiger auch ganz transparent machen.) Sind Sie mit Ihrem Zeiger-Design fertig, können Sie auch noch den heißen Punkt an eine beliebige Stelle setzen, indem Sie auf den Knopf **Set Point** (*Punkt setzen*) klicken und dann auf den Punkt im vergrößerten Bild des Zeigers klicken, wo er sich befinden soll. Legen Sie den heißen Punkt aber an die Position, die man bei einer bestimmten Zeigerform auch "intuitiv" für "die wichtigste" halten würde. Bei einem Pfeil z.B. an die Spitze und bei einem Kreuz oder Kreis ins Zentrum (s.o.). Ansonsten kann der Umgang mit der Maus sehr verwirrend sein. Wenn Sie z.B. bei einem Freund zum Scherz den heißen Punkt an das Ende des kleinen Pfeils verlegen, so wird er große Schwierigkeiten haben, Fenster auf der Workbench zu öffnen oder zu schließen.

Die einfache Bewegung des Zeigers mit der Maus bewirkt in den meisten Programmen aber nichts. Falls das Programm nicht ausdrücklich darum bittet, teilt Intuition ihm diese Bewegungen noch nicht einmal mit. Erst wenn einer der Knöpfe auf der Oberseite der Maus niedergedrückt wird, bewirkt dies eine Aktion.

5.1.3 Die schlafende Maus

Die Änderung der Form des Mauszeigers ist manchmal auch eine Mitteilung des Programms. Eine Änderung, die häufig vorkommt, ist der Wechsel vom Pfeil zu einer Sprechblase, in der zwei Z's erscheinen. Jeder Comic-Leser weiß, was das zu bedeuten hat: jemand schläft oder döst. In diesem Fall ist es die Maus, die schläft. d.h. der Amiga ist mit einem bestimmten Vorgang längere Zeit beschäftigt und kann nicht sofort auf Ihre Eingaben reagieren.

Andere Programme besitzen noch andere Konventionen für das Aussehen der Maus. Manchmal ändert sich der Mauszeiger, wenn man ihn über bestimmte Gebiete eines Fensters bewegt. Dies bedeutet üblicherweise, daß ein Mausklick in diesem Moment eine bestimmte Wirkung hätte, die durch das Aussehen des Zeigers angedeutet wird. Wählt man im Programm *Musicraft* z.B. eine Note aus und bewegt dann den Mauszeiger über das Notenblatt, so wird der Mauszeiger zur Note. Würde man nämlich in diesem Moment klicken, so würde diese Note auf dem Blatt erscheinen.

5.1.4 Die Maustasten

Drückt man die linke Maustaste, die auch oft *Selektions-Taste* genannt wird, nieder und läßt sie sofort wieder los, so nennt man dies einen *Mausklick*. Klickt man zweimal schnell hintereinander die Taste nieder und läßt sie wieder los, spricht man von einem *Doppelklick*. Wenn sich ein bestimmtes, von der Umgebung deutlich abgegrenztes Objekt (z.B. ein Icon) unter dem heißen Punkt des Mauszeigers befand, so gilt es durch einen Mausklick üblicherweise als selektiert oder ausgewählt. Ein Mausklick auf ein Icon heißt auch Auswahl dieses Objekts und hat üblicherweise noch keine Aktion des Programms zur Folge. Klickt man jedoch auf ein sogenanntes *Gadget* (s.u.), so bewirkt ein Mausklick meist eine sofortige Aktion des Programms.

Ein Doppelklick bewirkt oft aber auch bei Objekten, die keine Gadgets sind, eine sofortige Reaktion des Programms. In der Workbench und einigen anderen Programmen auf dem Amiga ist ein Doppelklick auf eine Objekt der universale Sesam-Öffne-Dich-Befehl. In der Workbench zeigt der Doppelklick z.B. den Inhalt von Disketten und Schubladen an und öffnet (startet) Werkzeuge und Projekte. Neue Programme für den Amiga werden sich hoffentlich an diese Konvention halten.

Drückt man die linke Maustaste nieder und bewegt dann die Maus bei festgehaltener Taste, so nennt man dies *Ziehen der Maus*. Auf diese Weise wird meist ein größerer Bereich auf dem Bildschirm selektiert oder ein Objekt

bewegt. Icons auf der Workbench können auf diese Weise z.B. verlegt werden und in vielen Grafik-Programmen können Sie rechteckige Bereiche markieren, indem Sie in die obere linke Ecke des Bereichs klicken, die Taste dann festhalten und die Maus in die rechte untere Ecke des gewünschten Rechtecks ziehen.

Die rechte Taste auf der Maus heißt *Menü-Taste* oder *Menü-Knopf*. Drücken Sie diese Taste nieder, erscheinen in der Titel-Leiste am oberen Rand des Bildschirm die Überschriften der Befehls-Menüs. Auch die Auswahl eines Befehls aus einem dieser Menüs spielt sich weitgehend mit der rechten Maustaste ab. Mehr darüber erfahren Sie weiter unten.

Die rechte Maustaste wird normalerweise nur für die Menü-Auswahl verwendet. Klickt man z.B. mit ihr auf ein Objekt oder Gadget, hat dies in der Regel keinerlei Wirkung. Nur die linke Taste bewirkt üblicherweise eine Selektion oder eine Aktion bei Gadgets. Wie bei allen Aspekten von Intuition ist aber kein Programm gezwungen, sich an diese Bedeutung zu halten. Mindestens ein Programm (Deluxe Paint, das auch in diesem Buch beschrieben wird) nutzt die Menütaste auch noch für andere Zwecke.

5.1.5 Die "Tastatur-Maus"

Sollten Sie einmal in die Verlegenheit kommen, Ihre Maus nicht gebrauchen zu können oder gebrauchen zu wollen, und müssen trotzdem den Mauszeiger bewegen, so können Sie dies auch mit der Tastatur machen. Hierzu dienen die beiden Amiga-Tasten, links und rechts neben der Leer-Taste sowie die vier Pfeil-Tasten auf der Tastatur. Die Amiga-Tasten sind die beiden Tasten mit dem roten "A", dem Symbol des Amiga.

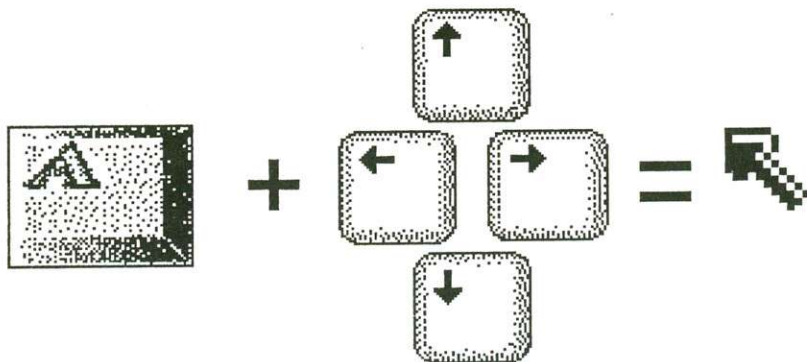


Bild 5 - 2: *Bewegung der Maus*

Halten Sie eine der beiden Amiga-Tasten (egal welche) fest und drücken dann auf eine Pfeil-Taste, so wirkt dies, als hätten Sie die Maus auf ihrer Unterlage in die angegebene Richtung bewegt. Tippen Sie einmal auf eine Pfeil-Taste, und die Maus bewegt sich einen Punkt in die angezeigte Richtung. Halten Sie die Pfeil-Taste aber längere Zeit fest, so bewegt sich die Maus mit wachsender Geschwindigkeit.

Wenn man den Mauszeiger in größeren Schritten über den Bildschirm springen lassen möchte, muß man gleichzeitig mit einer Amiga-Taste und der entsprechenden Pfeil-Taste auch noch eine der beiden Shift-Tasten festhalten. Die dann zustande kommenden Sprünge sind aber recht groß — ein Sprung entspricht etwa einer viertel Bildschirmhöhe.

Aber nicht nur die Mausbewegung läßt sich über die Tastatur simulieren. Auch die Maustasten können auf der Tastatur betätigt werden. Dazu drücken Sie, nachdem Sie den Mauszeiger zur gewünschten Stelle bewegt haben, gleichzeitig auf eine Amiga-Taste und auf eine der beiden <Alt>-Tasten. Dabei entspricht die linke <Alt>-Taste der linken (Selektions-) Taste und die rechte <Alt>-Taste der rechten (Menü-) Taste auf der Maus.



Bild 5 - 3: *Mausklick*

Die Maus auf diese Art und Weise zu simulieren ist sehr mühselig und verlangt einige Übung. Es ist im Grunde nur dann sinnvoll, wenn Ihre Maus einmal defekt sein sollte und Sie dringende Arbeiten zu erledigen haben, bevor sie repariert werden kann.

5.2 Menüs

Die Menüs, mit denen die meisten Programme des Amiga bedient werden können, sind eine sehr praktische Einrichtung von Intuition. Sie enthalten alle Befehle, die zu einem bestimmten Zeitpunkt einem Programm erteilt werden können. Sie verhindern, daß man diese Befehle vergißt, da man sie niemals einzugeben, sondern immer nur aus einer Liste (einem Menü) der möglichen Befehle auszusuchen braucht. Gleichzeitig zeigen sie dem Anwender an,

welche Befehle im Moment möglich sind und welche nicht und welche Stellung bestimmte *Programm-Schalter* gerade haben.

5.2.1 Ein Menü

Alle diese Möglichkeiten bieten die Intuition-Menüs, ohne dafür viel Platz auf dem Bildschirm zu benötigen.³ Erst wenn man sie sehen möchte, um einen Befehl zu erteilen, werden die Titel der einzelnen Menüs am oberen Bildschirmrand sichtbar. Und erst wenn man sich für eines der Menüs entschieden hat, werden kurz die einzelnen Punkte (Befehle) sichtbar und verschwinden wieder, sobald man einen davon gewählt hat.

Die Intuition-Menüs sind sogenannte PullDown- (Herunterzieh-) Menüs, da man sie erst aus der Titel-Leiste (mit der Maus) herabziehen muß, bevor sie sichtbar werden. Erst wenn man die rechte (Menü-) Taste der Maus drückt, werden die Menü-Titel sichtbar. Und erst, wenn man — bei gedrückter Menü-Taste den Mauszeiger über einen dieser Menü-Titel bewegt, klappt darunter das eigentliche Menü hervor — meist in Form eines schmalen, hellen Streifens, in dem sich mehrere getrennte Felder befinden, die sogenannten *Menü-Punkte*.

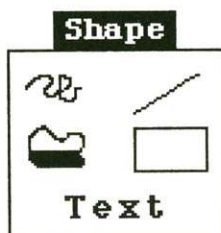


Bild 5 - 4: *Ein Menü mit 5 Menü-Punkten*

5.2.2 Menü-Punkte

Die Punkte, die ein Menü enthält, entsprechen den einzelnen Befehlen, die man in diesem Menü auswählen kann. Sie können aus einem einzelnen Wort, einem kurzen Satz oder auch einem kleinen Bild bestehen. Für jeden Menü-

³ Auch bisher gibt es schon auf anderen Computern andere Formen von Menüs. Diese sind aber manchmal nur recht umständlich und langsam zu bedienen und nehmen oft sehr viel Platz auf dem Bildschirm ein.

Punkt ist eine kleines rechteckiges Gebiet im Menü "reserviert". Diese Gebiete sind manchmal sichtbar begrenzt, meist aber ohnehin problemlos zu unterscheiden.

Bewegt man den Mauszeiger (selbstverständlich bei gedrückter Menü-Taste) in ein solches Gebiet, wird der entsprechende Menü-Punkt normalerweise *hervorgehoben*.

Es gibt allerdings mehrere verschiedene Möglichkeiten, einen Menü-Punkt hervorzuheben. Bei Menü-Punkten, die nur Text enthalten, geschieht das Hervorheben meist durch invertieren der Farben, d.h. die Farben, in denen der Text und das umgebende Gebiet am Bildschirm erscheinen, wechseln. Es gibt aber auch die Möglichkeit, daß das Gebiet des Menü-Punktes zur Hervorhebung plötzlich einen schmalen Rahmen bekommt. Und als dritte Möglichkeit kann das Aussehen des Menü-Punktes auch noch vollkommen wechseln, wenn er hervorgehoben werden soll. Dies wird üblicherweise nur bei bildhaften Menü-Punkten angewendet und auch dort nur selten.

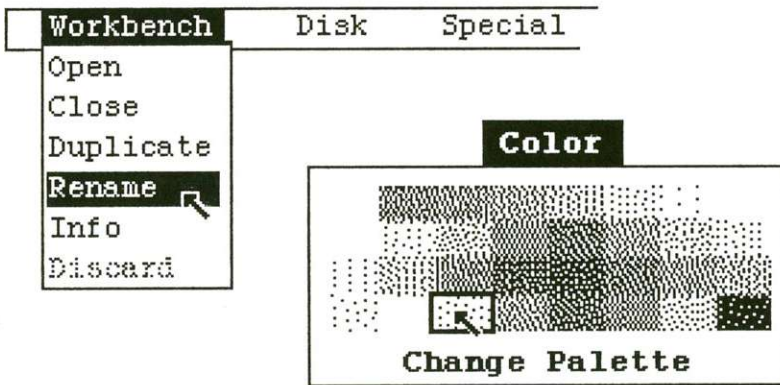


Bild 5 - 5: *Verschiedene Formen der Hervorhebung eines Menü-Punktes*

In jedem Menü ist immer nur ein Punkt hervorgehoben. Bewegt man den Mauszeiger über das Gebiet eines anderen Punktes, so wird dieser hervorgehoben und die Hervorhebung des letzten Punktes aufgehoben.

Läßt man die Menü-Taste los, während ein Menü-Punkt hervorgehoben wird, so gilt das als Auswahl dieses Punktes aus dem Menü. Eine solche *Menü-Auswahl* führt üblicherweise zu einer Aktion des Programms, entspricht also dem, was wir auf anderen Computern einen *Befehl* nennen würden.

Haben Sie es sich nach dem Herunterklappen eines Menüs anders überlegt und wollen keinen Befehl daraus auswählen, bewegen Sie den Mauszeiger

einfach ganz aus dem Rechteck des Menüs heraus und lassen dann die Menü-Taste los!

Es gibt noch eine zweite Möglichkeit der Menü-Auswahl, die bisher noch nie erwähnt wurde! Hierzu müssen Sie auf die (rechte) Selektions-Taste der Maus drücken, während sich der Mauszeiger über dem gewünschten Punkt befindet (bei gedrückter Menü-Taste also). Diese Möglichkeit ist besonders dann praktisch, wenn man schnell hintereinander mehrere Menü-Auswahlen treffen will. Im *Notepad* kann man so z.B. in einem schnellen Arbeitsgang den Text kursiv, fett und unterstrichen machen und braucht dazu nicht dreimal hintereinander die Menüs aufzurufen.⁴

Nicht alle Menü-Punkte sind zu jedem Zeitpunkt wählbar. Manchmal ist ein bestimmter Befehl in einer bestimmten Situation nicht sinnvoll. Intuition gibt dem Programm die Möglichkeit, dies dem Benutzer auch kundzutun. Hierzu können einzelne Punkte eines Menüs *deaktiviert* werden. Deaktivierte Menü-Punkte wirken üblicherweise gepunktelt bzw. etwas verschwommen. Kommt der Mauszeiger über einen deaktivierten Menü-Punkt, so wird dieser auch nicht hervorgehoben, um ganz deutlich zu zeigen, daß er nicht gewählt werden kann.

Nur wenn ein Menü-Punkt hervorgehoben ist, während die Menü-Taste gelöst wird oder die Selektions-Taste gedrückt wird, gilt er als ausgewählt!

5.2.3 Sub-Menüs

Nicht jeder Menü-Punkt kann "selbst" ausgewählt werden. Manche Menü-Punkte sind nämlich in Wirklichkeit gar keine echten Menü-Punkte, sondern der Titel eines neuen Menüs. Sobald ein solcher Menü-Punkt hervorgehoben wird, erscheint (meist rechts daneben) ein neues kleines Menü. Solche Menüs heißen *Unter-Menüs* oder *Sub-Menüs*, und die Menüs, die wir bisher kennengelernt haben, *Haupt-Menüs*. Einen Menü-Punkt, der ein Sub-Menü "hervorbringt" nennt man manchmal auch *Sub-Menü-Titel*.

⁴ Es kann Programme geben (eines gibt es schon), die diese zweite Möglichkeit der Menü-Auswahl nicht unterstützen!

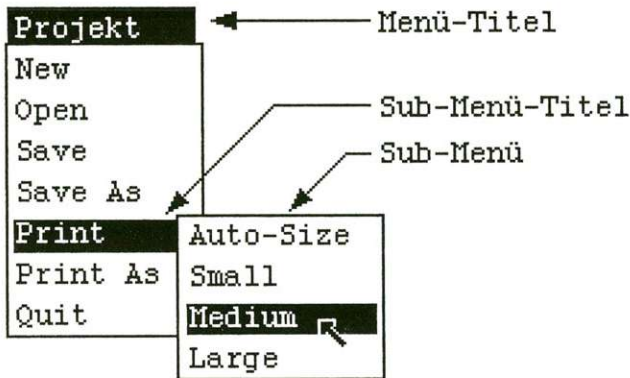


Bild 5 - 6: *Ein Sub-Menü*

Sie müssen nun den Mauszeiger in das Gebiet des Sub-Menüs bewegen und dort eine Auswahl treffen. Die Auswahl des Sub-Menü-Titels im Haupt-Menü bewirkt alleine nichts.

Ein Tip: Wenn Sie einen Punkt auswählen wollen, der in einem Sub-Menü ziemlich weit unten liegt, so unterliegt man manchmal der Versuchung, ihn direkt mit der Maus anzufahren, ohne zunächst nach rechts an den Anfang des Sub-Menüs zu gehen. Dabei verschwindet das Sub-Menü aber meistens wieder, weil man unterdessen einerseits das Gebiet des Sub-Menü-Titels verlassen hat, andererseits aber auch nicht über dem Gebiet des Sub-Menüs ist. Gewöhnen Sie sich deshalb an, bei Sub-Menüs zunächst immer mit der Maus nach rechts zu fahren und den obersten Punkt darin anzusteuern, bevor Sie andere Punkte anwählen!

5.2.4 Markierte Menü-Punkte

Manche Punkte in einem Menü entsprechen "Schaltern" des Programms, also dauerhaften Einstellungen, die einen oder mehrere Werte haben können. Die meisten Schalter enthalten die möglichen Positionen *An* und *Aus*. Das Programm *Clock* kann uns hier wieder als Beispiel dienen. Diese Uhr zeigt entweder digital oder analog an. Man kann auch sagen, der Schalter Digital steht auf *An* oder *Aus*.

Um zu zeigen, welche Stellung ein Schalter hat, werden oft die Schalter, die *An* sind, mit einem Häkchen versehen bzw. abgehakt. Wenn Schalter sich wechselseitig bedingen — wie z.B. *Analog* und *Digital* — so verschwindet das Häkchen bei einem (oder mehreren) Menü-Punkte(n), sobald es bei einem

anderen auftaucht. (Auch das *Style*-Menü des Werkzeugs *Notepad* bietet ein schönes Beispiel für dieses Verhalten.)

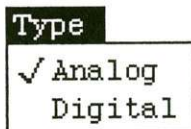


Bild 5 - 7: *Abgehakter Menü-Punkt*

Die Auswahl eines Menü-Schalters kann verschiedene Bedeutung haben. Wie sie interpretiert wird, ist Sache des zuständigen Anwendungsprogramms. Manchmal legt eine Menü-Auswahl einen Schalter um, d.h. wenn er an war, geht er aus und umgekehrt. Manchmal aber bedeutet die Auswahl stets das Einschalten und zum Ausschalten muß ein anderer Menü-Befehl ausgewählt werden. Die meisten Menü-Schalter gehören dieser zweiten Kategorie an.

5.2.5 Tastatur-Äquivalente

Menüs sind zwar recht praktisch und vor allem sehr leicht zu erlernen; es kann aber sehr lästig sein, die Maus in die Hand zu nehmen und einen Menü-Punkt auszuwählen, wenn man gerade beide Hände auf der Tastatur hat. Die Textverarbeitung ist ein typischer Fall, wo Menüs für bestimmte häufig anfallende Befehle auf Dauer einfach zu langsam sind.

Deshalb sieht Intuition die Möglichkeit vor, einen Menü-Punkt auch mit der Tastatur auszuwählen zu können. Hierzu muß man eine der beiden Amiga-Tasten festhalten und dann einen Buchstaben tippen. Dies hat dann dieselbe Wirkung, als hätte man den entsprechenden Menü-Punkt mit der Maus ausgewählt und heißt *Tastatur-Äquivalent* eines Menü-Punktes.

Welcher Menü-Punkt mit welcher Amiga-Tasten-Kombination äquivalent ist, wird im Menü selbst angezeigt. Neben dem eigentlichen Menü-Punkt erscheint dann das Symbol der Amiga-Taste und der entsprechende Buchstabe.



Bild 5 - 8: *Menü-Punkte mit Tastatur-Äquivalent*

Bei Tastatur-Äquivalenten verhalten sich die beiden Amiga-Tasten nicht gleich. Um die Zahl der zur Verfügung stehenden Möglichkeiten zu erhöhen, können auf Tastenkombinationen mit der linken Amiga-Taste andere Tastatur-Äquivalente gelegt sein als auf solche mit der rechten Amiga-Taste. Welche Amiga-Taste für welches Tastatur-Äquivalent benötigt wird, geht aber auch aus dem Menü hervor — es gibt verschiedene Symbole für beide Tasten, die dem unterschiedlichen Aussehen der linken und rechten Taste entsprechen.

5.3 Fenster

Neben den Menüs dürften die Fenster wohl das wichtigste Konzept des Amiga sein. Der geschickte Umgang mit Fenstern, z.B. die übersichtliche Positionierung vieler Fenster auf dem Bildschirm, trägt sehr viel zum zügigen Arbeiten mit dem Amiga bei.

5.3.1 Philosophie von Fenstern

Die Idee, die hinter Fenstern steckt, ist das Arbeiten mit Papier-Blättern auf einem Schreibtisch.⁵ Jedes Fenster entspricht einem Blatt Papier und kann hin- und hergeschoben, im Stapel der überlappenden Blätter nach unten gelegt oder nach oben geholt und am Ende auch ganz beiseite gelegt werden.

Zu dieser Idee, die noch nichts mit dem Namen Fenster zu tun hat, kommt als zusätzliches Konzept noch dazu, daß Fenster in ihrer Größe verändert werden können und vor allem, daß sie die Fähigkeit besitzen, einen Ausschnitt aus einem eventuell viel größeren Bild in ihrem Innern zeigen können. Sie sind also Fenster, mit denen man in dieses größere Bild hinausschaut.

Wenn ein Intuition-Fenster wirklich einen Ausschnitt aus einem größeren Bild zeigt, dann besitzt es meist auch Vorrichtungen zum Verschieben des sichtbaren Ausschnitts. Diese Funktion spielen die sogenannten Rollbalken, auf die wir weiter unten noch zu sprechen kommen.

⁵ Man hört in diesem Zusammenhang deshalb oft auch den Begriff *desktop metaphor*; zu deutsch *Schreibtisch-Methapher*.

5.3.2 Ein typisches Fenster

Ein typisches Fenster besteht aus zwei großen Bereichen: dem Fenster-Rahmen und dem Fenster-Inhalt. Intuition kümmert sich fast nur um den Fenster-Rahmen, während für das, was im Innern des Fensters sichtbar ist, im wesentlichen das dafür zuständige Anwendungsprogramm verantwortlich ist.

Der wichtigste Teil des Fenster-Rahmens ist die Titel-Leiste, in der unter anderem auch der Name des Fensters erscheint. Sie können die meisten Fenster beliebig auf dem Bildschirm verschieben, indem Sie mit der Maus diesen Namen oder die eventuell daneben sichtbare gestreifte Fläche ergreifen. Während Sie das Fenster bewegen, wird die Fläche, die das Fenster einnehmen würde, wenn die Maustaste in diesem Augenblick gelöst würde, permanent durch eine flimmernde Umrißlinie angezeigt.

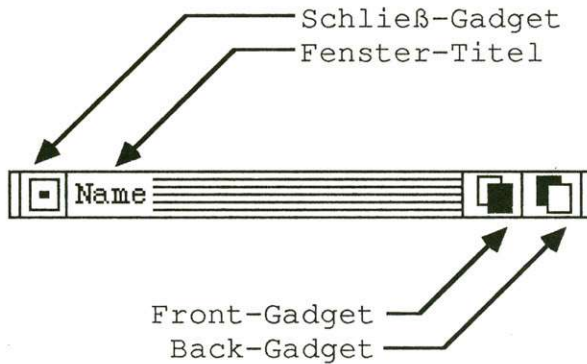


Bild 5 - 9: *Die Titel-Leiste eines Fensters*

Achten Sie beim Verschieben eines Fensters darauf, daß Sie wirklich den Namen oder die gestreifte Fläche der Titel-Leiste anklicken. Die anderen Gebiete der Titel-Leiste haben andere Funktionen, die Sie nicht versehentlich betätigen sollten.

Links neben dem Namen besitzen manche Fenster ein kleines Kästchen. Dies ist das *Schließ-Gadget*, mit dem das Fenster geschlossen bzw. vom Bildschirm entfernt werden kann. (Mehr über Gadgets erfahren Sie später in diesem Kapitel). Klicken Sie (mit der linken Maustaste) in dieses Kästchen und lassen die Taste los, solange der heiße Punkt des Zeigers innerhalb des Kästchens ist, so wird das Fenster geschlossen.

Nicht alle Fenster besitzen ein Schließ-Gadget. Fenster, die kein solches Hilfsmittel haben, können entweder gar nicht oder nur durch einen Menü-Befehl geschlossen werden.⁶

Rechts neben dem Namen des Fensters bzw. neben der gestreiften Fläche liegen zwei weitere Gadgets: das *Back* und das *Front-Gadget*. Diese beiden Gadgets dienen zur Manipulation der Reihenfolge der Fenster im überlappenden Stapel von Fenstern. Wenn sich Fenster überlappen, so kann ja nur eins davon (das oberste) komplett sichtbar sein. Alle anderen werden teilweise oder ganz verdeckt. Betätigt man das Front-Gadget eines Fensters (für die Betätigung des Front- und Back-Gadgets gelten dieselben Regeln, wie für das Schließ-Gadget), so wird dieses Fenster zum obersten im Stapel. Betätigt man das Back-Gadget, so wird das Fenster zum untersten im Stapel, was oft sogar dazu führt, daß es überhaupt nicht mehr sichtbar ist.

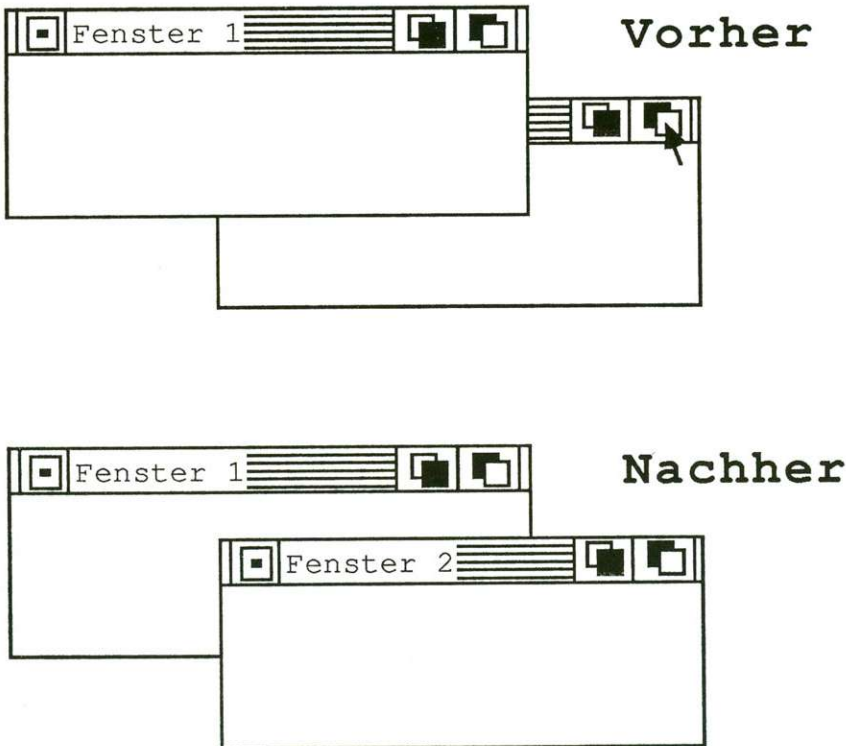


Bild 5 - 10: Vor und nach Betätigung des Front-Gadgets eines Fensters

⁶ Eine dritte Möglichkeit findet man bei den CLI-Fenstern, über die in den Kapitel, die sich mit dem CLI beschäftigen, mehr zu erfahren ist.

Unten rechts in einem Fenster befindet sich meist noch ein weiteres Gadget: das *Größen-Gadget*. Dieses Gadget wirkt nicht, wie die Gadgets der Titelleiste, als eine Art Schalter, sondern kann mit der Maus ergriffen und bewegt werden. Die Stelle, an die man es hinlegt (indem man die Maus-Taste wieder losläßt), wird danach zur neuen unteren linken Ecke des entsprechenden Fensters.

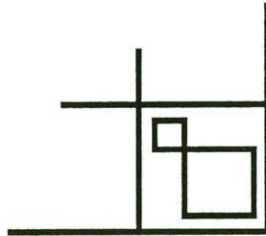


Bild 5 - 11: *Das Größen-Gadget eines Fensters (vergrößert)*

Bewegt man das Größen-Gadget, so wird die Fläche, die das Fenster einnehmen würde, wenn die Maustaste in diesem Augenblick gelöst würde, permanent durch eine flimmernde Umrißlinie angezeigt. Bei vielen Fenstern hat die Fläche eine maximale und minimale Ausdehnung, die nicht über- oder unterschritten werden kann. Erreicht man bei der Mausbewegung diese Grenzen, "stößt die Maus an" und kann den Mauszeiger nicht mehr weiter in die angegebene Richtung bewegen.

5.3.3 Rollbalken

Viele Amiga-Fenster sind Fenster, durch die man in größere Abbildungen und Texte "hinein" sehen kann und die einen Ausschnitt dieser größeren Fläche zeigen. Wie die Größe des Ausschnitts geändert werden kann (mit dem Back-Gadget) wurde im letzten Abschnitt erläutert. Mit Hilfe der *Rollbalken*, die manche Fenster am rechten und am unteren Rand enthalten, können Sie den sichtbaren Ausschnitt bewegen. Mit einem vertikalen Rollbalken am rechten Fensterrand kann man den Ausschnitt auf- und abbewegen, mit einem horizontalen Rollbalken nach links und nach rechts. Manche Fenster besitzen keine Rollbalken, manche nur einen horizontalen oder vertikalen und manche beides. Dies hängt hauptsächlich von der Art der Darstellung im Fensterinnern und ihrer horizontalen und vertikalen Ausdehnung ab. Viele Fenster, die keine Rollbalken besitzen, haben gleichzeitig eine unveränderliche Größe (kein Größen-Gadget).

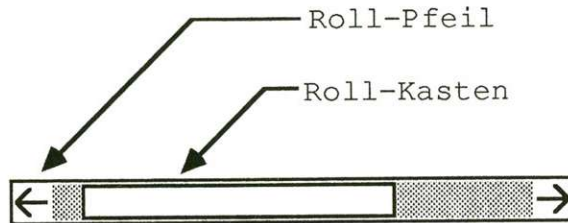


Bild 5 – 12: *Ein Rollbalken*

Ein Rollbalken besteht im wesentlichen aus drei Komponenten (Gadgets): zwei Roll-Pfeilen und einem Roll-Kasten. Die Pfeile an beiden Enden eines Rollbalkens verschieben, wenn sie betätigt (angeklickt) werden, den Bildausschnitt um ein gewisses Stück in die angegebene Richtung. Wie groß dieses Stück ist, ist Sache des Programms — meist ist es die Hälfte oder ein Drittel der Fensterhöhe bzw. -breite. Wenn Sie sich in größeren Stücken in eine Richtung bewegen wollen, so klicken sie in die blaue Fläche zwischen dem Rollpfeil, der in die gewünschte Richtung weist und dem Rollkasten. Der Ausschnitt springt dann ungefähr die volle Fensterhöhe bzw. -breite in die entsprechende Richtung.

Der Rollkasten symbolisiert den dargestellten Ausschnitt (in horizontaler bzw. vertikaler Richtung) der ganzen Fläche, die gezeigt werden könnte. Der blaue Hintergrund unter dem Rollkasten symbolisiert das Gesamtbild. Je größer der gezeigte Teil der Gesamt-Fläche ist, desto größer ist auch der Rollkasten. (Wenn das Fenster die ganze Fläche zeigt, nimmt der Rollkasten den ganzen Platz zwischen den Pfeilen ein.) Wenn die Rollpfeile betätigt werden, bewegt sich der Rollkasten auch in die angegebene Richtung. Die Breite der blauen Streifen links und rechts von ihm deutet immer an, wieviel des Bildes in dieser Richtung noch verborgen ist.

Der Rollkasten ist aber nicht nur eine Anzeige. Er kann zugleich auch selbst für das Verschieben des Bildausschnittes eingesetzt werden. Hierzu kann man ihn mit der Maus "ergreifen" und ihn, und damit den Bildausschnitt, zu der gewünschten Position schieben.

5.3.4 Fenster und Werkzeuge

Fenster können zwei Zustände einnehmen, die durch ein etwas verschiedenes Aussehen der Titel-Leiste angedeutet werden. Ist der Name des Fensters gut lesbar und die Fläche rechts daneben liniert, so ist das Fenster *aktiv*. Sind der Name und die Fläche rechts daneben jedoch gepunktelt, so ist das Fenster *inaktiv*. Ob ein Fenster aktiv oder inaktiv ist, hat vor allem im Zusammenhang

mit Werkzeugen große Bedeutung — stammen alle Fenster auf dem Bildschirm von einem einzigen Werkzeug oder der Workbench, spielt es keine große Rolle.

Intuition merkt sich zu jedem Fenster, von welchem Werkzeug es geöffnet wurde. (Ein Werkzeug kann ein oder mehrere Fenster öffnen.) Klickt man mit der Maus auf ein Fenster, so wird dieses zum aktiven Fenster.⁷ Gleichzeitig wird aber auch das Werkzeug, welches das Fenster geöffnet hat, zum aktiven Programm — sofern es dies nicht schon war. Dieses Werkzeug "übernimmt" dadurch zum Beispiel die Titel-Leiste des Bildschirms, und wenn man nun die Menü-Taste auf der Maus niederdrückt, erscheinen die Menüs des entsprechenden Werkzeugs.

5.4 Gadgets

In diesem Kapitel war schon mehrfach von *Gadgets* die Rede. Gadgets sind ziemlich elementare Komponenten von Intuition, die für eine Vielzahl von Aufgaben eingesetzt werden können, sich aber alle "irgendwie" ähneln. *Gadget* heißt übersetzt soviel wie Schalter, Gerät oder Bedienungselement, und genau darum handelt es sich auch. Es sind (meist kleine) Objekte, die vor allem auf Mausklicks und Maus-Bewegungen reagieren.

Zu jedem Gadget gehört ein genau abgegrenztes Gebiet auf dem Bildschirm. Befindet sich der heiße Punkt des Mauszeigers über diesem Gebiet und man drückt die linke Maustaste, so wird das Gadget aktiv. Ähnlich einem Menü-Punkt wird dies meist durch eine Hervorhebung des Gadgets deutlich gezeigt. Diese Hervorhebung besteht meistens aus einem Farbwechsel der einzelnen Komponenten des Gadgets. Es kann aber auch durch eine kompletten Austausch gegen ein völlig neues Bild hervorgehoben werden. Letzteres ist allerdings etwas unüblich, da so etwas meist etwas verwirrend wirkt.

Das Niederdrücken der Maustaste betätigt aber nicht unbedingt ein Gadget, d.h. es führt nicht unbedingt zu einer Reaktion des zuständigen Programms. Einige Gadgets werden erst dann betätigt, wenn man die Maustaste wieder losläßt, während sich der heiße Punkt des Mauszeigers (*immer noch oder wieder*) innerhalb des Gadget-Gebiets befindet. Andere Gadgets reagieren

⁷ Im Gegensatz zu Fenster-Systemen auf anderen Computern muß das aktive Fenster nicht unbedingt mit dem obersten Fenster im Stapel identisch sein.

allerdings bereits auf das Drücken der Maustaste. Wiederum andere Gadgets bewirken nur dann etwas, wenn man die Maus (bei gedrückter Selektions-Taste) und damit dann das Gadget bewegt.

Dies läßt sich durchaus mit Bedienungselementen vergleichen, die uns aus dem täglichen Leben bekannt sind. Manche Tipp-Tasten bzw. Sensoren an elektrischen Geräten reagieren schon auf Berührung; andere müssen niedergedrückt und wieder losgelassen werden. Bei Schiebe- und Dreh-Reglern hingegen bewirkt nur eine Bewegung dieses "Gadgets" etwas. Kombinationen der unterschiedlichen Reaktionsmuster gibt es bei Intuition-Gadgets genauso wie in der Wirklichkeit.

5.4.1 Tasten- und Sensoren-Gadgets

Einige Gadgets, die sich wie Tasten bzw. Sensoren an elektrischen Geräten verhalten, haben Sie ja bereits kennengelernt. Die Rollpfeile in Rollbalken, die Schließ-, Front- und Back-Gadgets in den Titel-Leisten von Fenstern verhalten sich wie Schalter.

Bei allen diesen Gadgets gilt aber, daß sie erst als betätigt gelten, wenn die Maustaste nach dem Klick in das Gebiet des Gadgets auch wieder gelöst wird, während sich der Mauszeiger in diesem Gebiet befindet. Wenn Sie aus Versehen ein solches Gadget angeklickt haben, gibt es also immer noch eine Chance. Bewegen Sie einfach den Mauszeiger zur Seite, und lassen Sie erst dann die Taste los.

Weiterhin sind in diese Kategorie natürlich auch als sozusagen reinstes Beispiel die echten "Tasten" aufzunehmen, die manchmal innerhalb von Fenstern auftauchen, wie SAVE und QUIT, die am unteren Rand des Fensters erscheinen, wenn der Befehl *Info* auf der Workbench aufgerufen wird.



Bild 5 - 13: *Tasten*

Aber auch die blauen Flächen zwischen dem Rollkasten und einem Rollpfeil oder links und rechts neben einem der Schieberegler in den Preferences sind eine Art Gadgets. Klickt man z.B. in einem Rollbalken in die Fläche zwischen Rollkasten und Rollpfeil, so springt der Rollkasten ein großes Stück in die angegebene Richtung. Anders als bei den bisher kennengelernten Beispielen reagieren diese Flächen schon auf das Niederdrücken der Maus -

taste. Man könnte Sie deshalb in die Unterkategorie *Sensoren* einordnen, da sie bereits bei "Berührung" betätigt werden.

5.4.2 Schiebe- und Drehregler

Die zweite wichtige Kategorie von Gadgets sind die Schiebe- und Drehregler. Schieberegler haben wir bei den bisher vorgestellten Programmen schon eine ganze Menge gesehen, selbst wenn sie nicht immer diesen Namen trugen. Drehregler sind theoretisch auch möglich, werden aber bisher von keinem existierenden Programm verwendet — wahrscheinlich, weil sie sich nicht so gut für eine Bedienung mit der Maus eignen.

Schieberegler "in Rein-Kultur" werden z.B. oft für die Farbauswahl verwendet. (Aber auch die Rollkästen in Rollbalken sind eine Art Schieberegler.) Im Haupt-Fenster des Werkzeugs *Preferences* tauchen z.B. solche Regler auf. Das folgende Bild stammt allerdings aus einem Grafikprogramm.

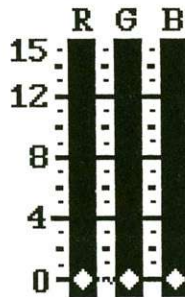


Bild 5 - 14: *Drei Schieberegler*

Schieberegler werden immer betätigt, indem man den eigentlichen *Knopf* oder *Griff* des Reglers anklickt, die Maustaste dann aber festhält und den Mauszeiger in die gewünschte Richtung verschiebt. Der Knopf folgt dabei den Bewegungen des Mauszeigers und bleibt an der letzten Position liegen, wenn die Maustaste gelöst wird. Fast alle Schieberegler begrenzen die Richtung, in die der Knopf verschoben werden kann, auf die Horizontale oder Vertikale. Selbst wenn der Mauszeiger das Gebiet des Gadgets "in der falschen Richtung" verläßt, bleibt der Knopf auf einen schmalen Streifen begrenzt.

Eine Ausnahme bildet in dieser Hinsicht der Regler, mit dem die Bildschirm-zentrierung geändert werden kann (auch im Haupt-Fenster des Werkzeugs *Preferences*). Er kann (innerhalb eines einschränkenden Rahmens) beliebig

in alle Richtungen verschoben werden und gehört damit auch in die Kategorie der Schieberegler.

Ebenfalls in diese Kategorie könnte das Größen-Gadget eines Fensters eingeordnet werden, obwohl die Analogie zu Reglern an elektrischen Geräten in der Wirklichkeit hier wohl nicht mehr funktioniert. Aber auch das Größen-Gadget (oder z.B. auch ein Icon) dient als ergriffen, sobald man darüber mit der linken Maustaste klickt. Es folgt danach jeder Mausbewegung, wobei allerdings, wie bei anderen Gadgets ja auch, Einschränkungen auf bestimmte Bewegungs-Gebiete möglich sind.

5.4.3 Text-Gadgets

Fast immer, wenn ein kurzer Text eingegeben werden muß, bedient sich ein Programm dazu eines Text-Gadgets. Text-Gadgets enthalten z.B. auch die meisten Info-Fenster auf der Workbench. Sie werden z.B. verwendet, um einen Kommentar zu einem bestimmten Werkzeug oder einer Schublade zu vermerken.



Bild 5 - 15: *Ein Text-Gadget*

Um Text in einem Text-Gadget eingeben oder ändern zu können, müssen Sie es in der Regel zuerst einmal mit der Maus anklicken. Sobald dies geschehen ist, erscheint auch die Schreibmarke, ein Buchstabe (es kann auch ein Leerzeichen sein) der in anderen Farben erscheint als seine Umgebung. Wenn Sie jetzt Text eintippen, so erscheint dieser vor der Schreibmarke und schiebt die Zeichen ab der Schreibmarke nach rechts.

Wollen Sie einen bereits in einem Text-Gadget stehenden Text ändern, so werden Sie meist die Schreibmarke erst zu der gewünschten Stelle bewegen müssen. Hierzu dienen die Pfeiltasten, die die Schreibmarke in die auf der Taste angegebene Richtung bewegen, solange es geht. Ist das Text-Gadget nur eine Zeile hoch, dann bewirken die beiden Pfeil-Tasten, die nach oben und nach unten weisen, natürlich nichts. Bei mehrzeiligen Text-Gadgets allerdings kann man mit ihnen auch die Zeile wechseln.⁸

⁸ Die Maus kann in der augenblicklichen Version von Intuition leider noch nicht für die Verschiebung der Schreibmarke eingesetzt werden. Zukünftige Versionen werden diese Möglichkeit aber wahrscheinlich bieten.

Sind Sie unter Verwendung der Pfeil-Tasten an der Stelle angekommen, wo Sie Änderungen vornehmen wollen, so können Sie dort neuen Text einfügen (s.o.) oder auch löschen. Zum Löschen dienen die beiden Tasten `<BACKSPACE>` und `<DEL>` rechts oben auf der Tastatur. `<DEL>` löscht den Buchstaben an der Schreibmarke, wobei die rechts davon liegenden Buchstaben nach links "aufrücken". Die `<BACKSPACE>`-Taste löscht den Buchstaben links der Schreibmarke, wobei die Buchstaben ab der Schreibmarke nach links aufrücken. Die `<BACKSPACE>`-Taste dient vor allem zur sofortigen Korrektur von Eingabe-Fehlern, während mit der `<DEL>`-Taste wohl meist alter Text in Text-Gadgets entfernt wird, bevor man neuen eintippt.

Ist der Text im Text-Gadget zu Ihrer Zufriedenheit ausgefallen, brauchen Sie diese Eingabe dann nur noch durch Betätigung der `<RETURN>`-Taste (die große L-förmige Taste neben den Pfeil-Tasten) oder der `<ENTER>`-Taste (unten im Zahlenblock) zu beenden.

Wenn sie häufig mit Texteingaben zu tun haben, werden Sie die Möglichkeit der *automatischen Tastenwiederholung* zu schätzen wissen. Halten Sie eine beliebige Taste nämlich längere Zeit fest, so wirkt dies nach einer kurzen Pause, als würden Sie diese Taste schnell hintereinander immer wieder betätigen. Wollen Sie z.B. größere Textstellen löschen oder die Schreibmarke große Strecken bewegen, ist diese automatische Wiederholung sehr praktisch. Wie schnell die Wiederholungs-Geschwindigkeit ist und wie lange die Pause dauert, bis die automatische Wiederholung beginnt, können Sie mit dem Werkzeug **Preferences** einstellen.

5.4.4 Kombinierte Gadgets

Viele Gadgets, die wie ein Bedienungselement aussehen, sind in Wirklichkeit aus mehreren Intuition-Gadgets zusammengesetzt. Die meisten Schieberegler bestehen z.B. aus mehreren Komponenten, von denen einige in die Kategorie *Sensoren* einzuordnen sind. Bei vielen Schieberegler kann man den eigentlichen Regler nicht nur verschieben, indem man ihn mit der Maus ergreift, sondern auch, indem man in das Gebiet des Schiebereglers neben den Knopf des Reglers klickt. Dieser springt dann ein kleines Stück auf den Mauszeiger zu. Dieses Gebiet verhält sich also wie ein Schalter bzw. Sensor.

Auch Rollbalken sind solche *kombinierten Gadgets*, und Ihnen werden in anderen Programmen bestimmt noch weitere Kombinationen dieser Art begegnen. Wenn Sie diese "neuen" Gadgets aber genau betrachten, werden Sie feststellen, daß Ihnen die Teile alle schon bekannt sind. Wenn man ein

Gadget auf diese Weise "durchschaut" hat, kann man es meist auch sofort problemlos bedienen.

5.5 Requester

Vielleicht ist Ihnen schon einmal aufgefallen, daß Gadgets üblicherweise gehäuft in einer ganz bestimmten Sorte von Fenstern auftauchen. Diese Fenster erscheinen neu auf dem Bildschirm, sobald Sie einen bestimmten Befehl aus einem Menü ausgewählt haben und verlangen eine Eingabe oder Entscheidung von Ihnen. Sobald diese erledigt ist, verschwinden sie (schließen sich) von selbst wieder.

Bei solchen Fenstern handelt es sich meist um sogenannte *Requester*. Der Name kommt daher, daß ein Programm etwas von Ihnen verlangt (*verlangen* auf englisch = *to request*), wenn ein solches Fenster auftaucht. Requester sind ebenfalls ein besonderes Intuition-Konzept, da sie sich teilweise von "normalen" Fenstern unterscheiden.

5.5.1 Erscheinen von Requestern

Requester tauchen immer dann auf, wenn ein Programm eine Entscheidung oder zusätzliche Informationen vom Benutzer benötigt. Ein Beispiel für einen sehr simplen Requester ist das einsame Text-Gadget, das erscheint, wenn man auf der Workbench den Befehl **Rename** auswählt. In diesem Fall wird ein neuer Name für ein Icon benötigt. Solche (und natürlich auch komplexere) Requester tauchen meist in der Folge der Auswahl eines Menü-Befehls auf. Einige Programme verbinden einen bestimmten Requester auch mit einem Doppelklick der Menü-Taste. Wenn Sie wissen wollen, ob ein bestimmtes Programm auf einen Menü-Tasten-Doppelklick reagiert, probieren Sie es einfach aus (die meisten Programme haben aber keinen solchen Requester).

Eine häufige Anwendung für Requester besteht auch in der Ausgabe von Fehlermeldungen. Ein neues Fenster am Bildschirm fällt auf und ist somit ideal dazu geeignet, die Aufmerksamkeit des Anwenders auf eine Fehlermeldung zu richten. Solche Fehlermeldungen müssen meist durch einen Mausklick in ein Tasten-Gadget bestätigt werden, bevor sie wieder verschwinden. Das Programm weiß dadurch, daß der Anwender die Meldung zur Kenntnis genommen hat. Manchmal haben Sie verschiedene Optionen,

nach einem Fehler fortzufahren. Ein Requester ist eine gute Möglichkeit, dem Anwender diese Optionen anzubieten, damit nicht das Programm, sondern er darüber entscheiden kann.

Viele System-Requester sind ein gutes Beispiel für die gerade beschriebene Nutzung von Requestern. Sie heißen deshalb System-Requester, weil sie nicht von einem bestimmten Anwendungsprogramm geöffnet werden, sondern von der System-Software des Amiga.

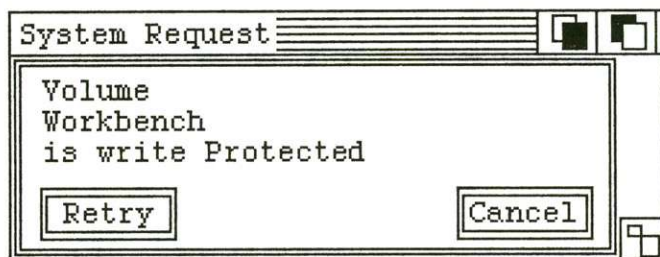


Bild 5 - 16: *Ein System-Alert*

Versucht man z.B., einen Schnappschuß von einem Objekt zu machen, während die Diskette, auf der sich das Objekt befindet, mit einem Schreibschutz versehen ist, so erscheint ein System-Requester mit entsprechender Meldung, der zugleich zwei Optionen (mit zwei Tasten-Gadgets) anbietet. Klickt man auf die Taste **Cancel**, so wird der **Snapshot**-Befehl abgebrochen. Wollen Sie aber, daß der **Snapshot**-Befehl auf alle Fälle ausgeführt wird und hatten nur vergessen, den Schreibschutz zu entfernen, so können sie die Diskette aus dem Laufwerk nehmen, den Schreibschutz beseitigen, die Diskette wieder einlegen und dann den Knopf **Retry** betätigen. (Sind sie nicht schnell genug beim Betätigen des Knopfes, versucht es der Amiga schon von selbst noch einmal.)

5.5.2 Requester und Fenster

Zunächst einmal sehen die meisten Requester aus wie ganz gewöhnliche Fenster, die allerdings manchmal auffällige Farben tragen. Sie können Schließ-, Front-, Back- und (seltener) Größen-Gadgets besitzen, und fast alle Requester können auch auf dem Bildschirm hin und her bewegt werden, indem Sie an ihrer Titel-Leiste ergriffen werden.

Genau wie bei anderen Fenstern speichert Intuition zu den Requestern auch ab, zu welchem Programm sie gehören. Klickt man deshalb in ein Requester-

Fenster, das inaktiv war, so wird das zugehörige Programm ebenfalls aktiv und übernimmt z.B. die Titel-Leiste des Bildschirms.

5.5.3 Besonderheiten von Requestern

Requester haben jedoch einige wesentliche Besonderheiten, die sie von normalen Fenstern deutlich abheben. Dies sind zunächst einmal einige Äußerlichkeiten. Außer den üblichen Gadgets, die die meisten Fenster enthalten, besitzen die meisten Requester mindestens noch ein oder zwei beschriftete Knöpfe oder Tasten. Eine Betätigung dieser Tasten schließt den Requester zumeist und teilt dem Programm gleichzeitig Ihre Entscheidung mit — sofern mehrere Tasten zur Auswahl standen.

Typisch ist oft eine Auswahl zwischen den Tasten **OK** (alles klar; weitermachen) und **CANCEL** (Abbrechen, alles vergessen). **OK** dient typischerweise zur Bestätigung aller Eingaben, die man in dem Requester gemacht hat, und **CANCEL** als Notbremse, wenn man "es sich anders überlegt hat" und der Amiga die Eingaben ignorieren soll.

Neben den Äußerlichkeiten gibt es noch einen sehr wichtigen Aspekt des Verhaltens von Programmen, solange ein Requester dieses Programms geöffnet ist. Das Programm befindet sich nämlich in einem Wartezustand, solange der Requester nicht beendet wurde. Klickt man auf ein Fenster des Programms, dann ändert der Mauszeiger seine Form zur ZZ-Sprechblase und ignoriert alle Eingaben innerhalb des Fensters. Drückt man auf die Menü-Taste, wird die Titel-Leiste des Bildschirms nur weiß und es erscheinen keine Menüs. Der Requester muß erst beendet (geschlossen) werden, bevor mit diesem Programm weitergearbeitet werden kann.

Dieses Verhalten (der Wartezustand des Programms) ist auch verständlich. Der Requester ist ja ein Zeichen dafür, daß das Programm Informationen oder Entscheidungen benötigt, bevor es fortfahren kann. Diese Entscheidung kann nicht durch einen Trick umgangen werden.

Mit anderen Programmen kann man allerdings weiterarbeiten, solange kein Requester für sie offen ist. Klickt man auf das Fenster eines anderen Programms, normalisiert sich der Mauszeiger wieder (falls er überhaupt in Schlafstellung war) und drückt man die Menü-Taste, so erscheinen auch die Menüs des anderen Programms. Ein Requester "sperrt" nur ein Programm vor der Benutzung.

5.5.4 Alerts

Eine besondere Art von Requestern sind die *Alerts*. Sie werden im Gegensatz zu den Requestern aber nur für wirklich katastrophale Fehlermeldungen verwendet, bei denen keine Rettung mehr möglich ist. Bei einem Alert wird der ganze Bildschirm oder eine Teil des Bildschirms pechschwarz und mit roter Schrift erscheint die Fehlermeldung. Das sieht recht beängstigend aus und ist es meist auch, weil es auf einen Fehler hinweist, bei dem eigentlich kein Ausweg mehr möglich ist. Es ist im allgemeinen ein Fehler des Programmierers der entsprechenden Software gewesen, der eine mögliche Programmsituation nicht berücksichtigt hat. (Regel Nummer 1 des Computer-Anwenders: Kein Programm — auch nicht das kleinste — ist fehlerfrei!)

In diesem Buch finden Sie keine Abbildung eines Alerts, aber Sie werden ihn zweifellos erkennen, sobald er ihnen begegnet. Alerts fordern Sie zumeist auf, eine der beiden Maustasten zu drücken, bevor Sie fortfahren. Bevor Sie dies tun, vergewissern Sie sich bitte, daß die Diskettenlaufwerke nicht aktiv sind und nehmen sie alle Disketten aus den Laufwerken. Drücken Sie dann auf die angegebene Maustaste.

Begegnet Ihnen in einem bestimmten Programm immer wieder ein bestimmter Alert, so ist es ganz ratsam, sich die Nummer zu notieren, die Ihnen (im Alert) hinter dem Ausdruck *Guru Meditation Number* genannt wird. Bitten Sie dann Ihren Händler, Ihnen eine funktionierende Version dieses Programms zu beschaffen und geben Sie ihm den Zettel mit dieser Zahl. Der Programmierer, der das fehlerhafte Programm geschrieben hat, kann daraus unter Umständen schließen, welchen Fehler er gemacht hat. Sie können durch das Notieren der entsprechenden Zahl eventuell dazu beitragen, daß schneller eine neue, fehlerfreie Version des Programms entsteht.

5.6 Screens

Eine letzte Möglichkeit von Intuition wurde in diesem Buch bisher nur sehr wenig erwähnt. Dies sind die *virtuellen Bildschirme* oder *Screens*.

Die Grafik-Hardware des Amige erlaubt es glücklicherweise, jedem Programm auf Wunsch einen eigenen *virtuellen* Bildschirm zur Verfügung zu stellen, über dessen Eigenschaften das Programm völlig frei verfügen kann, ohne andere Programme dadurch zu beeinträchtigen. So kann z.B. die Anzahl und natürlich der Farbton der Farben, die für Abbildungen zur Verfügung

stehen, frei gewählt werden. Auch die horizontale und vertikale Auflösung des Bildschirms kann bei unterschiedlichen Screens unterschiedlich sein, usw.. Zwischen diesen verschiedenen virtuellen Bildschirmen können Sie als Anwender entweder mit einem schnellen Tastendruck hin und her wechseln oder sie können den wirklichen Bildschirm in verschiedene Bereiche aufteilen, in denen Sie Teile der verschiedenen Screens gleichzeitig betrachten können. Wie groß der Anteil welches virtuellen Bildschirms am wirklichen Bildschirm ist, bleibt dabei weitgehend Ihrer Entscheidung überlassen.

5.6.1 Bedienung von Screens

Virtuelle Bildschirme verhalten sich im wesentlichen wie Fenster. Mit dem großen Unterschied allerdings, daß man ihre Größe nicht ändern und sie nicht horizontal verschieben kann. Mehrere Screens liegen z.B. auch in der Art eines Stapels übereinander, wobei oben liegende die darunter liegenden (teilweise) verdecken. Erst wenn der oben liegende Screen nicht mehr den ganzen Bildschirm bedeckt, ist es überhaupt möglich, andere Screens zu sehen.

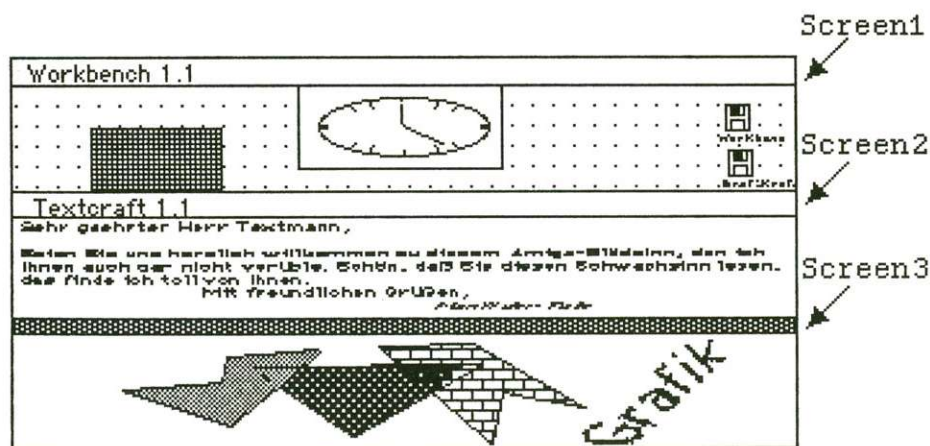


Bild 5 - 17: Mehrere Screens am Bildschirm

Genau wie bei Fenstern, kann man auch bei Screens die Reihenfolge im Stapel ändern. Hierzu dienen das Back-Gadget und das Front-Gadget am rechten Rand der Titel-Leiste eines Screens. Diese Gadgets sehen genauso aus wie die entsprechenden Fenster-Gadgets — und funktionieren auch genauso. Betätigt man das Front-Gadget eines Screens, wird dieser der

Oberste. Betätigt man das Back-Gadget, wird der entsprechende Screen der Unterste (und ist vielleicht nicht mehr zu sehen).

Da die Workbench immer ein besonders wichtiger Screen ist und einige Programme das Umlegen von Screens möglicherweise nicht korrekt durchführen, gibt es für die Workbench noch besondere Tastenkombinationen, die Sie entweder ganz nach vorne oder ganz nach hinten legen — egal ob die dafür eigentlich zuständigen Gadgets gerade sichtbar sind oder nicht.

Hält man die linke Amiga-Taste fest und drückt dann die Taste <N>, so kommt der Workbench-Screen im Stapel nach oben. Hält man die linke Amiga-Taste fest und drückt auf <M>, so wird er ganz nach hinten gelegt. Gerade die erste Tastenkombination (A + <N>) ist sehr wichtig, da sie Ihnen eine Möglichkeit verschafft, "auf alle Fälle" zurück in die Workbench zu gelangen.

5.6.2 Mögliche Probleme mit Screens

Wenn Sie ein Programm starten, das einen eigenen Screen benötigt, so sorgt das Programm meist von selbst dafür, das dieser nach dem Start der oberste, also komplett sichtbar ist. Falls das Programm dies aber nicht tut, nach dem Programmstart also scheinbar keine Veränderung am Bildschirm zu sehen ist, sollten Sie versuchen, den Workbench-Bildschirm (mit dem entsprechenden Gadget) nach hinten zu legen. Falls keine weiteren virtuellen Bildschirme existieren, wird der des neuen Programms dadurch sichtbar.

Wenn die Front- und Back-Gadgets eines Screens scheinbar nicht funktionieren, hat der Programmierer des entsprechenden Programms wahrscheinlich einen Fehler gemacht. Bei einigen Demo-Programmen können z.B. solche Effekte auftreten. Versuchen Sie dann mit den entsprechenden Tasten-Kombinationen (A + <N> oder A + <M>), die Workbench nach vorne zu holen bzw. nach hinten zu legen.

Scheint auch A + <N> die Workbench nicht sichtbar zu machen, so kann es sein, daß der Workbench-Screen auf dem Bildschirm ganz unten liegt und deshalb nur ein kleiner Streifen davon sichtbar ist. Gehen Sie mit der Maus dann am Bildschirm so tief herunter wie möglich, drücken Sie die linke Maus-Taste und bewegen Sie die Maus dann wieder nach oben. Das sollte, von ganz wenigen Ausnahmen abgesehen, die Workbench zum Vorschein bringen.

Das größte Problem, das mit Screens verbunden ist, ist der Speicherplatz. Obwohl vielleicht nur ein Teil eines Screens auf dem Bildschirm sichtbar ist, benötigt er doch immer soviel Speicher wie ein kompletter Bildschirm mit den entsprechenden Attributen benötigen würde. Dies kann wenig sein, wenn der entsprechende Screen nur wenige Farben zeigt und/oder eine geringe Auflösung hat. Bildschirme, die viele Farben gleichzeitig zeigen, können aber bis zu einem Viertel des Speichers benötigen, den der Amiga mit 512 KByte RAM besitzt. Auf dem Amiga mit nur 256KByte RAM kann man solche Programme oft überhaupt nicht starten.

Seien Sie deshalb immer vorsichtig mit Programmen, die eigene Bildschirme mit vielen Farben benötigen. Versuchen Sie nicht, mehr als ein oder maximal zwei solcher Programme gleichzeitig aktiv zu haben!

6

Arbeiten mit dem Amiga

Im diesem Kapitel sollen einige der ersten echten Anwendungsprogramme, die bereits für den Amiga vorhanden sind, vorgestellt werden.

Dies sind einmal die Programme *Graphicraft*, *Textcraft* und *Musicraft* für das Zeichnen, Textbearbeiten und Musizieren mit dem Amiga, die alle von Commodore selbst vertrieben werden. Dazu kommt noch *Deluxe Paint* vom amerikanischen Softwarehaus *Electronic Arts*, das ebenfalls für die Erstellung von Bildern und Grafiken gedacht ist.

Graphicraft und Deluxe Paint werden am ausführlichsten behandelt, weil sie einen wichtigen Aspekt der Fähigkeiten des Amiga sehr gut demonstrieren und zum anderen die beiden Programme sind, die schon in relativ endgültigen Versionen zum Test vorlagen, als dieses Buch geschrieben wurde. Von Textcraft und Musicraft konnten nur Vorversionen (mit teilweise noch eingeschränktem Funktionsumfang) getestet werden. (Über Textcraft und Musicraft hinaus gibt es bereits eine ganze Reihe von Produkten, die in den USA bereits vertrieben werden, die aber nicht rechtzeitig zur Fertigstellung dieses Buches beschafft werden konnten.)

6.1 Graphicraft

Graphicraft stammt vom amerikanischen Software-Haus Island Grafics, wird aber direkt über Commodore vertrieben. Jeder Amiga-Händler sollte ihnen deshalb ein Exemplar beschaffen können. Es kann sogar sein, daß Commodore dieses Programm dem Amiga kostenlos beilegt; dies ist aber im

Augenblick noch nicht sicher. Graphicraft kann am ehesten als ein Malprogramm der Einstiegsklasse bezeichnet werden. Es ist von den Möglichkeiten vergleichsweise beschränkt, aber eben auch recht preiswert (falls nicht sogar umsonst).

Auch wenn Sie sich gleich das leistungsfähigere Deluxe Paint angeschafft haben, sollten Sie die folgenden Abschnitte über Graphicraft sorgfältig durchlesen. Es werden einige der "grundsätzlichen" Eigenschaften der Amiga-Grafik darin erläutert werden, die bei den Beschreibungen von Deluxe Paint nicht noch einmal wiederholt werden.

6.1.1 Grundzüge und Konzepte von Graphicraft

Mit Graphicraft (und auch mit Deluxe Paint) können Sie eine fast bildschirmgroße Zeichnung erstellen, als wäre ihr Monitor eine Maler-Leinwand. Eine etwas merkwürdige Leinwand allerdings, denn Sie müssen sich beim Malen an ein — glücklicherweise recht feines — Raster von 320 Punkten horizontal mal 200 Punkten vertikal halten. Jedes dieser Kästchen ist auf dem Monitor 1080 ungefähr 1 Millimeter hoch und 1 Millimeter breit (in Wirklichkeit sind sie etwas rechteckig und nicht genau quadratisch, aber so genau wollen wir es hier nicht nehmen). Wenn Sie nahe an den Bildschirm herangehen, können Sie diese Kästchen oder Pünktchen sogar erkennen. Horizontal gehen sie ineinander über und Sie können zwei benachbarte nur dann unterscheiden, wenn sie unterschiedliche Farben tragen. Zwei übereinanderliegende Pünktchen kann man aber immer recht gut auseinanderhalten. Treten Sie wieder etwas vom Bildschirm zurück, so verschwimmen die kleinen Kästchen wieder zu Linien, Rechtecken und Kreisen. In diese Kästchen ist die Leinwand aufgeteilt. Kleinere Details können Sie nicht bearbeiten.

Jedes dieser kleinen Rechtecke kann eine Farbe aus einer Palette von insgesamt 32 verschiedenen Farben tragen. Diese 32 Farben können Sie sich selbst "zusammenmischen"; Sie sind nicht auf eine bestimmte Auswahl festgelegt!

Natürlich werden Sie im allgemeinen nicht jeden Punkt einzeln mit einer Farbe versehen wollen, sondern eine Zeichnung zunächst mit groben Pinselstrichen konzipieren und dann allmählich mit einem feineren Pinsel die Details bearbeiten. Graphicraft erlaubt ihnen genau diese Vorgehensweise. Sie nehmen einen bestimmten Pinsel, tauchen ihn in einen Farbtopf und malen los. Nur können Sie natürlich nicht mit einem wirklichen Pinsel malen. Der Pinsel wird Ihnen, genau wie auch das Bild, am Bildschirm gezeigt und

Sie bewegen ihn mit der Maus. Genau, wie man in der Workbench mit dem Zeiger auf eine Diskette oder eine Schublade deuten kann, bewegt man in Graphicraft den Pinsel. Diese Bewegung bleibt aber solange ohne Folgen, bis man den linken Mausknopf drückt. So lange schwebt der Pinsel praktisch über der Zeichnung. Erst indem sich der linke Mausknopf senkt, senkt sich auch der Pinsel auf die Leinwand.

Auch der Pinsel verhält sich etwas anders als sein wirkliches Gegenstück. Er kann nicht nur verschiedene Stärken, sondern auch ganz verschiedene Formen haben. Es stehen ihnen kreisförmige, rechteckige, ganz feine punktförmige und auch unregelmäßig geformte Pinsel zur Verfügung. Sie können sich sogar eigene Formen erzeugen. Wenn Sie z.B. einen Pinsel in Form eines Fußabdrucks haben und damit einen "Klecks" auf die Leinwand machen, erscheint eben ein Fußabdruck auf der Leinwand.

Neben dem "freien Malen" mit dem Pinsel besteht zusätzlich noch die Möglichkeit, garantiert gerade Linien oder Rechtecke zu zeichnen, beliebig geformte Flächen mit einer bestimmten Farbe zu füllen und Texte in das Bild einzufügen. Für feine Detail-Arbeiten schließlich gibt es ein Vergrößerungsglas, mit dem Sie einen Ausschnitt ihres Bildes um den Faktor 4 vergrößern und in dieser Vergrößerung dann auch wirklich mühelos jedem einzelnen Punkt die gewünschte Farbe geben können.

6.1.2 Überblick über die Menüs von Graphicraft

Zu Ihrer Orientierung folgt zunächst eine kurze Zusammenfassung aller Menü-Befehle von Graphicraft. Sie werden im Anschluß daran näher erläutert. Für diesen Test stand uns leider nur die englische Version des Programms zur Verfügung. Die Befehle sind deshalb natürlich alle englisch. Deutsche Übersetzungen, die den englischen Begriffen hoffentlich recht nahe kommen, sind aber in Kursivschrift hinter diesen angefügt worden.

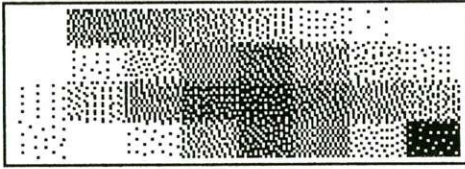
Ein paar Worte zum Aufbau der Befehlsliste: die fettgedruckten, unterstrichenen Worte sind immer die Titel eines Menüs. Es sind die Worte, die in der Titel-Leiste des Bildschirms erscheinen, wenn Sie die rechte Maustaste drücken. Die fettgedruckten und etwas eingerückten Texte unter dem Menü-Titel sind die einzelnen Menü-Punkte, die man sieht, wenn man mit dem Mauszeiger bei gedrückter Menü-Taste auf den Menü-Titel fährt. Noch weiter eingerückte Texte sind die Punkte von Sub-Menüs, die in dem Moment erscheinen, in dem sich der Mauszeiger über dem entsprechenden Punkt des Haupt-Menüs befindet. Die Texte in Klammern hinter den eigentlichen Menü-Texten sind knappe Erläuterungen der Bedeutung der

entsprechenden Befehle (meist nur eine deutsche Übersetzung). Bei Menü-Punkten, die nicht ein einfacher Text sind, wird meist auch die entsprechende Grafik, manchmal aber nur eine verbale Umschreibung davon gezeigt.

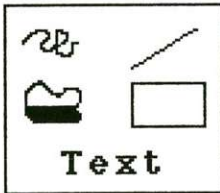
<u>Project</u>	(<i>Projekt bzw. Bild</i>)
New	(<i>Neues Bild</i>)
Open	(<i>Öffnen bzw. Einlesen</i>)
Painting	(" eines <i>Bildes</i>)
Brushes	(" eines <i>Pinzel-Satzes</i>)
Save	(<i>Bild unter alten Namen sichern</i>)
Save As	(<i>Bild oder Pinzel-Satz sichern</i>)
Revert	(<i>Zurück zur letzten Version</i>)
Show Title Bar	(<i>Titelzeile einblenden</i>)
Hide Title Bar	(<i>Titelzeile ausblenden</i>)
Print	(<i>Bild drucken</i>)
Quit	(<i>Programm verlassen</i>)
<u>Edit</u>	(<i>Editieren bzw. Modifizieren</i>)
Undo	(<i>Letzten Befehl rückgängig machen</i>)
Frame	(<i>Bereich markieren bzw. umrahmen</i>)
Cut	(<i>Bereich ausschneiden</i>)
Copy	(<i>Bereich kopieren</i>)
Paste	(<i>(kopierten) Bereich einsetzen</i>)
Erase	(<i>Bereich löschen</i>)
Erase Screen	(<i>Bildschirm löschen</i>)
<u>Special</u>	(<i>Spezial bzw. Einstellungen</i>)
Magnify	(<i>Vergrößerungsglas ansetzen</i>)
Pick Position	(" bei einer bestimmten Position)
Continous	(" mit dem Zeiger wandernd)
Mirrors	(<i>Spiegelachsen definieren</i>)
Mirrors Off	(<i>Spiegelachsen deaktivieren</i>)
Cycle Draw	(<i>Farbrotation des Pinsels anschalten</i>)
Cycle Draw Off	(<i>Farbrotation des Pinsels ausschalten</i>)
Cycle Colors	(<i>Farbrotation einschalten</i>)
Cycle Colors Off	(<i>Farbrotation ausschalten</i>)

Color

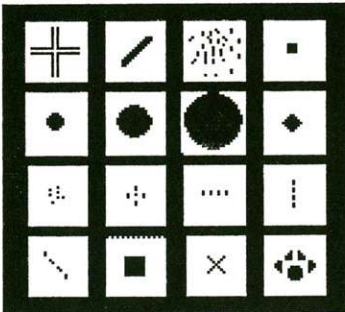
(Farbe)

**Change Palette** (Farbpalette ändern)**Shape**

(Form)

**Brush**

(Pinsel)

**Custom**

(eigene Pinselform editieren)

Kommen wir nun zu den einzelnen Menüs im Detail.

6.1.2.1 Das *Projekt*-Menü

Die Befehle im Projekt-Menü dienen dazu, das ganze Bild zu verwalten. Der Begriff *Project* meint beim Amiga ja im allgemeinen immer ein Dokument, das mit einem Programm erstellt wurde. Im Fall von Graphicraft sind Projekte Zeichnungen. Die Befehle im Projekt-Menü betreffen immer das

gerade bearbeitete Projekt. Von diesem existieren meistens zwei verschiedene Versionen: eine auf der Diskette/Festplatte und eine im Hauptspeicher des Amiga, die sie gerade bearbeiten. Wenn Sie *Graphicraft* über ein Projekt starten oder ein Bild (mit dem Befehl *Open*) einlesen, stimmen beide Versionen überein. Sobald aber auch nur ein Punkt an diesem Bild geändert wurde, gibt es zwei Versionen. Die Änderungen an einem Bild werden erst dann wirklich permanent, wenn Sie den Befehl *Save* benutzen. Bis dahin können Sie es sich immer anders überlegen.

Der Name des aktuellen Projektes (Bildes) erscheint stets in der linken Ecke der Titel-Leiste. Nur falls ein neues Bild gemalt wird, steht dort stattdessen der Name des Programms (*Graphicraft*) um damit anzudeuten, daß noch kein Projektname vergeben wurde. Der Befehl *New* leitet z.B. das Malen eines neuen Bildes ein. Das alte wird vorher gelöscht. Der Befehl *Open* liest Daten von der Diskette bzw. Festplatte ein. Dies kann entweder ein komplettes Bild (Befehl *Painting* im Sub-Menü) oder ein Satz von Pinsel-Formen (Befehl *Brushes* im Sub-Menü) sein.

Entsprechend dienen die Befehle *Save* und *Save As* dazu, das komplette Bild oder einen Satz Pinsel auf der Diskette abzuspeichern. Bei *Save* wird das Bild unter dem alten Namen abgespeichert (und auch in derselben Schublade). Die letzte auf Diskette gespeicherte Version geht dabei natürlich verloren. *Save As* hat zwei Unterpunkte. *Save As Painting* speichert das aktuelle Bild unter einem neuen Namen ab, den man in einem Requester eingeben kann. *Save As Brushes* sichert den aktuellen Pinsel-Satz ebenfalls unter einem wählbaren Namen auf der Diskette ab. *Revert* ist ein sehr praktischer Befehl, wenn Sie eine Reihe von Änderungen an einem Bild durchgeführt haben und es sich dann "anders überlegen". *Revert* kehrt immer zurück zu der Version des Bildes, die zuletzt gesichert wurde.

Show Title Bar und *Hide Title Bar* sind rein "kosmetische" Funktionen. Sie blenden die Titel-Leiste ein, bzw. verbergen sie. Erst bei verborgener Titel-Leiste ist es ja möglich, den ganzen Bildschirm zu bemalen. Auch bei verborgener Titel-Leiste werden die Menüs aber in dem Moment sichtbar, da die Menü-Taste gedrückt wird.

Print druckt Ihr Gemälde auf einem angeschlossenen Drucker. Falls Sie allerdings nicht stolzer Besitzer eines Farbdruckers sind, dürften Sie etwas enttäuscht sein. Mit nicht farbfähigen Druckern versucht der Amiga, durch Graustufen einen ungefähren Eindruck des Bildes wiederzugeben. Sie sollten sich aber deshalb daran gewöhnen, meist nur einen recht müden Abklatsch des Bildschirms auf Papier zu bekommen. Dies gilt leider auch, aber in nicht ganz so starkem Maße, für die preiswerteren Farbdrucker (z.B. von Oki und Epson). Die Brillanz des Farb-Monitors ist kaum auf eine dauerhafteres Medium (sprich: Papier) zu übertragen.

Quit schließlich verläßt das Programm. Dazu ist kaum noch mehr zu sagen. Wie jedes wirklich freundliche Programm fragt Graphicraft Sie aber vor jeder *schwerwiegenden* Operation (z.B. **Quit**) mit einem Requester um Bestätigung.

6.1.2.2 Das *Edit*-Menü

Das **Edit**-Menü stellt Ihnen "grobe" Operationen zum Ändern (Editieren) des Bildes zur Verfügung. Zentral für die meisten Befehle des **Edit**-Menüs ist der gerade *ausgewählte Bereich*. Mit dem Befehl **Frame** (s.u.) kann man einen rechteckigen Bereich aus dem Bild auswählen und diesen löschen, duplizierten, etc.

Der Befehl **Undo** steht an erster Stelle im **Edit**-Menü und ist auch besonders wichtig. **Undo** gibt Ihnen — ähnlich wie **Revert**, aber in kleinerem Maßstab — die Möglichkeit "es sich noch einmal anders zu überlegen". Haben Sie eine Änderung am Bild durchgeführt und stellen fest, daß das Ergebnis nicht Ihren Wünschen entspricht, so rufen Sie einfach **Undo** auf. In vielen Fällen wird die letzte Änderung dann rückgängig gemacht. Das ist doch wesentlich komfortabler als ein Arbeiten mit Farbe und Leinwand, oder?

Manche Änderungen kann **Undo** allerdings nicht rückgängig machen. Dann müsste man auf die traditionelle Vorgehensweise des Überpinselns und Neumalens zurückgreifen. Würde man wirklich mit Farbe und Leinwand arbeiten, bliebe einem schließlich auch nicht anderes übrig.

Frame ist der zentrale Befehl des Edit-Menüs. Nachdem man **Frame** aufgerufen hat, kann man den zu bearbeitenden Bereich aus dem Bild auswählen. Hierzu klickt man in eine Ecke des auszuwählenden Rechtecks, bewegt dann die Maus in die diagonal gegenüber liegende Ecke und klickt dort noch einmal (beidesmal mit der linken Taste natürlich). Während die Maus nach dem ersten Klick bewegt wird, wird der Bereich durch ein flimmerndes Rechteck mit dünnem Rand hervorgehoben. Diesen Bereich kann man dann mit den folgenden Befehlen bearbeiten.

Cut (engl. für *ausschneiden*) legt den markierten Bereich in eine Art Zwischenablage und füllt ihn danach mit der aktuellen Hintergrundfarbe (vgl. den Abschnitt über das **Color**-Menü). **Copy** (engl. für *kopieren*) legt ihn nur in die Zwischenablage, läßt das Bild aber unverändert. **Paste** (engl. für *einkleben, einsetzen*) setzt eine Kopie dieser Zwischenablage wieder ins Bild ein. Diese Kopie besitzt wie ein ausgewählter Bereich einen dünnen Rahmen und kann mit der Maus ergriffen und zu einer beliebigen Stelle im Bild bewegt werden. Dazu klickt man einmal innerhalb des schwarzen Rahmens und hält den Knopf dann fest. Der gesamte Bereich folgt der Maus so lange,

bis der Knopf losgelassen wird. **Paste** kann man auch mehrmals hintereinander anwenden und so sehr schnell und einfach mehrere Kopien ein und desselben Bildteils erzeugen.

Erase (engl. für *löschen*) löscht den markierten Bereich und **Erase Screen** den ganzen Bildschirm. "Löschen" heißt bei Graphicaft dabei immer Füllen mit der Hintergrundfarbe (der ersten Farbe auf der Palette)

6.1.2.3 Das *Special*-Menü

Wie der Name schon sagt, ist das **Special**-Menü für die Spezial-Effekte bei Graphicaft zuständig. Hierzu gehören das Vergrößerungsglas, die Spiegelachsen und die automatische Farbenänderung beim Zeichnen.

Der Befehl **Magnify** aktiviert das Vergrößerungsglas von Graphicaft. Dies ist ein separates Fenster, in dem ein Teil des Bildes in vierfacher Vergrößerung dargestellt wird. Das Fenster kann in der Größe nach Wunsch verändert werden. (Wenn Sie einen Amiga mit nur 256 KByte Speicher verwenden, kann es aber nicht sehr groß werden.) Mit dieser Vergrößerung ist es natürlich viel leichter, einzelne Punkte mit einer bestimmten Farbe zu versehen, als bei der "normalen" Darstellung. Innerhalb des Vergrößerungs-fensters kann man aber leider nur einzelne Punkte mit einer bestimmten Farbe belegen. Weder das **Edit**-Menü noch die verschiedenen Pinsel-Formen sind hier wirksam.

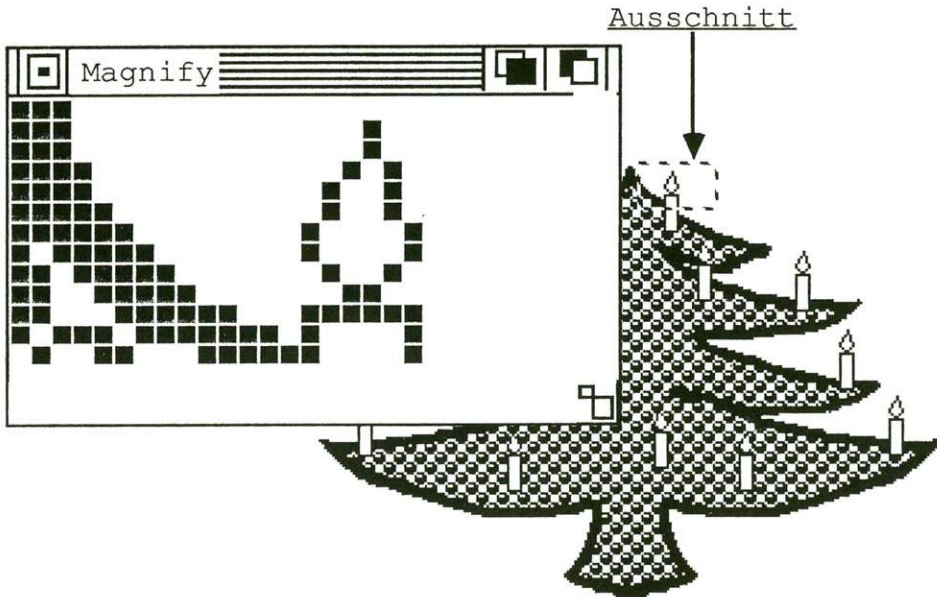


Bild 6 - 1: Das Graphicraft-Vergrößerungsglas

Die beiden Unterpunkte von **Magnify** dienen verschiedenen Zwecken des Vergrößerungsglases. Wählt man **Pick Position** (engl. für *Position aufnehmen*), so folgt der im Vergrößerungsglas dargestellte Ausschnitt zunächst dem Mauszeiger bzw. Pinsel. Klickt man dann aber mit der linken Taste, so bleibt der Ausschnitt konstant und man kann ihn im Vergrößerungsfenster bearbeiten. Wählt man aber **Continuous** (engl. für *kontinuierlich*), so folgt der vergrößerte Ausschnitt immer der Maus. Auch wenn man andere Operationen durchführt, so sieht man immer den Ausschnitt vergrößert, über dem sich gerade der Mauszeiger bzw. Pinsel befindet. **Continuous** ist deshalb nicht für Detailarbeiten am Ausschnitt geeignet, sondern vor allem für ein rasches Betrachten verschiedener Bilddetails.

Das Vergrößerungsglas zeigt übrigens auch immer die Teile des Bildes korrekt an, die von der Titel-Leiste (oder vom Vergrößerungsglas-Fenster selbst) verdeckt werden. Dazu muß man den Mauszeiger nur während **Pick Position** (oder **Continuous**) über die Titel-Leiste bewegen. Mit dem Vergrößerungsglas können die verdeckten Bildteile also auch bearbeitet werden, während die Titel-Leiste sichtbar ist.

Das Vergrößerungsglas können Sie wieder abschalten, indem Sie das entsprechende Fenster schließen (mit dem Schließ-Gadget) oder den Befehl **Magnify Off** aufrufen.

Der **Mirror**-Befehl (engl. für *Spiegel*) erlaubt es, relativ leicht symmetrische Abbildungen zu erstellen. Ruft man ihn auf, so erfragt ein Requester, wieviele horizontale, wieviele vertikale Spiegel man möchte und ob es diagonale Spiegel geben soll. Schließt man den Requester mit **OK!** ab, so werden diese Spiegel (-Achsen) wirksam und vervielfachen nun jeden Pinselstrich, den man im folgenden macht. Der Pinselstrich wird dabei immer an jeder der Achsen gespiegelt (im mathematischen Sinn einer Spiegelachse). Die Wirkung von vielen Spiegelachsen ist vorher nur sehr schwer abzuschätzen. Die beiden folgenden Abbildungen versuchen, einen Eindruck davon zu vermitteln, wie einige wenige Spiegelachsen wirken. Der schwarze Strich ist jeweils der "wirkliche" Pinselstrich. Die grauen sind die von Graphicraft erzeugten gespiegelten. Die aktiven Spiegelachsen sind hier als dünne gestrichelte Linien angedeutet. Am Bildschirm sieht man sie nicht.

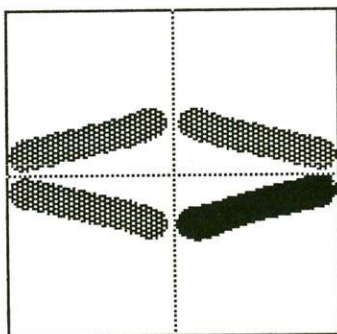


Bild 6 - 2: *Zeichnung mit einer horizontalen und vertikalen Spiegelachse*

Die folgende Abbildung zeigt denselben Pinselstrich wie die vorangehende. Nur sind zusätzlich noch die diagonalen Spiegelachsen wirksam.

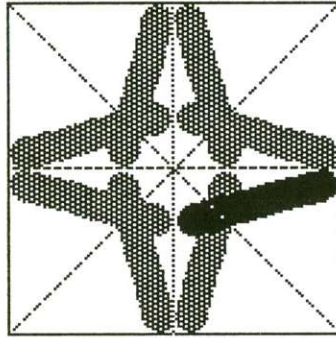


Bild 6 - 3: *Zeichnung mit zusätzlichen diagonalen Spiegelachsen*

Mehrere horizontale oder vertikale Spiegelachsen testen Sie am besten selbst aus — sie sind schwer in allgemeingültiger Form darzustellen. Ihre Wirkung ist auch vorher sehr schwer abzuschätzen. Spielen Sie einfach bei Interesse ein wenig damit. Und denken Sie immer daran: der **Undo**-Befehl hebt bei unerwarteten Effekten die Wirkung des letzten Pinselstrichs, ob gespiegelt oder nicht, wieder auf!

Der Befehl **Mirrors Off** macht die vorher definierten Spiegelachsen wieder unwirksam. Pinselstriche verhalten sich dann wieder so wie erwartet.

Die letzten Befehle im **Special**-Menü werden gesondert im Abschnitt *Farbpaletten-Animation* behandelt. An dieser Stelle wollen wir deshalb nicht weiter darauf eingehen.

6.1.2.4 Das *Color*-Menü

Das Besondere am Amiga sind ja unter anderem seine faszinierenden Farbfähigkeiten. Das **Color**-Menü hilft Ihnen, diese zu nutzen. Es ist praktisch Ihre Farbpalette, in die Sie den Pinsel tauchen, bevor Sie zu malen beginnen. Das Aufrufen eines Befehls aus dem **Color**-Menü wirkt dabei wie das Eintauchen des Pinsels. Glücklicherweise verbraucht sich die Farbe in einem Amiga-Pinsel nie; Sie brauchen ihn erst dann wieder in eine Farbe eintauchen, wenn Sie die Farbe wechseln wollen.

Wenn Sie das Color-Menü "herunterklappen", so sehen Sie insgesamt 32 verschiedene Farben und den Befehl **Change Palette**. Geht man mit der Maus über eine der Farben und läßt die Menü-Taste dann los, so erhält der Pinsel diese Farbe. Der Befehl **Change Palette** tut genau dies: er ändert die gezeigte Palette (engl. *change palette*) und damit die Farben, die zum Malen zur Verfügung stehen.

Bei Aufruf des Befehls **Change Palette** erscheint ein Requester, in dem wieder dieselbe Palette zu sehen ist und darüber die drei Regler **R**, **G** und **B**, die wir ja schon aus dem Programm **Preferences** kennen. Klickt man eine Farbe an, so wird sie zur Bearbeitung ausgewählt. Durch Verschieben der drei Regler kann man dann die Rot-, Grün- und Blau-Anteile ändern, die dieser Farbe beigemischt werden.

Es ist nicht gerade sehr einfach, eine bestimmte Farbe aus diesen drei Bestandteilen zusammenzumischen. Mit eine wenig Herumprobieren und etwas Übung geht es aber dann meist recht schnell.

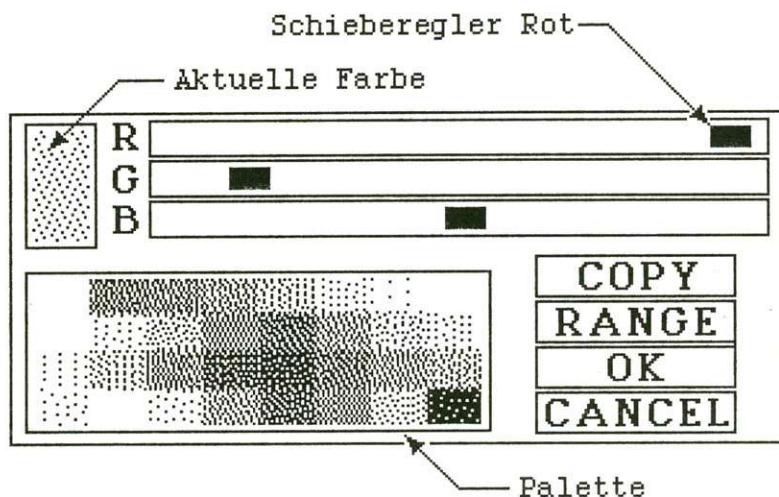


Bild 6 - 4: Der *Change Palette-Requester*

Einen etwas merkwürdigen Effekt, den Sie bei einer echten Leinwand auch nicht sehen werden, bringt der **Change Palette**-Befehl ebenfalls zu Tage. Ändert man nämlich eine Farbe in der Palette, so ändern sich gleichzeitig auch alle Flächen des Bildes, die mit der alten Farbe, die vorher an dieser Stelle der Palette war, gefüllt waren. Jeder Fleck auf dem Bildschirm merkt sich praktisch, aus welchem Farbtopf die Farbe kam, die er besitzt. Ändert sich der Inhalt des Farbtopfes, ändert er sich (aus Solidarität?) mit. Die Gründe dafür können Sie im Kapitel über die Grundlagen der Amiga-Grafik und die Grafik-Chips nachlesen. Wir wollen uns hier nicht im Detail damit beschäftigen.

Eine Konsequenz dieser "nachträglichen Änderungsmöglichkeit" ist es, daß man mit der Palette auch die Farben ändern kann, die für die Darstellung der

Titel-Leiste und auch des Mauszeigers verwendet werden.¹ Bei Änderungen in der Palette kann es durchaus passieren, daß man den Mauszeiger kaum noch sieht, oder die Menüs unleserlich werden, weil es den Texten an Kontrast mangelt. Achten Sie bei Paletten-Änderungen auf diese speziellen Farben!

Eine weitere Konsequenz ist die Möglichkeit der *Farbpaletten-Animation*, die in einem der folgenden Abschnitte beschrieben wird und die selbst einem einfachen Programm wie Graphicraft die Möglichkeit gibt, "bewegte" Bildsequenzen zu erstellen.

Eine besonders wichtige Farbe ist noch die erste, rechts oben in der Palette. Dies ist die sogenannte *Hintergrundfarbe*. Die Hintergrundfarbe erscheint z.B. auf den Flächen am Bildschirm, die Sie nicht bemalen können (über, unter, links und rechts neben dem bemalbaren Rechteck). Sie wird aber auch für alle Operationen verwendet, die Flächen löschen. *Erase* und *Erase Screen* sind eigentlich Füll-Operationen, die die Hintergrundfarbe verwenden.

6.1.2.5 Das *Shape*-Menü

Im *Shape*-Menu stehen dem Maler vier "Werkzeuge", die in bildlicher Form angedeutet werden (s.o), sowie *Text* zur Verfügung. Diese Werkzeuge helfen bei der Erstellung regelmäßiger Formen in einem Bild.

Die diagonale Linie ersetzt z.B. ein Lineal. Nachdem man sie ausgewählt hat, klickt man irgendwo ins Bild und zieht dann die Maus zu einem anderen Punkt. Dort klickt man noch einmal und Graphicraft zieht dann mit dem aktuellen Pinsel eine gerade Line vom ersten zum zweiten Punkt. Der "Kringel" ist der "normale" Pinsel. Wählen sie ihn aus, so malt der Pinsel immer dann, wenn Sie die linke Maustaste drücken und folgt dabei den Bewegungen der Maus.

Das Rechteck malt naheliegenderweise Rechtecke. Das funktioniert ähnlich wie das Ziehen von Linien. Man klickt zunächst in eine Ecke des Rechtecks und klickt in der gegenüberliegenden Ecke noch einmal. Die vier Seiten des Rechtecks werden dann nacheinander mit dem aktuellen Pinsel und natürlich in der aktuellen Farbe von Graphicraft gezogen. Wenn Sie bereits den *Frame*-Befehl ausprobiert haben, werden Sie die Funktionsweise des Rechtecks sofort verstehen.

¹ Die Titel-Leiste und Requester verwenden die Farben 1 bis 4 in der Palette. Der Standard-Mauszeiger verwendet die Farben 17, 18 und 19 – in der Palette von links nach rechts und von oben nach unten gezählt.

Das letzte Symbol des **Shape**-Menüs schließlich, das erste in der zweiten Zeile, dient dem Füllen von Flächen mit einer Farbe. Wählen Sie dieses Symbol aus und klicken dann auf einen bestimmten Punkt des Bildes, so wird ausgehend von diesem Punkt die ganze Fläche, in der er liegt, und die dieselbe Farbe hat, mit der aktuellen Farbe gefüllt.

Haben Sie also z.B. ein rotes Rechteck und möchten es violett haben, ohne die Palette zu ändern, so würden sie dies mit dem Füll-Befehl machen. Wählen Sie dazu einfach die violette Farbe aus dem **Color**-Menü, dann den Füll-Befehl aus dem **Shape**-Menü und klicken dann in das Rechteck.

Der Füll-Befehl ist recht "gefährlich"! Er füllt das Bild nämlich — ausgehend vom angeklickten Punkt — so lange, bis er auf eine andere Farbe trifft. Hat die Fläche, die Sie eigentlich färben wollen, nur ein winziges Loch in ihrer Umrandung, so läuft die Farbe dort aus und färbt Flächen, deren Farbe Sie gar nicht ändern wollten. Bei einem solchen Malheur leistet, wie üblich, der **Undo**-Befehl (im **Edit**-Menü) gute Dienste.

Der letzte Befehl im **Shape**-Menü lautet **Text**. Mit ihm kann man recht komfortabel auch Texte in ein Bild einbringen. Man könnte dies natürlich auch mit dem Pinsel und zur Not mit der Lupe machen, mit dem **Text**-Befehl ist es aber ungleich leichter. Nachdem Sie ihn aufgerufen haben, erscheint zunächst ein Requester, in dem Sie genauer angeben müssen, wie der Text im Bild erscheinen soll. Sie können hier z.B. verschiedene Farben für die Buchstaben selbst und den umgebenden Raum wählen. Klicken Sie dazu auf den Knopf **Text** und wählen dann eine Farbe aus der angezeigten Palette. Klicken Sie dann auf den Knopf **Space** (= umgebender *Raum*) und klicken Sie dann auf die Farbe, mit der die Buchstaben umgeben werden sollen. (Wählen sie dieselbe Farbe für beides, wird der Text natürlich unleserlich)

Sie können aber auch auswählen, wie der Text auf die Farbe "darunter" wirkt: ob er sie völlig mit der für Buchstaben und umgebenden Raum gewählten Farbe auslöscht oder nicht. Es ist z.B. auch möglich, Texte zu verwenden, die verschiedene Farben auf verschiedenem Hintergrund annehmen, damit sie immer lesbar bleiben, egal welche Farbe schon auf dem Bildschirm stand. Welche Farbe dabei an welcher Stelle erscheint, ist aber schwer abzuschätzen.

Alle gerade gültigen Einstellungen werden Ihnen bei der Arbeit im Text-Requester permanent angezeigt, indem ein Beispiel gezeigt wird, wie Text mit den augenblicklichen Einstellungen auf verschiedenen Hintergründen wirken würde.

Wenn Sie mit allen Einstellungen zufrieden sind, drücken Sie auf den Knopf **OK** und können nun mit einem Mausklick die Stelle wählen, an der der Text erscheinen soll. Tippen Sie dann den gewünschten Text auf der Tastatur und

Sie können am Bildschirm zusehen, wie er Buchstabe für Buchstabe über ihrem Bild erscheint.

Vertippen Sie sich dabei, haben Sie aber keinerlei Korrekturmöglichkeiten! Nicht zur Wahl steht auch leider der Zeichensatz für den Text, von denen der Amiga je einige zu bieten hat, wie schon das *Notepad* zeigt. *Graphicraft* ist eben nur ein sehr einfaches Malprogramm.

6.1.2.6 Das *Brush*-Menü

Im *Brush*-Menü steht eine Auswahl von insgesamt 16 verschiedenen Pinsel-Formen zur Verfügung. Wählen Sie einen davon aus, so wird es der aktuelle Pinsel. Dieser Pinsel wird dann sowohl beim "Freihand-Malen" mit dem Pinsel verwendet, wie auch beim Ziehen von geraden Linien und beim Malen von Rechtecken. Normalerweise enthält das *Brush*-Menü eine Reihe von verschieden großen, runden und rechteckigen Pinseln. Sie können sich aber jederzeit mit dem Befehl *Custom Brush* ihre eigenen Pinsel-Formen erstellen und auch für späteren Gebrauch abspeichern (mit dem Befehl *Save As Brushes* im *Project*-Menü).

Custom Brush zeigt Ihnen einen Requester, in dem sie einen der 16 Pinsel auswählen und dann Punkt für Punkt bearbeiten können. Klicken Sie auf den Knopf *OK*, und ein neuer Pinsel erscheint im *Brush*-Menü. Einer der Pinsel (der erste links oben) ist aber recht hartnäckig. Egal, welche Form Sie ihm geben, er malt immer, als bestünde er nur aus einem Punkt. Diese Hartnäckigkeit hat aber auch den Vorteil, daß ihnen immer ein feiner "Haarpinsel" für Detail-Arbeiten zur Verfügung steht.

Besonders interessant sind Pinsel-Formen, die kleine Bilder sind. Unter dem Namen *footprints* (engl. für *Fußabdrücke*) befindet sich z.B. ein Satz Pinsel auf der *Graphicraft*-Diskette, der aussieht, wie Fuß- bzw. Tatzen- oder Pfoten-Abdrücke von Tieren. Wählen Sie einen solchen Pinsel und klicken Sie einmal kurz ins Bild. Schon hinterläßt der Pinsel Fußabdrücke. Sie können sich so leicht auch eine Reihe von "Bausteinen" als Pinsel definieren, aus den Sie dann größere Bilder zusammensetzten. Pinsel haben dafür nur einen kleinen Nachteil: sie sind immer einfarbig.

6.1.3 Farbpaletten-Animation

Eine der interessantesten Funktionen, die das ansonsten eher simple *Graphicraft* zu bieten hat, ist die sogenannte "Farbpaletten-Animation" (engl. *color register animation*) oder "Farbenrotation". Ihr liegt das Prinzip von

Amiga-Grafik zugrunde, das schon im Abschnitt über das Color-Menü erläutert wurde. Jeder Punkt auf dem Bildschirm behält ja immer die Farbe des Farbtopfes, aus dem er gefüllt wurde. Wenn der Farbtopf seinen Inhalt (also seine Farbe) wechselt, so ändert sich die Farbe aller Punkte, die mit seiner "alten" Farbe bemalt wurden, mit. Nachträglich können auf diese Weise große Flächen auf dem Bildschirm sehr rasch ihre Farben ändern.

Graphicraft bietet nun die Möglichkeit, eine Gruppe von Farben in der Palette automatisch in bestimmtem Rythmus zu ändern. Diese Gruppe tauscht die Farben zyklisch aus. Jeweils eine Farbe wird zum nächsten Farbtopf weitergereicht, und dieser gibt zuvor seine Farbe an den übernächsten weiter. Der letzte Farbtopf der Gruppe schließlich gibt seine Farbe an den ersten der Gruppe weiter.

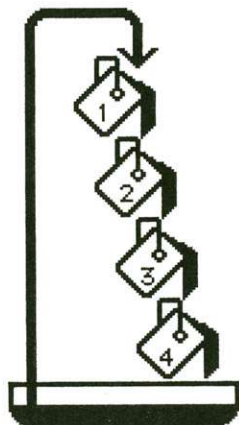


Bild 6 - 5: Das Prinzip der Farbpaletten-Animation

Diesen automatischen Farbwechsel kann man für die Illusion von einfachen Bewegungen einsetzen. Enthalten die Farbtöpfe, die ihre Farben austauschen, z.B. verschiedene Farbtöne einer Farbe, so wirkt der Farbwechsel unter Umständen wie eine Bewegung der mit den dunklen Farben bemalten Flächen. Mit ein wenig Geschick können Sie auf diese Weise die tollsten Effekte erzielen. Ein berühmtes Beispiel hierfür ist das Bild eines Wasserfalls, bei dem das Wasser in verschiedenen Blautönen gemalt wurde. Läßt man diese Blautöne nun "rotieren", so beginnt der Wasserfall in dem vorher stehendem Bild auf einmal zu fließen.

In Graphicraft startet man die Farbpaletten-Animation mit dem Befehl **Cycle Colors** (engl. für *Farben rotieren*) im Menü **Special**. Die Animation kann wieder gestoppt werden, indem der Befehl **Cycle Colors Off** (engl. für *Farbrotation aus*) aufgerufen wird. Nach **Cycle Colors** erscheint ein

Requester, in dem man angeben kann, welche Farben in die "rotierende" Gruppe aufgenommen werden sollen. Dazu drückt man den Knopf **Range** (engl. für *Bereich*) und klickt dann in der gezeigten Palette auf die erste und danach auf die letzte Farbe einer Gruppe. Diese Farben und alle dazwischenliegenden² nehmen dann an der Rotation teil. (Normalerweise ergeben nur recht ähnliche Farbtöne den gewünschten Bewegungseffekt.) Mit einem Regler unter der Palette kann man das gewünschte Tempo einstellen. (Zu langsame Rotation wirkt meist etwas sprunghaft.) Drückt man auf den Knopf **Back**, so läuft die Rotation in umgekehrter Richtung ab. Der Knopf **OK** startet die Farbpaletten-Animation schließlich.³

Diesen automatischen Farbwechsel kann man für die Illusion von einfachen Bewegungen einsetzen. Enthalten die Farbtöpfe, die ihre Farben austauschen, z.B. verschiedene Farbtöne einer Farbe, so wirkt der Farbwechsel unter Umständen wie eine Bewegung der mit den dunklen Farben bemalten Flächen. Mit ein wenig Geschick können Sie auf diese Weise die tollsten Effekte erzielen. Ein berühmtes Beispiel hierfür ist das Bild eines Wasserfalls, bei dem das Wasser in verschiedenen Blautönen gemalt wurde. Läßt man diese Blautöne nun "rotieren", so beginnt der Wasserfall in dem vorher stehendem Bild auf einmal zu fließen.

Eine weitere interessante Möglichkeit ähnlicher Art ist es, die Farbe des Pinsels wirklich kontinuierlich ändern zu lassen. Besonders zusammen mit Farbpaletten-Animation ergibt dies interessante Effekte. Ähnlich wie die Farbtöpfe bei der Farbpaletten-Animation ihre Farben austauschen, wird beim zyklischen Pinselfarbentausch andauernd und während des Malens der Farbtopf gewechselt, aus dem der Pinsel seine Farbe bekommt.

Diesen Effekt schaltet man mit dem Befehl **Cycle Draw** (engl. für *zyklisches Zeichnen*) ein und kann ihn mit **Cycle Draw off** wieder abschalten. Der Requester, der nach **Cycle Draw** erscheint, sieht genauso aus und funktioniert auch genau so, wie der bei **Cycle Colors**. Wieder müssen bzw. können Sie eine Gruppe von Farben und die Wechselgeschwindigkeit auswählen.

² Bei einer Zählung von links nach rechts und von oben nach unten in der Palette.

³ Wenn Sie Interesse an Ihrer geistigen Gesundheit haben, sollten Sie die ersten vier Farben aus jeder rotierenden Gruppe heraushalten. Besonders dann, wenn die Rotationsgeschwindigkeit hoch ist. Diese vier Farben werden nämlich für den Bildschirmhintergrund, die Menüs, etc. verwendet. Wenn der gesamte Bildschirm flackert, sind Kopfschmerzen geradezu vorprogrammiert.

Um einen Eindruck von den Effekten zu bekommen, die mit **Cycle Draw** und **Cycle Colors** möglich sind, können Sie das folgende kleine Experiment probieren.

Starten Sie Graphicraft von der Workbench aus und wählen Sie dann einen großen runden Pinsel aus. Rufen Sie nun **Cycle Draw** auf, klicken Sie auf **Range** und dann zuerst auf die fünfte Farbe in der ersten Zeile der Palette (Mittelrot) und zuletzt auf die achte in der ersten Zeile (Gelb). Den Regler mit der Bezeichnung **Cycle Speed** stellen Sie bitte kurz vor den rechten Anschlag und drücken dann **OK**.

Nun krakeln Sie bei gedrückter linker Maustaste nach Herzenslust ein wenig auf dem Bildschirm herum und betrachten den schönen bunten Wurm, der dabei entsteht. Die Länge seiner farbigen Segmente richtet sich nach dem Tempo, mit dem Sie den Pinsel bewegen. Experimentieren Sie ein wenig damit! Es können auch ruhig mehrere einzelne Würmer werden. Wenn Ihnen die Ergebnisse Ihrer Malversuche nicht gefallen, löschen Sie einfach alles mit **Erase Screen** und fangen noch einmal von vorne an. Nun rufen Sie bitte den Befehl **Cycle Colors** auf und klicken im auftauchenden Requester sofort auf **OK** und ... staunen. Nett, nicht wahr?

Damit soll unsere kurze Betrachtung des ersten Malprogramms für den Amiga aber auch abgeschlossen werden.

6.2 Deluxe Paint

Graphicraft erlaubt es Ihnen, mit den Mal- und Zeichen-Fähigkeiten des Amiga schon etwas "warm zu werden". Sie werden aber bei intensiver Beschäftigung mit dem Programm wahrscheinlich sehr schnell an seine Grenzen stoßen. Ein Programm, das ihnen erlaubt, die bis dahin schon erstellten Zeichnungen weiterzuverwenden und zusätzlich aber neue Zeichnungen mit wesentlich mehr Komfort und wesentlich mehr Möglichkeiten zu erstellen, ist **Deluxe Paint**..

Deluxe Paint ist eines der ersten Amiga-Produkte von *Electronic Arts*, einem amerikanischen Softwarehaus, das für seine sehr gute Software im Spiel- und Heim-Bereich und die unkonventionelle Art und Weise, in der es diese vertreibt, bekannt ist. Electronic Arts versucht, Software ungefähr wie Schallplatten zu produzieren und zu verkaufen. Die Verpackung ist ähnlich und im Gegensatz zu anderen Firmen werden die Leistungen des Programm-Autors stark betont. Auf der Verpackung von Deluxe Paint wird der Autor

dieses Programms, Daniel Silva, z.B. ausführlich vorgestellt. Deluxe Paint ist ein sehr leistungsfähiges Malprogramm, das auch keinen Vergleich mit anderen Programmen auf anderen Rechnern zu scheuen braucht. Zusammen mit der extrem leistungsfähigen Grafik-Hardware des Amiga gibt es eine geradezu explosive Kombination.

Ohne zu übertreiben, kann man sagen, daß Deluxe Paint eines der besten Malprogramme ist, das für nichtprofessionelle Systeme zu haben ist. Für professionelle Künstler und viel teurere Computer gibt es einige Programme, die mehr leisten (z.B. auch mehr Farben ermöglichen). Die meisten dieser Programme sind aber nicht nur viel teurer und benötigen wesentlich größere Computer sondern sind auch viel schwieriger zu bedienen.

Deluxe Paint bietet so vielfältige Möglichkeiten, daß Sie innerhalb dieses Abschnitts unmöglich alle erläutern werden können. Wir können nur einen ersten Eindruck des Programms vermitteln. Viele der raffinierteren Einsatzmöglichkeiten und Tricks lassen sich sowieso am besten durch Ausprobieren verstehen.

Die Idee hinter Deluxe Paint (oder **DPaint**, wie wir im folgenden immer sagen wollen) ist im wesentlichen dieselbe wie bei Graphicraft. Sie bekommen eine zunächst blanke Leinwand, auf der Sie mit einem Pinsel und verschiedenen anderen Werkzeugen Farbe auftragen können. Das kleinste Detail, das Sie mit diesen Werkzeugen bearbeiten können, ist wieder ein Bildschirmpunkt — wobei Sie bei DPaint aber die Auswahl zwischen 3 verschiedenen Stufen von Detailliertheit haben. Der Bildschirm wird entweder in 320 * 200 Punkte, in 640 * 200 Punkte oder in 640 * 400 Punkte aufgeteilt; Sie haben die Wahl.

Etwas anders als bei Graphicraft ist die Bedienung des Programms. Die meisten Operationen des Programms werden über eine Kontroll-Leiste am rechten Bildschirmrand gesteuert. Nur selten benötigte Funktionen befinden sich in den Menüs und selbst diese können oft noch zusätzlich durch einen Tastendruck aufgerufen werden. Auch ist der Aufbau der Menüs wesentlich anders.

6.2.1 Das Starten von Deluxe Paint

Das Starten von DPaint ist etwas gewöhnungsbedürftig, weil anders als bisher von anderen Programmen bekannt. DPaint kann in der vorliegenden Version nicht von der Workbench aus gestartet werden. Der Amiga muß mit der DPaint-Diskette selbst gestartet werden. Nachdem der Amiga die Kickstart-Diskette verdaut hat, schieben sie dazu statt der normalen

Workbench-Diskette die Original-DPaint-Diskette ein. Es erscheint nun nach kurzer Zeit ein Fenster mit dem Namen **AmigaDOS**, in dem ein paar Zeilen Copyright-Informationen ausgegeben werden und danach der folgende Text in einer einzelnen Zeile.⁴

1>

Tippen Sie hinter dem Größerzeichen nun das Wort **dpaint** und dann die <RETURN>-Taste.

1>**dpaint**

DPaint startet nun und begrüßt Sie zunächst mit einem phantastischen, vielfarbigen Bild eines Farbeimers. Sie müssen einige Sekunden warten, bis das Programm eingabebereit ist und können dann mit dem Malen beginnen. Ihnen steht nun eine Leinwand mit 320 * 200 Punkten zur Verfügung, auf der Sie mit 32 Farben arbeiten können.

Wenn Sie statt einfach nur **dpaint** den Text **dpaint med** (für *dpaint medium resolution*) tippen, so erhalten Sie eine Leinwand von 640 * 200 Punkten, allerdings nur noch mit 16 Farben. Tippen Sie **dpaint hi** (für *dpaint high resolution*), stehen ihnen sogar 640 * 400 Punkte in 16₈ verschiedenen Farben zur Verfügung.⁵

6.2.2 Grundzüge von Deluxe Paint

DPaint besitzt neben der Titel-Zeile am oberen Rand des Bildschirms noch eine Kontroll-Leiste am rechten Rand. Diese besteht aus einer Reihe von Symbolen, die mit der Maus angeklickt werden müssen, um etwas zu bewirken. In dieser Kontroll-Leiste wählt man z.B. die aktuelle Pinsel-Form, Pinsel-Farbe usw. aus. Die folgende Abbildung zeigt das Aussehen dieser

⁴ Das Fenster ist ein sogenanntes CLI-Fenster. Wenn Sie mehr über das CLI erfahren wollen, lesen Sie bitte die entsprechenden Kapitel im zweiten Buchteil (ab Kapitel 7) nach.

⁵ Tippen Sie hinter **med** oder **hi** noch eine Zahl, können Sie auch die Zahl der Farben, die Ihnen in der Palette zur Verfügung stehen, noch beeinflussen. Geben Sie keine Zahl an, nimmt DPaint automatisch die oben angegebenen Maximalwerte. Die hinter **hi** oder **med** anzugebene Zahl ist aber nicht die Anzahl von Farben, sondern die Anzahl der *Bit-Ebenen* oder *Farb-Ebenen*, die **DPaint** für die Speicherung des Bildes verwendet. Mehr zu Bit-Ebenen erfahren Sie bei Interesse im vierten Teil dieses Buches bei der Beschreibung der Grundlagen der Amiga-Grafik.

Leiste. Die Erklärungen links und rechts daneben sind nur Stichwörter. Mehr darüber erfahren Sie in den folgenden Abschnitten.

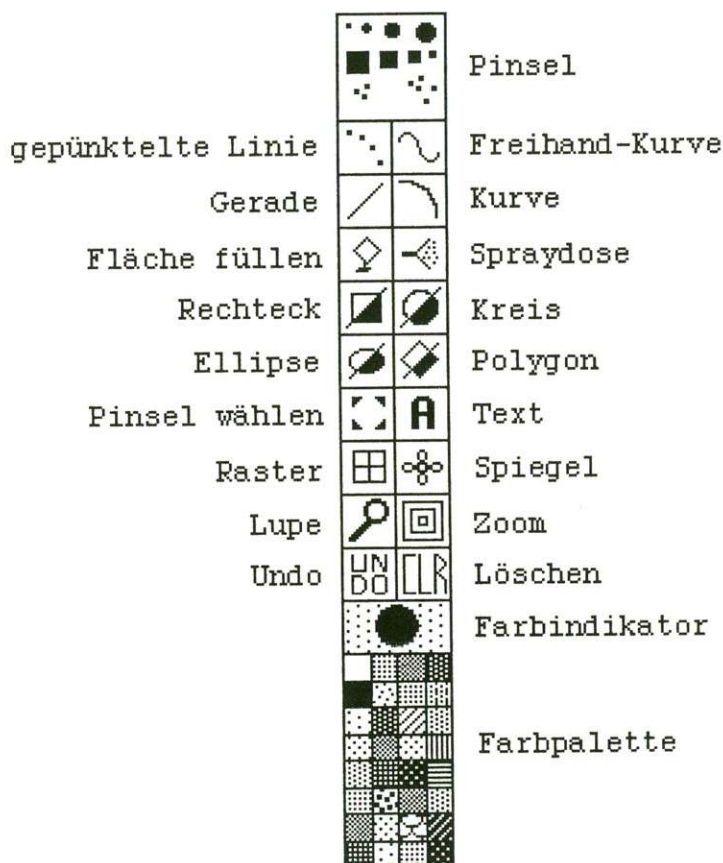


Bild 6 - 6: Die Kontroll-Leiste von Deluxe Paint

Jedes der kleinen (oder größeren) Quadrate in der Kontroll-Leiste stellt ein Hilfsmittel von DPaint dar, mit dem Sie Ihre Bilder auf einfachere Art und Weise erstellen können. Klicken Sie in dieses Quadrat und das Hilfsmittel wird wirksam. Klicken Sie z.B. auf die diagonale Linie, so können Sie mit dem Pinself gerade Linien ziehen, ganz ähnlich, wie es der entsprechende Befehl im *Shape*-Menü von Graphicraft bewirkt. Klicken Sie auf eine der Farben am unteren Ende der Kontroll-Leiste wechselt die Farbe des Pinsels sofort.

Die Handhabung der Maus ist innerhalb von DPaint ein wenig gewöhnungsbedürftig. Die Menütaste hat z.B. meist eine andere Wirkung als in anderen Programmen. Bewegen Sie den Mauszeiger z.B. über das eigentliche Bild und drücken den rechten Mausknopf (Menü-Knopf), so erscheinen in der Titel-Leiste nicht die Menüs. Erst wenn sich der Mauszeiger über der Titel-Leiste befindet, und Sie dann den Menü-Knopf drücken, erscheinen die Menü-Titel und Sie können einen Befehl auswählen.

Dies mag umständlich erscheinen, hat aber durchaus seine Gründe. Während sich der Mauszeiger (und damit auch der Pinsel) über dem Bild befindet, können Sie bei Druck auf die linke Maustaste mit der Vordergrundfarbe malen und bei Druck auf die rechte Maustaste mit der Hintergrundfarbe. (Wie Sie die Vorder- und Hintergrundfarbe auswählen, erfahren Sie im nächsten Abschnitt.) Sie haben also ohne weitere umständliche Aktionen immer zwei Farben zum Malen zur Verfügung.

Wenn Sie in eines der Kästchen in der Kontroll-Leiste klicken, so ist die linke Maustaste am allgemeinen die "normale" bzw. häufigste Verwendung des entsprechenden Hilfsmittels, während ein Klick mit der rechten Taste einen besonderen Befehl einleitet. Ein Klick mit rechts auf eine neue Pinselform erlaubt Ihnen z.B., im Anschluß die Pinselgröße bei unveränderter Grundform zu ändern. Bei manchen Hilfsmitteln gibt es allerdings keinen solchen besonderen Befehl. Bei diesen ist es egal, ob Sie mit rechts oder links klicken.

Die beiden wichtigsten Felder der Kontroll-Leiste, sind neben der Palette **CLR** und **UNDO**. Klicken Sie auf **CLR** (egal mit welcher Taste), so wird das gesamte im Moment sichtbare Bild gelöscht (mit der Hintergrundfarbe). **UNDO** macht immer die letzte Ihrer Aktionen rückgängig. Im Gegensatz zum **Undo**-Befehl in Graphicraft macht **UNDO** innerhalb von DPaint wirklich jede Aktion rückgängig und ist deshalb ein sehr mächtiges Hilfsmittel bei Experimenten mit dem Programm — aber auch bei der späteren Arbeit. Sie finden diese beiden Befehle direkt über der Palette in der Kontroll-Leiste.

6.2.3 Farben

Wie man eine andere Farbe für den Pinsel auswählt, wurde ja bereits erwähnt. Ein Klick mit der linken Maustaste in eine der Farben in der Palette am unteren Ende der Kontroll-Leiste ändert die Vordergrundfarbe. Ein Klick mit der rechten Maustaste ändert die Hintergrundfarbe. Der Farbindikator über der Palette (vgl. Bild 6-6) zeigt stets an, welche Vorder- und

Hintergrundfarbe gerade gültig ist. Der Kreis in der Mitte enthält die Vordergrundfarbe, das Rechteck darum die Hintergrundfarbe.

Genauso wie Vorder- und Hintergrundfarbe ausgewählt werden, kann auch mit ihnen gemalt werden. Befindet sich der Pinsel über der Bildfläche und Sie halten die linke Maustaste fest, malt der Pinsel mit der Vordergrundfarbe. Halten Sie aber die rechte Maustaste fest, malt er mit der Hintergrundfarbe.

Wie Sie es ja bereits aus der Beschreibung von Graphicraft kennen, können Sie natürlich auch in DPaint die Palette ändern. Hierzu klicken Sie einfach mit der **rechten** (Menü-)Taste in den Farbindikator. Es erscheint der **Color Palette**-Requester, der Ihnen die Palette, mehrere Schieberegler und einige Knöpfe zeigt.⁶ In der Palette können Sie die Farbe auswählen, die Sie ändern wollen, indem Sie sie anklicken. Sie können aber auch einfach irgendwo im Bild, das hinter der Palette ja nach wie vor sichtbar ist, auf einen Punkt in einer bestimmten Farbe klicken, wodurch diese Farbe zur Bearbeitung ausgewählt wird. Es gibt zwei Gruppen von je drei Schiebereglern, mit denen Sie die Farbe ändern können.

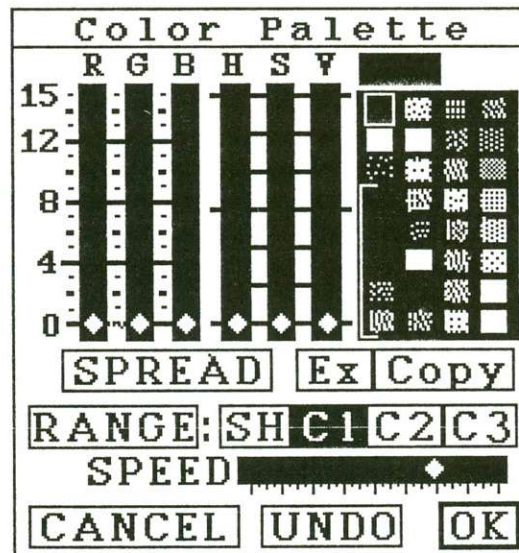


Bild 6 - 7: Ändern der Palette in Deluxe Paint

⁶ Sie können diesen Requester auch mit dem Menü-Befehl **Palette**, Sub-Menü **Color Control** im Menü **Picture** hervorbringen.

Die drei Regler **R**, **G**, und **B** kennen wir ja bereits von Graphicraft oder aus dem Programm *Preferences*. Sie erlauben ein Zusammenmischen der Farbe aus den drei Grundfarben Rot, Grün und Blau. Die Regler **H**, **S** und **V** sind neu. Mit diesen kann die Farbe ebenfalls — aber auf andere Weise — zusammengemischt werden. **H** steht für *Hue* (engl. für *Farbton*). **S** steht für *Saturation* (engl. für *Sattheit* der Farbe). **V** steht für *Value* (engl. für *Wert* oder *Helligkeit* der Farbe). Mit diesen drei Reglern läßt sich der gewünschte Farbton oft einfacher einstellen als mit **R**, **G**, und **B**. Falls Sie bereits Besitzer von Deluxe Paint sind, probieren Sie es einfach einmal aus.

Hierzu stellen Sie zunächst die beiden Regler **S** und **V** an den oberen Anschlag der Skala oder knapp darunter. Bewegen Sie nun den Regler **H** und beobachten Sie, wie die verschiedenen Farben durchlaufen werden. Ganz unten ist Rot. Dann kommen Gelb, Grün, Blau, Violett und schließlich wieder Rot am oberen Ende. Lassen Sie den Regler auf einer dieser Positionen (am besten bei einer satten Farbe, wie Rot oder Tiefblau) stehen, und bewegen Sie nun den Regler **S**. Am oberen Ende der **S**-Skala bekommen Sie einen sehr satten Farbton und unten immer Weiß. Die Positionen dazwischen durchlaufen die verschiedensten Abtönungen der Farbe. Bei Rot werden z.B. alle Rosa-Töne durchlaufen. Wählen Sie nun eine Position nahe dem oberen Ende und probieren Sie den Regler **V** aus. Am oberen Ende bekommen Sie eine reine, helle Farbe. Je mehr Sie aber den Regler tiefer ziehen, desto mehr Schwarz bzw. Grau wird dem Farbton beigemischt, bis die Farbe endlich tiefschwarz ist.

Jedesmal, wenn Sie einen Regler (bzw. den Mausknopf) loslassen, zeigen Ihnen die drei RGB-Regler die entsprechenden Einstellungen im RGB-System. Dies ist ein ganz nettes Experiment, wenn Sie einmal sehen wollen, wie eine bestimmte Farbe im RGB-System zustande kommt. Umgekehrt gilt natürlich das selbe. Wenn Sie einen der RGB-Regler verschieben und loslassen, zeigen Ihnen die HSV-Regler die entsprechenden Positionen an. Es gibt natürlich auch eine Umrechnungsformel zwischen beiden Farb-Systemen, die uns hier aber nicht interessieren soll — wir können es schließlich sofort ausprobieren.⁷

Die wichtigsten Knöpfe im *Color Palette*-Requester sind natürlich **OK** und **Cancel**. Ihre Wirkung dürfte aus anderen Programmen bekannt sein. Der Knopf **Undo** macht die jeweils letzte Änderung an der Palette unwirksam. Mit **Copy** kann man die aktuelle Farbe in einen anderen Farbtopf kippen und mit **Ex** (*exchange*, engl. für *austauschen*) die aktuelle Farbe mit einer anderen

⁷ Bei Interesse kann man sie aber in jedem besseren Lexikon unter *Farbtheorie* oder *Farbfernsehen* nachschlagen.

austauschen. Nach dem Klick auf **Copy** oder **Ex** muß man mit der Maus jeweils auf die zweite beteiligte Farbe klicken.

Der Knopf **Range** kann eine ganze Folge von sanft abgestuften Farbtönen erzeugen. Hierzu klickt man zunächst auf eine Farbe, dann auf Range und dann auf eine zweite Farbe. DPaint erzeugt dann automatisch alle Farbtöne, die entstehen, wenn man die beiden Farben in verschiedenen Verhältnissen miteinander mischt und füllt die Farbtöpfe, die zwischen den beiden angeklickten liegen, mit diesen Mischfarben. Je mehr Farbtöpfe dazwischen liegen, desto feiner ist die Abstufung. Liegt nur ein Farbtopf zwischen den beiden angeklickten, so befindet sich darin nach dieser Operation die Farbe, die entstehen würde, wenn man die beiden Farbtöpfe zusammengekippt hätte. Besonders schön wirkt **Range** zusammen mit der Farbpaletten-Animation, zu der natürlich auch DPaint fähig ist. Die restlichen Kontrollen im **Color Palette**-Requester haben auch alle mit Farbpaletten-Animation bzw. ähnlichen Effekten zu tun und werden erst im entsprechenden Abschnitt besprochen.

6.2.3 Pinsel

Pinsel sind in DPaint — im Gegensatz zu Graphicraft — das wichtigste Konzept überhaupt. Sie sind unglaublich flexibel und können für Dinge eingesetzt werden, bei denen ein echter Pinsel oder auch ein Farbstift oder eine Spraydose hoffnungslos überfordert wären. Die Bedeutung der Pinsel erkennt man aber zunächst kaum. Zu Anfang sehen Pinsel und ihre Bedienung auch aus wie im simpleren Graphicraft.

6.2.3.1 Vordefinierte Pinselformen

Die gewünschte Pinselform kann oben in der Kontroll-Leiste ausgewählt werden. Klicken Sie dazu einfach einen der zehn Pinsel an. Falls Ihnen die Pinselgröße nicht paßt, aber die Form gefällt, so klicken Sie einfach den entsprechenden Pinsel mit der rechten Maustaste an. Lassen die Taste wieder los und bewegen dann den Mauszeiger über das Bild. Unter dem Zeiger erscheint nun der Text **SIZE** (engl. für *Größe*). Drücken Sie nun eine Maustaste und bewegen den Zeiger so lange, bis der Pinsel die gewünschte Größe hat. Die quadratischen und kreisförmigen Pinsel können nahezu jede beliebige Größe annehmen. Die wie Pünktchenwolken aussehenden Pinsel können insgesamt nur sechs verschiedene Größen annehmen.

Auf diese Art und Weise kann man schon einige verschiedene Pinselformen erreichen. Es gibt jedoch in DPaint keine Möglichkeit, Pinsel zu editieren, wie in Graphicraft. Man muß zunächst mit den drei Grundformen in verschiedenen Größen auskommen. DPaint kennt dafür aber eine viel mächtigere Methode, neue Pinsel zu bilden: Ein beliebiger (!) Bereich eines Bildes kann zum Pinsel "ernannt" werden und danach genauso verwendet werden wie z.B. ein kleiner quadratischer Pinsel. Dies bietet ungeheure Möglichkeiten, wenngleich diese nicht sofort zu erkennen sind. Wir wollen uns im folgenden deshalb noch etwas damit beschäftigen.

6.2.3.2 Selbstdefinierte Pinselformen

Wollen Sie einen ersten Eindruck von den Möglichkeiten der selbstdefinierten Pinsel in DPaint bekommen wollen, wählen Sie einfach zunächst den großen runden Pinsel aus der Kontroll-Leiste aus. (Starten Sie DPaint vorher, damit Sie mit einer neuen leeren Leinwand anfangen.) Wählen Sie dann mit einem Klick in die Palette die Farbe "Knallrot" aus und setzen Sie einen schönen roten Klecks auf die Leinwand. Wählen Sie dann die grüne Farbe und setzen Sie einen grünen Klecks daneben. Als letztes setzten Sie in die Mitte unter den roten und grünen Klecks noch einen blauen. Jetzt sollen diese drei Kleckse zum Pinsel ernannt werden. Dazu klicken Sie in das Feld der Kontroll-Leiste, neben dem in Bild 6-6 *Pinsel wählen* steht. Wenn Sie den Mauszeiger jetzt wieder über das Bild bewegen, erscheint ein Fadenkreuz, mit dem Sie ein Rechteck markieren können, aus dem der Pinsel genommen werden soll. Gehen Sie mit dem Fadenkreuz an einen Punkt links oberhalb des roten Kleckses, drücken Sie die linke Maustaste und bewegen dann die Maus bei gedrückter Taste, bis das so markierte Rechteck die drei farbigen Kleckse umschließt. Wenn Sie nun die Maustaste loslassen, erscheint ein Pinsel unter dem Mauszeiger, der aus genau diesen drei farbigen Klecksen besteht.

Mit diesem können Sie jetzt malen. Gehen Sie damit z.B. zu einer "freien" Stelle der Leinwand und klicken kurz mit der linken Maustaste. Eine Kopie des Pinsels (drei Kleckse) bleibt daraufhin zurück. Halten Sie die Maustaste fest und bewegen Sie die Maus ein wenig; so erscheint eine ganze Folge dieser Kleckse auf dem Bildschirm. Sie haben jetzt bestimmt schon mindestens einen der entscheidenden Vorteile der DPaint-Pinsel gegenüber den Graphicraft-Pinseln kennengelernt: DPaint-Pinsel können mehrfarbig sein!

Mit diesen selbstdefinierten, mehrfarbigen Pinseln können Sie genauso arbeiten wie mit den einfacheren Pinseln aus der Kontroll-Leiste auch. Sie können nicht nur Kleckse in die Landschaft setzen, sondern damit auch Linien, Rechtecke, Kreise oder Kurven ziehen usw. Mehr dazu im folgenden Abschnitt.

Das Auswählen eines Pinsels in DPaint ersetzt vollständig das **Edit**-Menü aus Graphicraft. Wenn Sie einen Bildteil duplizieren wollen, so machen Sie ihn einfach zum Pinsel und klicken dann an die Stelle, an der das Duplicat erscheinen soll. Der Bereich, der so dupliziert wird, kann sogar unregelmäßig geformt sein. Vielleicht ist es Ihnen schon bei dem oben beschriebenen einfachen Pinsel-Experiment aufgefallen: Nur die drei Farbkleckse wirken nachher als Pinsel. Die umgebende schwarze Fläche, die ja auch in dem zunächst ausgewählten Rechteck lag, wird nicht "in den Pinsel" aufgenommen. Selbstdefinierte Pinsel sind also nicht immer rechteckig, wie man vielleicht annehmen sollte. Als Regel dafür gilt, daß die Farbe, die gerade als Hintergrund-Farbe im Farbindikator angezeigt wird, nicht in einen selbstdefinierten Pinsel aufgenommen wird. Auf Wunsch kann dieses Verhalten aber auch mit einem Menü-Befehl abgestellt werden, so daß das komplette ausgewählte Rechteck zum aktuellen Pinsel wird.

Auch das Löschen von Bildteilen ist mit selbstdefinierten Pinseln möglich. Es ist nämlich von entscheidender Bedeutung, ob die Auswahl des Pinsels mit der rechten oder der linken Maustaste erfolgte. Verwendet man die linke Maustaste, so wird eine Kopie des Rechtecks zum neuen Pinsel (wie oben beschrieben). Nimmt man aber die rechte Maustaste zur Selektion des Rechtecks, so wird ebenfalls das so ausgewählte Rechteck zum Pinsel, wird danach aber aus dem Bild entfernt (gelöscht). Dazu wird es mit der aktuellen Hintergrundfarbe gefüllt!

Wenn Sie weiter mit selbstdefinierten Pinseln experimentieren wollen, so sollten Sie komplexere Bilder dazu nehmen. Auf Ihrer DPaint-Diskette befinden sich eine ganze Reihe von Beispielen, die Sie mit dem Befehl **Load** im Menü **Picture** gezeigt bekommen können. Falls Sie mit der Bedienung dieses Befehls nicht klarkommen, blättern Sie einfach zum Abschnitt *Menü-Befehle von Deluxe Paint* vor. Dort wird er näher erklärt.

Auch das Abspeichern und Wiedereinlesen einzelner Pinsel ist möglich. Hierzu dienen die Befehle **Load** und **Save** im Menü **Brush**, die ebenfalls weiter unten noch näher erläutert werden. Pinsel eignen sich deshalb sehr gut dazu, Bilder aus Bausteinen aufzubauen, die man zuvor separat gemalt und dann als Pinsel abgespeichert hat.

6.2.4 Linien und Kurven

Die vier Felder in der Kontroll-Leiste unter den Standard-Pinseln von DPaint entsprechen den verschiedenen Werkzeugen, mit denen Sie gerade und andere Linien malen können. Das Symbol, das aussieht wie eine Welle,

entspricht dem "normalen" Pinselverhalten. Wenn Sie dieses Symbol ausgewählt haben, malt der Pinsel immer dann, wenn Sie die Maustaste gedrückt halten und wird von der Leinwand "abgehoben", wenn Sie sie loslassen. Die gepunktelte Linie links daneben bewirkt auf den ersten Blick kaum ein anderes Verhalten. Unterschiede bemerkt man erst, wenn der Pinsel sehr schnell bewegt wird. Das rechte Symbol sorgt dann dafür, daß automatisch immer eine durchgezogene Linie entsteht. Beim linken Symbol ergibt sich eine gepunktelte Linie, wobei die Abstände umso größer werden, je schneller die Pinsel-Bewegung wird. Experimentieren Sie ein wenig damit, um ein Gefühl für die unterschiedliche Wirkungsweise dieser beiden Instrumente zu bekommen. Besonders bei der Verwendung mehrfarbiger Pinsel (s.o.) können große Unterschiede auftreten, je nachdem, ob man die durchgezogene oder die gepunktelte Kurve wählt.

Unter der gepunktelten Kurve befindet sich die gerade Linie. Sie wirkt ähnlich wie das entsprechenden Instrument in Graphicraft. Klicken Sie mit der Maus in einen Punkt Ihres Bildes, und halten Sie die Taste dann fest. Bewegen Sie die Maus nun zu einer zweiten Position, und lassen Sie die Taste los. DPaint zieht dann eine gerade Linie mit dem aktuellen Pinsel zwischen den beiden Positionen. Ist der Pinsel mehrfarbig, ist es die Linie auch.

Die letzte Linienform, die in DPaint zur Verfügung steht, ist eine universelle Kurve. Sie ist nicht immer ein Viertelkreis, wie es das entsprechende Symbol in der Kontroll-Leiste andeutet, sondern erzeugt im allgemeinen Hyperbeln oder Parabeln. Man klickt dazu zunächst auf einen Punkt und zieht zu einem zweiten, wo man die Maus-Taste losläßt. Dies ergibt erst einmal eine gerade Linie. Bewegt man nun die Maus, so krümmt sich die Linie in nahezu beliebiger Weise. (Dies ist schwer zu beschreiben, man muß es in Aktion sehen.) Der nächste Mausklick zeichnet dann die Linie endgültig ins Bild. Hierzu wird wieder der aktuelle Pinsel verwendet.

Bei allen Linienformen gilt wie immer eine Grundregel von DPaint: Arbeitet man mit der linken Maustaste, so wird die aktuelle Vordergrundfarbe (bzw. die Farben des selbstdefinierten Pinsels) verwendet. Nimmt man die rechte Maustaste, so entsteht die Linie in der Hintergrundfarbe — auch bei einem selbstdefinierten Pinsel.

6.2.5 Flächen

Neben Linien und Kurven können Sie mit den Hilfsmitteln aus der Kontroll-Leiste auch eine Vielzahl von regelmäßig und unregelmäßig geformten Flächen in einem Bild erzeugen.

Die universellsten Hilfsmittel für diesen Zweck sind zweifellos der Farbeimer (das Hilfsmittel, neben dem in der Abbildung der Kontroll-Leiste *Flächen füllen* steht) und die Spraydose. Der Farbeimer verhält sich genauso, wie das entsprechende Hilfsmittel von Graphicraft. Wenn Sie in irgendeine Fläche klicken, so wird die gesamte Fläche mit der aktuellen Vorder- oder Hintergrundfarbe gefüllt, je nachdem ob Sie mit der linken oder der rechten Taste klicken.⁸

Einen ähnlichen Farbeimer hatte Graphicraft ja auch schon zu bieten. Ein weiteres interessantes Hilfsmittel ist in DPaint aber die Spraydose. Sie wirkt am besten zusammen mit einer der gepunkteten Pinselformen. Wenn Sie ins Bild klicken, nachdem Sie die Spraydose ausgewählt haben, und die Maus etwas bewegen, werden Sie feststellen, daß Ihr Pinsel wirklich zur Spraydose geworden ist. Die Farbe wird in einem gewissen "Umfeld" des Pinsels auf das Bild "gesprüht", und je länger Sie den Pinsel dabei über einer bestimmten Stelle festhalten, desto gleichmäßiger wird diese Fläche gefärbt. Besonders Schattierungen und Farb-Übergänge lassen sich auf diese Weise sehr einfach erzielen.

Unter dem Farbeimer und der Sprühdose befinden sich vier Hilfsmittel, deren Quadrate diagonal geteilt sind. Sie dienen zum Zeichnen von Rechtecken, Kreisen, Ellipsen und Polygonen, also geschlossenen Flächen. Klicken Sie nun in den oberen linken Teil des Quadrats, mit dem das jeweilige Hilfsmittel ausgesucht wird, so wird die entstehende Fläche mit dem aktuellen Pinsel umrahmt. Klicken Sie jedoch in die rechte untere Hälfte, so wird die Fläche mit der aktuellen Vorder- oder Hintergrundfarbe gefüllt. Solange Sie mit der Maus noch nicht sehr vertraut sind, kann es schon einmal vorkommen, daß Sie aus Versehen die falsche Fläche anklicken. Sie können dies aber jederzeit überprüfen, indem Sie sich nach dem Anklicken noch einmal das Hilfsmittel anschauen. Es zeigt dann entweder eine hohle oder gefüllte Form, je nachdem ob jetzt eine gefüllte Fläche oder nur die Umrandung gezeichnet würde.

⁸ Der genaue Punkt, aus dem die Farbe "herausläuft", ist die Stelle wo die beiden Arme des unteren Teils des Mauszeiger zusammentreffen. An dieser Stelle hat der Zeiger eine kleine Ausbuchtung, die eine sehr genaue Positionierung erlaubt.

Nach dieser langen Vorrede sollen Sie nun aber auch erfahren, wie man Flächen zeichnet. (Zunächst müssen Sie dazu natürlich in das entsprechende Hilfsmittel klicken.) Um ein Rechteck zu zeichnen, klicken Sie an die Stelle, an der eine Ecke des Rechtecks sein soll, halten die Maus fest und ziehen den Mauszeiger in die rechte untere Ecke. Sobald Sie die Maustaste loslassen, erscheint das Rechteck dann wirklich in ihrem Bild. Vorher wird es aber immer schon angedeutet, während Sie die Maus bewegen. Bei Kreisen klicken Sie zunächst in das Zentrum des Kreises und bewegen dann den Mauszeiger zu einem Punkt auf dem Kreisrand. Auch hier wird der Kreis erst dann endgültig gezeichnet, wenn Sie die Maustaste wieder loslassen.

Um eine Ellipse zu zeichnen, müssen Sie sich zunächst ein Rechteck vorstellen, das dieses Oval gerade noch umschließt. Dann verfahren Sie genauso, als wollten Sie dieses Rechteck zeichnen — zuvor klicken Sie aber auf Ellipse in der Kontroll-Leiste. Achten Sie bitte darauf, daß die Achsen der Ellipsen, die DPaint malen kann, immer waagerecht und senkrecht liegen, es also nicht möglich ist, eine Ellipse zu malen, deren lange Achse schräg liegt.⁹

Ganz anders von der Bedienung her als die anderen Flächen-Hilfsmittel verhalten sich **Polygone** (Vielecke). Nachdem Sie das Polygon (die Raute) in der Kontroll-Leiste angeklickt haben, müssen Sie nacheinander die Eckpunkte des Polygons anklicken. Sie können also nach jedem Punkt die Maustaste wieder loslassen und zum nächsten Punkt gehen. Erst, wenn Sie den ersten Punkt noch einmal anklicken, gilt das Polygon als abgeschlossen und erscheint im Bild. Das genaue Aussehen von gefüllten Polygenen, deren Seiten sich überschneiden, ist dabei übrigens nicht immer so genau vorauszusehen. Falls das Ergebnis einmal nicht Ihren Vorstellungen entsprechend sollte, gibt es immer noch die **UNDO**-Funktion!

6.2.6 Text

Die Möglichkeiten von DPaint, Text in ein Bild einzufügen, sind etwas flexibler als die von Graphicraft. Sie wählen dazu zunächst das große "A" in der Kontroll-Leiste aus und klicken dann an die Stelle, an der Ihr Text erscheinen soll. Nun können sie ihn eintippen, wobei Tippfehler mit <BACKSPACE> korrigiert werden können. Falls Ihnen das Aussehen der Zeichen aber nicht gefällt, können Sie jederzeit einen anderen Zeichensatz wählen. (Die verschiedenen Zeichensätze, die der Amiga bietet, haben wir ja schon beim Werkzeug Notepad kennengelernt.)

⁹ Dies ist aber auf einem Umweg möglich, indem Sie eine Ellipse zum Pinsel machen und diesen Pinsel dann rotieren lassen (s.u.).

Das Menü **Font** von DPaint dient zur Auswahl eines Zeichensatzes für neu eingegebenen Text. Bevor Sie dieses Menü aber das erste Mal benutzen, befinden sich keine Zeichensatznamen darin, sondern nur der eine Menü-Punkt **Load Fonts** (*Zeichensätze laden*). Rufen Sie diesen Punkt einmal auf, und wenn Sie dann nach einer kurzen Wartezeit — während der DPaint nachschaut, welche Zeichensätze auf dieser Diskette zur Verfügung stehen — das Menü erneut herunterklappen, erscheinen die Zeichensatznamen.

Anders als im Notepad können Sie in DPaint den Zeichensatz von Texten aber nie nachträglich ändern. Wenn ein Buchstabe erst einmal im Bild zu sehen ist, kann er zwar eventuell noch mit <BACKSPACE> gelöscht werden, aber nicht mehr seinen Zeichensatz ändern.

6.2.7 Andere Hilfsmittel

Damit sind alle Hilfsmittel, die wirklich etwas im Bild erscheinen lassen, vorgestellt wurden. DPaint bietet aber noch einige Hilfsmittel an, die auf andere Art beim Malen helfen.

Mit dem **Raster** können Sie z.B. erzwingen, daß sich alle Maloperationen an einem unsichtbaren Gitter orientieren, das die Bewegungen des Pinsels einschränkt. Rechtecke können dann z.B. keine beliebige Seitenlänge mehr haben, sondern nur noch ganzzahlige Vielfache der Raster-Schrittweite. Wählen Sie z.B. einen kleinen Pinsel aus, und klicken Sie dann mit der linken Maustaste auf das Raster-Symbol. Der Mauszeiger selbst kann sich dann nach wie vor kontinuierlich bewegen. Der Pinsel aber "hüpft" von einem Punkt des Rasters zum nächsten. Bei freihändigem Malen stört dieses Raster meist erheblich. Bei exakten Zeichnungen jedoch kann es eine große Hilfe darstellen.

Sie können die Raster auch verstellen. Klicken Sie dazu mit der rechten Maustaste in das Raster-Symbol und wenn Sie nun mit dem Mauszeiger über die Bildfläche zurückkehren, wird ein 4*4-großer Ausschnitt aus dem Raster angezeigt. Klicken Sie dann erneut und halten Sie die Maustaste fest, so bleibt die linke obere Ecke des Rasters fest und Sie können die rechte untere mit der Maus verschieben. Lassen Sie die Maustaste wieder los, so bekommt das Raster die in diesem Moment eingestellte Schrittweite. Die horizontalen und vertikalen Schrittweiten brauchen dabei nicht übereinzustimmen. Es kann vertikal z.B. ein sehr grobes Raster geben und horizontal gar keines, indem man die horizontale Schrittweite auf 1 stellt.

Rechts neben dem Raster-Symbol liegt ein Hilfsmittel, welches wie eine **Schneeflocke** aussieht. Damit werden die Muster angedeutet, die man

typischerweise damit malen kann. Klickt man mit der linken Taste auf dieses Symbol, so werden die Spiegelachsen von DPaint aktiv. Alles, was man danach zeichnet, wird dann an sechs Achsen gespiegelt, die sich im Mittelpunkt des Bildschirms kreuzen. Dies ergibt — mit einem kleinen Pinsel — "blumige" Effekte, wenn man damit eine wenig auf dem Bildschirm "herumkrakelt".

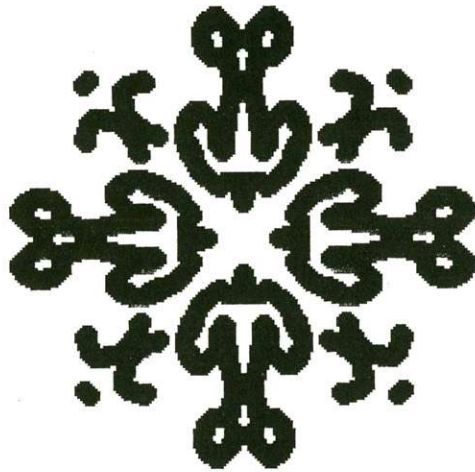


Bild 6 - 8: *Ein typisches gespiegeltes Bild*

Wenn Ihnen sechs Spiegelachsen zu viel sind, können Sie diese Zahl auch verringern — oder vergrößern. Dazu klicken Sie mit der rechten Taste in die Schneeflocke, woraufhin ein Requester erscheint, in dem man die gewünschte Achsenzahl und *Achsenart* angeben kann. Neben den normalen Spiegelachsen kennt DPaint nämlich auch noch die zyklische Spiegelung. Diese ist etwas schwer mit Worten zu beschreiben, und Sie probieren sie am besten einmal mit zwei, drei oder höchstens vier Spiegelachsen aus, um ein Gefühl dafür zu bekommen. Mit zyklischer Spiegelung lassen sich sehr leicht spiralförmige Formen erstellen.

Die beiden letzten der speziellen Hilfsmittel, deren Erklärung noch aussteht, sind die Lupe und der Zoom. Was die **Lupe** tut, ist relativ naheliegend — sie vergrößert einen Ausschnitt aus dem Bild, damit man Details besser bearbeiten kann. Klickt man auf die Lupe und bewegt den Mauszeiger zurück über das Bild, so erscheint ein kleines Rechteck, das den Mausbewegungen folgt. Sobald man nun eine Maustaste niederdrückt, wird der Teil des Bildes, der in diesem Moment innerhalb dieses Rechtecks lag, in vergrößerter Form dargestellt. Hierzu erscheint kein neues Fenster wie bei Graphicraft, sondern

der Bildschirm wird geteilt. Im linken Drittel ist ein Ausschnitt des Bildes in normaler Größe zu sehen, rechts die Vergrößerung eines Teils daraus.

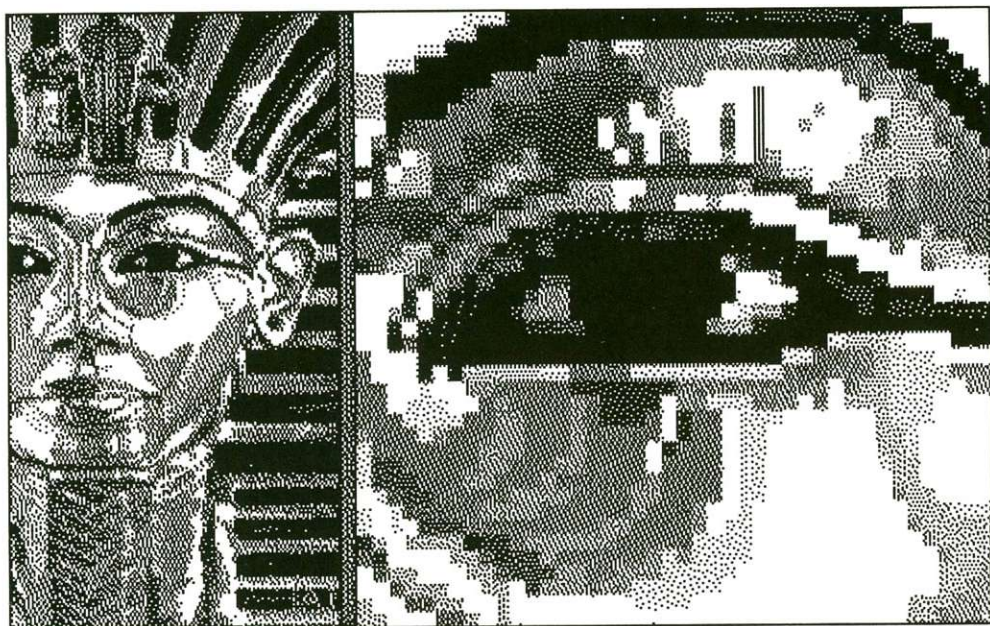


Bild 6 - 9: *Die Lupe in Aktion*

Man kann nun sowohl innerhalb der Vergrößerung, als auch innerhalb des normalen Bildes weiterarbeiten. Alle Hilfsmittel der Kontroll-Leiste funktionieren in der Vergrößerung genauso wie im normalen Bild. Jede Änderung in einer Hälfte des Bildschirms wird auch in der anderen sofort gezeigt. Die Lupe zeigt wirklich nur einen Teil des Bildes in detaillierterer Form; sie schränkt die möglichen Operationen in keiner Weise ein und ist somit ein nahezu ideales Hilfsmittel.¹⁰ Damit nicht jedesmal die Lupe wieder "abgeschaltet" werden muß, wenn der zu vergrößernde Ausschnitt geändert werden soll, kann man diesen mit den Cursor-Tasten bewegen. Viele

¹⁰ Bei der augenblicklichen (ersten) Version von DPaint hat die Lupe etwas Schwierigkeiten mit dem Raster und den Spiegelachsen. Sie sollten diese deshalb am besten abschalten, bevor Sie die Lupe aktivieren. Gerade innerhalb der Vergrößerung sind diese Hilfsmittel aber entbehrlich.

detaillierte Bilder werden mit DPaint deshalb typischerweise gemalt, während die Lupe ständig an ist.

Durch den **Zoom** wird die Lupe noch vielseitiger. Mit diesem kann die "Stärke" der Lupe verstellt werden. Klickt man mit der linken Maustaste in den Zoom, so wird die Vergrößerung stärker. Klickt man mit der rechten, wird sie kleiner.

Schauen Sie sich doch einmal einige der Bilder, die sich bereits auf Ihrer DPaint-Diskette befinden, mit der Lupe an! Sie werden dabei einige interessante Tricks finden, die man auch für eigene Bilder verwenden kann. Besonders die Übergänge zwischen verschiedenen gefärbten Flächen sind oft raffiniert gemacht.

6.2.8 Die Menüs von DPaint

Die Menü-Funktionen sind — von einigen Ausnahmen abgesehen — in DPaint bei weitem nicht so wichtig wie in anderen Programmen. Die meisten Dinge werden über die Kontroll-Leiste abgewickelt. Die Menüs von DPaint dienen hauptsächlich "globalen" Aufgaben, wie dem Abspeichern und Wiedereinlesen von Bildern und Pinseln.

6.2.8.1 Das *Picture*-Menu

Das Menü **Picture** dient z.B. der Bearbeitung des Bildes "als Ganzes". Am einfachsten zu erläutern ist der letzte Befehl darin **Quit** (engl. für *Beenden*). Mit diesem Befehl wird DPaint verlassen. Alle Änderungen am gerade bearbeiteten Bild, die noch nicht gesichert wurden, gehen dadurch verloren.

Der nächste Befehl, **Load...** (engl. für *Laden, Einlesen*) dient zum Bereitstellen eines auf Diskette schon vorhandenen Bildes zu Bearbeitung. Nachdem er ausgewählt wurde, erscheint stets ein Requester, der unter anderem eine Liste der Namen aller Bilder enthält, die sich auf der Diskette in einer bestimmten Schublade befinden. Der Name dieser Schublade ist unterhalb dieser Liste zu lesen. Er steht in einem Text-Gadget neben dem Wort **Drawer** (engl. für *Schublade*). Den Inhalt dieses Gadgets können Sie natürlich auch ändern. Wenn Sie dann **<RETURN>** tippen, werden ihnen die Namen der Bilder in der Schublade angezeigt, deren Namen Sie eingegeben

haben. (Gibt es keine Schublade dieses Namens, so werden natürlich keine Namen gezeigt.)¹¹

Die Liste der Bilder in einer Schublade kann natürlich länger sein als der dafür vorgesehene Platz im Requester. In diesem Fall können Sie die Liste mit dem Rollbalken rechts davon auf und ab bewegen, bis Sie den Namen des Bildes sehen, das Sie suchen. Klicken Sie darauf, und der Name erscheint in dem Text-Gadget, neben dem **File** (engl. für *Datei*) steht.

Es kann auch passieren, daß in der Liste der Bilder Namen auftauchen, hinter denen in Klammern **Dir** steht, die Kurzform von *Directory* (engl. für *Dateiverzeichnis*). Auf diese Weise markiert DPaint die Schubladen in der aktuellen Schublade. Wenn Sie sehen wollen, welche Bilder eine solche Schublade enthält, müssen Sie einfach auf den Namen klicken. Diese Schublade wird dadurch zur aktuellen Schublade.

Sobald Sie die Taste **Load** in diesem Requester betätigen, wird das Bild eingelesen, dessen Name neben **File** steht und das in der Schublade liegt, deren Name neben **Drawer** steht.

Die beiden nächsten Befehle im **Picture**-Menü **Save <Name>** und **Save As...** (engl. für *Sichern, Abspeichern*), dienen, genau wie die entsprechenden Befehle in Graphicraft, zum Abspeichern des gerade bearbeiteten Bildes. Bei **Save As...** können Sie auch noch einen Namen angeben unter dem das Bild abgespeichert werden soll. Der Requester, in dem dies möglich ist, wird genauso bedient wie der bei **Load**.

Mit dem Befehl **Symmetry Center** (engl. für *Symmetrie-Zentrum*) kann der Mittelpunkt festgelegt werden, in dem sich die Spiegelachsen (s.o) schneiden. Dazu wird der Befehl einfach ausgewählt und dann im Bild an dem gewünschten Mittelpunkt einmal geklickt.

Der nächste Befehl **Color Control** (engl. für *Farb-Kontrolle*) zeigt zunächst ein Sub-Menü mit den folgenden Punkten: **Palette** zeigt den Palette-Requester, der weiter oben schon beschrieben wurde. Es gibt allerdings deutlich schnellere Methoden, ihn hervorzubringen (s.o). **Use Brush Palette** und **Restore Palette** beziehen sich auf Konzepte, die wir hier aus Platzgründen nicht erläutern können und **Cycle** schaltet die Farbpaletten-Animation (s.u.) ein oder aus.

Spare (engl. für *Ersatz*) zeigt ebenfalls ein Sub-Menü, dessen Punkte sich alle auf ein Konzept beziehen, das wir hier leider ebenfalls nicht mit der ihm

¹¹ Wie die Namen von Schubladen aussehen, die selbst in Schubladen oder auf anderen Disketten liegen, erfahren Sie im Kapitel *Dateien und Dateiverzeichnisse*.

gebührenden Ausführlichkeit behandeln können. Der erste Punkt im Sub-Menü **Spare**, **Swap** (engl. für *auswechseln*) schaltet den Bildschirm auf das *Ersatzbild* um, ein komplettes zweites Bild, das DPaint gleichzeitig mit dem Hauptbild bearbeiten kann. Die anderen Punkte beziehen sich auf ein Mischen dieses Ersatzbildes mit dem Hauptbild. Das Ersatzbild dient meist nicht wirklich zum Bearbeiten eines zweiten Bildes, sondern eher zum Vorbereiten einzelner Details für das Hauptbild, die man dann als Pinsel definiert und nach einem erneuten Bildwechsel ins Hauptbild einsetzt.

Der Befehl **Print** schließlich tut genau das, was man von ihm erwartet: er gibt das aktuelle Bild auf einem angeschlossenen Drucker aus. Dies ist natürlich nur sinnvoll, wenn dieser Drucker grafikfähig (und am besten sogar farbfähig) ist.

6.2.8.2 Das Brush-Menü

Das **Brush**-Menü von DPaint dient der "globalen" Bearbeitung von selbstdefinierten Pinseln. Diese sind, wie man inzwischen wohl schon gesehen hat, ein sehr wichtiges Konzept des Programms und werden in diesem Menü und dem folgenden Menü deshalb auch ausführlich berücksichtigt. Die beiden ersten Befehle im Brush-Menü, **Load...** und **Save As...** funktionieren genauso wie die entsprechenden Befehle im Picture-Menü und bedürfen wohl keiner weiteren Erläuterung. Als Schublade, in der die Pinsel abgespeichert sind, schlägt DPaint übrigens standardmäßig **brushes** (engl. für *Pinsel*) vor. Sie können aber auch andere Schubladen wählen — z.B. eine Schublade, die Bilder enthält. In diesem Fall wird ein komplettes Bild als Pinsel geladen! Sie können auf diese Weise dann sogar eigentlich drei komplette Bilder mit DPaint bearbeiten. Das Hauptbild, das Ersatzbild und den Pinsel.

Der Befehl **Size** (engl. für *Größe*) erlaubt es, die Größe eines selbstdefinierten Pinsels nachträglich zu verändern. Die Befehle **Half**, **Double**, **Double Horiz** und **Double Vert** halbieren oder verdoppeln die Abmessungen des Pinsels; die beiden letzten Befehle jeweils nur in der Horizontalen oder Vertikalen. Der erste Befehl, **Stretch**, im Sub-Menü **Size** erlaubt eine beliebige (kontinuierliche) Änderung der Pinselgröße. Dies geht prinzipiell genauso wie die Größenänderung eines Standard-Pinsels, die oben schon beschrieben wurde.

Mit dem Menü-Punkt **Flip** (engl. für *Umlegen*, *Spiegeln*), der ebenfalls ein Sub-Menü hervorbringt, kann ein selbstdefinierter Pinsel entweder in der Horizontalen oder in der Vertikalen gespiegelt werden.

Auch **Rotate** zeigt zunächst ein Sub-Menü, mit dessen Punkten der Pinsel entweder um 90 Grad oder um einen beliebigen Winkel (*Any Angle*) gedreht werden kann. Rotationen um einen beliebigen Winkel kosten allerdings einige Rechenzeit, weshalb Sie bei größeren Pinsel etwas Geduld mitbringen müssen.

Der Punkt **Change Color** im **Brush**-Menü soll übergangen werden, weil auch er einige zusätzliche Erläuterungen verlangen würde, für die hier kein Platz ist.

Der letzte Befehl im **Brush**-Menü, **Bend** (engl. für *Verbiegen*, *Verzerren*), ist wieder ein Sub-Menü und erlaubt eine in Worten schwer beschreibbare Verzerrung des Pinsels. Rechteckige Pinsel können mit den beiden Punkten in diesem Sub-Menü z.B. zu hyperbolischen Flächen verzerrt werden — die dabei entstehenden Randbegrenzungen sind den Linien, die man mit dem Kurven-Werkzeug malen kann, sehr ähnlich.

Die folgende Abbildung zeigt das Bild Venus, das sich auf der Original-DPaint-Diskette befindet. Der Kopf wurde in diesem Bild als Pinsel aufgegriffen und gedreht und verzerrt. Dies ist nicht gerade eine sinnvolle Anwendung des Pinsels, illustriert die Möglichkeiten aber recht schön.



Bild 6 - 10: *Ein Bild mit rotiertem und verzerrtem Pinsel*

6.2.8.2 Das *Modes*-Menu

Das *Modes*-Menü von DPaint können wir leider auch wieder nicht komplett beschreiben. Nur einige der Befehle darin lassen sich nämlich ohne komplizierte Erläuterungen und sehr viele (Farb-)Bilder verständlich machen.

Das *Modes*-Menü ist vor allem für Tricks mit selbstdefinierten Pinsel "zuständig". Ein solcher Pinsel kann nämlich in mehreren Modi malen. Der gerade gültige Modus wird auch immer oben in der Menü-Leiste angezeigt. Meist steht dort *Object*, was auch der erste Befehl im *Modes*-Menü ist. *Object* bedeutet, daß der Pinsel sich so verhält, wie bisher immer beschrieben wurde. Nur die Flächen eines selbstdefinierten Pinsels, die beim "Aufnehmen" des Pinsel nicht die Hintergrundfarbe trugen, malen, d.h. erscheinen beim Absenken des Pinsels im Bild. Die Flächen, die die

Hintergrundfarbe trugen, als der Pinsel aufgenommen wurden, bleiben hingegen "transparent".

Der Befehl **Color** aus dem **Modes**-Menü bewirkt, daß auch ein selbstdefinierter Pinsel nur mit der Vordergrundfarbe und nicht mit den "aufgenommenen" Farben malt. Nur seine frei wählbare Form unterscheidet ihn dann noch von einem der vordefinierten Pinsel. Wählt man den Modus **Replace**, so malt der Pinsel wieder mit den aufgenommenen Farben. Jetzt malen aber nicht mehr nur die von der Hintergrundfarbe verschiedenen Flächen des Pinsels. Auch die transparenten Flächen des Pinsels überdecken jetzt darunterliegende Punkte des Bildes.

Die folgenden drei Befehle im Modes-Menü, **Smear**, **Blend** und **Shade** dienen zum Schattieren und Verschmieren von Farben im Bild. Sie können aus Platzmangel hier leider nicht ausführlich erklärt werden. Ein Tip aber, wenn Sie damit experimentieren wollen: sobald einer dieser Modi aktiviert ist, wirkt der Pinsel manchmal nur noch auf Farben in einem bestimmten Bereich der Palette. Diesen Bereich können Sie markieren, nachdem Sie den Palette-Requester aufgerufen und die Taste **SH** betätigt haben.

Cycle im **Modes**-Menü schließlich wirkt genauso, wie **Cycle Colors** in Graphicraft. Während man malt, ändert der Pinsel dann kontinuierlich seine Farbe. Er durchläuft dabei die Farben, die in dem Bereich liegen, der auch für die Farbpaletten-Animation selektiert wurde (s.u.). Die Geschwindigkeit der Farbänderung entspricht ebenfalls der Einstellung des entsprechenden Schiebereglers für die Farbpaletten-Animation. **Cycle** "funktioniert" nur dann, wenn die aktuelle Vordergrundfarbe in dem Bereich der Palette liegt, der auch für die Farbpaletten-Animation selektiert wurde.

6.2.8.4 Das **Font**-Menu

Das Font-Menü wurde weiter oben ja bereits beschrieben. Es dient der Auswahl eines Zeichensatzes (**Fonts**) für das Einsetzen von Texten in ein Bild. Um diese Zeichensätze sehen zu können, muß man erst einmal den Befehls **Load Fonts** aufrufen.

6.2.8.5 Das **Prefs**-Menu

Im Menü **Prefs** (für **Preferences**) können einige Einstellungen (Schalter) vorgenommen werden, die die Arbeit mit DPaint unter gewissen Umständen leichter machen können. Wenn einer dieser "Schalter" auf **An** steht, ist er übrigens mit einem Sternchen versehen, damit Sie dies auch sofort sehen.

Wenn der Schalter **Brush Handle** (engl. für *Pinzel-Griffe*) auf *An* steht, bevor ein neuer Pinsel definiert wird, so "ergreift" der Mauszeiger diesen Pinsel nicht mehr in der Mitte, sondern an der rechten unteren Ecke, was für präzise Arbeiten mit kleinen Pinseln manchmal ganz praktisch ist, weil der Mauszeiger keine Teile des Pinsels mehr verdeckt.

Der Schalter **Coordinates** (engl. für *Koordinaten*) sorgt dafür, daß oben links in der Menü-Leiste stets die genaue Position des Mauszeigers angezeigt wird. Während man die Maus bei gedrückter Maustaste bewegt (zieht) wird die relative Position (Entfernung) des Mauszeigers zu dem Punkt, an dem die Maus "aufgesetzt" wurde, angezeigt.

Der Schalter **Fast Feedback** (engl. für *schnelle Rückkopplung*) schließlich kann das Zeichnen von Linien, Kurven und Flächen mit selbstdefinierten Pinseln leichter machen. Dies kann nämlich bei großen Pinseln teilweise sehr träge vor sich gehen, weil bei jeder Mausbewegung das gerade entstehende Objekt neu gezeichnet und vorher gelöscht werden muß. Bei großen Pinseln dauert das Zeichnen eines längeren Strichs jedoch oft eine Sekunde und mehr. Schaltet man aber auf **Fast Feedback**, so wird z.B. eine Linie immer erst durch einen dünnen Strich angedeutet und erst wenn man die Maustaste losläßt, noch einmal — aber jetzt richtig mit dem ganzen Pinsel — gemalt. Ähnlich wirkt **Fast Feedback** auch auf andere Hilfsmittel.

6.2.9 Farbpaletten-Animation

Ähnlich wie **Graphicraft** bietet natürlich auch DPaint die Möglichkeit zur Farbpaletten-Animation. (Falls Sie den entsprechenden Abschnitt über Graphicraft noch nicht gelesen haben, sollten Sie es spätestens jetzt tun!) Aktiviert wird sie mit dem Befehl **Cycle** im Sub-Menü **Color-Control** des Menüs **Picture** (oder durch einen Druck auf die <TAB>-Taste). Ein sehr schönes Beispiel für den mit Farbpaletten-Animation erzielbaren Effekt bietet das Bild **Waterfall**, das sich auf der Original-DPaint-Diskette in der Schublade **lo-res** befindet. Durch Druck auf die <TAB>-Taste beginnt in diesem Bild der Fluß plötzlich wirklich zu fließen.

Zum Einstellen des Bereichs der Palette, der an der Animation beteiligt sein soll, dient der Palette-Requester.¹² Klicken Sie in diesem zunächst die erste Farbe des gewünschten Bereichs an, dann den Schalter **Range** und dann die letzte Farbe des Bereichs. Mit dem Schieberegler **Speed** können Sie dann

¹² Sie können ihn z.B. durch einen Mausklick mit der rechten Maustaste in den Farbindikator oder durch Druck auf die Taste <P> hervorbringen.

noch die Geschwindigkeit einstellen, mit der die Farben rotieren sollen. Diese Einstellungen sind auch für das Malen mit einem Pinsel im **Cycle**-Modus wichtig. Sie bestimmen, welche Farben dieser wechselt und wie schnell. Der Modus **Cycle** "funktioniert" auch nur dann, wenn die aktuelle Vordergrundfarbe in dem Bereich der Palette liegt, der im Palette-Requester für die Farbpaletten-Animation selektiert wurde.

Dies ist alles noch sehr ähnlich wie bei Graphicraft. Im Gegensatz zu Graphicraft können in DPaint aber drei unterschiedliche Bereiche der Palette gleichzeitig mit unterschiedlichen Geschwindigkeiten rotieren. Hierzu dienen die drei Tasten **C1**, **C2** und **C3** im Palette-Requester. Normalerweise ist die Taste **C1** aktiv und die erste Gruppe kann eingestellt werden. Klickt man aber auf **C2** oder **C3**, so kann die zweite bzw. dritte Gruppe von Farben eingestellt werden. Verstellt werden können der Bereich und die Geschwindigkeit der Rotation — nicht die Farben selbst. Startet man die Animation, nachdem man den Bereich **C2** und/oder **C3** geändert hat, so rotieren alle drei Farbgruppen gleichzeitig.

Auf diese Weise kann man man mehrere verschiedene Objekte in einem Bild sich mit verschiedenen Geschwindigkeiten bewegen lassen. Wählen Sie z.B. im oben schon kurz erwähnten Bild **Waterfall** den Bereich der grauen Farben als zweiten Animationsbereich aus, so beginnen sich neben dem Fluß auch die Wolken darüber zu bewegen.

6.2.10 Tastaturabkürzungen

Viele der Möglichkeiten, die die Kontroll-Leiste bietet, können auch über die Tastatur erreicht werden. Dies ist besonders dann wichtig, wenn Sie den ganzen Bildschirm bearbeiten wollen und nicht nur den Bereich, der von der Titel-Leiste des DPaint-Bildschirms und der Kontroll-Leiste freigelassen wird.

Drücken Sie die Funktionstaste <F10> (in der oberen Zeile auf der Tastatur), so verschwinden Titel-Leiste und Kontroll-Leiste. Sie können sie mit einem erneuten Druck auf <F10> wieder zum Vorschein bringen.

Jetzt haben Sie die Möglichkeit, den ganzen Bildschirm zu bearbeiten. Es wäre aber sehr lästig, wenn Sie jedesmal, wenn Sie die Pinselfarben oder das Werkzeug wechseln wollen, die Kontroll-Leiste wieder zum Vorschein bringen müßten, um sie nach der gewünschten Aktion wieder auszublenden. Deshalb sind (fast) alle Felder der Kontroll-Leiste auch durch einen Tastendruck "simulierbar".

Taste Funktion

F10	Titel-Leiste und Kontroll-Leiste ein-/ausblenden
F9	Titel-Leiste ein-/ausblenden
F8	Fadenkreuz des Pinsels ein-/ausblenden
[	Vorangehende Palettenfarbe für den Vordergrund wählen
]	Nächste Palettenfarbe für den Vordergrund wählen
-	Pinsel größer machen
=	Pinsel kleiner machen
s	Gepunktete Linie
d	Durchgezogene Linie
D	Durchgezogene Linie und kleinste Pinselgröße (ein Punkt)
v	Gerade Linie
q	Kurve
f	Flächen füllen
a	Spraydose
r	Rechteck (hohl)
R	Rechteck (gefüllt)
c	Kreis (hohl)
C	Kreis (gefüllt)
e	Ellipse (hohl)
E	Ellipse (gefüllt)
b	Pinsel auswählen (mit linker Maustaste)
B	Pinsel auswählen (mit rechter Maustaste)
t	Text
g	Gitter
/	Spiegelachsen ein/aus
m	Lupe ein/aus
n	Lupe zurück an die Stelle, wo sie aktiviert wurde
>	Lupe stärker
<	Lupe schwächer
u	UNDO
K	CLR
,	Farbe mit der Maus (aus dem Bild) wählen
p	Palette zeigen

Tabelle 6-1: Tastenbefehle von Deluxe Paint

Zusätzlich gibt es noch die Möglichkeit, die Lupe mit den Cursor-(Pfeil-)Tasten über das Bild zu bewegen. (Dies ist aber keine *Befehls-Abkürzung* im eigentlichen Sinne des Wortes, weil es gar keinen anderen

Befehl oder Mausklick gibt, mit dem man eine Bewegung der Lupe erreichen kann.)

Alle diese Befehle funktionieren aber nur, wenn sich der Mauszeiger nicht über der Menü- oder Kontroll-Leiste befindet. Falls also einer der Befehle scheinbar nicht funktioniert, überprüfen Sie zunächst die Position der Maus!

Diese Befehle sollten Sie sich sehr gut einprägen, wenn Sie häufiger mit DPaint arbeiten wollen. Es ist sehr praktisch — auch ohne ausgeschaltete Kontroll-Leiste — diese Befehle mit einer Hand auf der Tastatur zu geben, und die Maus mit der anderen Hand immer über den Stellen zu halten und zu bewegen, die Sie gerade bearbeiten.

6.3 Textcraft

Eine der Anwendungen, für die Microcomputer heutzutage am meisten benötigt werden, ist die Textverarbeitung. Auch dieses Buch entstand natürlich auf einem Textverarbeitungssystem, weil die Vorzüge gegenüber der klassischen Schreibmaschine einfach überwältigend sind. Alles Geschriebene erscheint vor allem erst einmal auf dem Bildschirm und kann zig-mal korrigiert, umgestellt und überarbeitet werden, bevor man es dem Papier anvertraut. Besonders wichtig sind vor allem die vielfältigen Möglichkeiten von Textumstellungen und Umformulierungen. Solche Änderungen verlangen bei der Arbeit mit der Schreibmaschine meistens ein Neutippen großer Teile des Textes und werden daher typischerweise nur sehr zögerlich angegangen. Solche Hemmungen gibt es bei der Textverarbeitung auf einem Computer nicht und deshalb ist man auch mit dem Endprodukt seiner Arbeit oft viel zufriedener.

Das erste Textverarbeitungssystem für den Amiga ist *Textcraft* und es ist gleich ein recht "guter Wurf" geworden. Textcraft ist leicht zu erlernen, aber doch vielseitig und leistungsfähig genug, um praktisch für jeden, der Textverarbeitung in kleinerem Maße betreibt, ausreichend zu sein. Es ist vielleicht nicht gerade ein Programm, mit dem ein Buch wie dieses erstellt werden könnte, für den Privatbedarf und den Einsatz in kleineren Betrieben für Briefe, Lebensläufe, Berichte, Rechnungen usw. aber ideal geeignet.

6.3.1 Ein erster Überblick über Textcraft

Die Textcraft-Original-Diskette ist startfähig, d.h. Sie können Sie nach dem Einlesen der Kickstart-Diskette sofort einschieben und gelangen dann auf die Workbench. Wenn Sie die Diskette öffnen, finden Sie zwei Versionen des Programms. Das Icon **Textcraft** ist eine Version, die auch auf dem "kleinen" Amiga mit nur 256K Speicher läuft und **Textcraft512** enthält eine Version, die nur für den Amiga mit mindestens 512K Speicher gedacht ist und auf diesem etwas schneller arbeitet. Sie können die Version, die nicht für Sie zutrifft, natürlich löschen, um Platz zu sparen (natürlich nur auf der Kopie der Original-Diskette, die Sie zuvor anfertigen müssen!). Falls Sie **Textcraft** löschen, sollten Sie danach **Textcraft512** in **Textcraft** umbenennen (mit **Rename**).

Wie jedes gute Werkzeug kann Textcraft entweder selbst geöffnet werden oder über ein Projekt-Icon. Auf der Diskette befindet sich bereits ein solches Dokument namens **Read Me For News** (*Lies mich wegen Neuigkeiten*). Um einen ersten Eindruck von den Möglichkeiten des Programms zu bekommen, sollten Sie am besten gleich dieses Projekt öffnen.

Wollen Sie Textcraft wieder verlassen und auf die Workbench zurückkehren, so dient dazu üblicherweise der Befehl **Quit Textcraft** im **Project**-Menü (s.u.). Gegenüber der Workbench verhält sich Textcraft übrigens manchmal etwas anders als andere Programme. Unter gewissen Umständen schließt Textcraft nämlich beim Starten die Workbench und öffnet sie erst wieder beim Verlassen des Programms. Sie können die Workbench dann auf keinerlei Weise sichtbar machen, ohne das Programm zu verlassen.

Die Bedienung von Textcraft enthält viele Aspekte, die Sie von anderen Programmen nun bereits kennen. Der Text wird z.B. in einem großen, fast den ganzen Bildschirm füllenden Rechteck gezeigt, das Sie mit Hilfe von Rollbalken, die sich am unteren und am rechten Rand dieses Fensters befinden über eine potentiell sehr große Fläche verschieben können. Die Rollbalken sehen etwas anders aus als von der Workbench gewohnt, sind aber unverkennbar. Ihre Bedienung dürfte deshalb niemandem, der sie von der Workbench kennt, Schwierigkeiten bereiten.

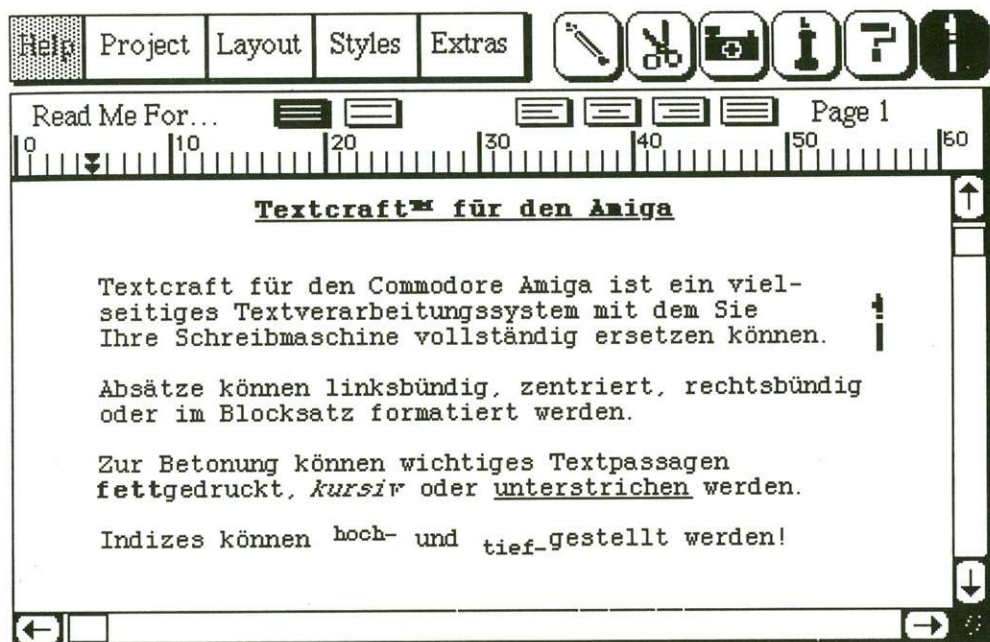


Bild 6 - 11: Textcraft

Auch die Menüs von Textcraft sehen etwas anders aus als gewohnt, verhalten sich aber recht ähnlich wie die Workbench-Menüs. Textcraft besitzt nur fünf Menüs, die auch nicht in einer Menü-Leiste liegen, sondern eine immer sichtbare Gruppe von fünf kleinen Rechtecken am oberen linken Rand des Bildschirms bilden.

Das erste dieser Menüs, **Help**, hat eine besonders auffällige Färbung und dient dazu, Ihnen Hilfestellung bei der Bedienung des Programms zu geben. Die augenblicklichen Hilfestellungen sind zwar noch in Englisch, aber wenn Sie dieser Sprache einigermaßen mächtig sind, kann das **Help**-Menü ein recht guter Ersatz für das Handbuch werden. Die ersten beiden Befehle in diesem Menü, **Quick Reminder** und **Keyboard Reference**, zeigen jeweils ein Bild, in dem einige wichtige Aspekte des Programms in einer Form zusammengestellt sind, die vor allem als Gedankenstütze beim täglichen Einsatz gedacht ist.

Der letzte Punkt im **Help**-Menü, **1 Minute Tutorials**, ist vor allem für das Erlernen des Programms gedacht. Nachdem er aus dem Menü ausgewählt wurde, erscheint ein Requester, in dem ca. 20 Aufgaben zur Auswahl angeboten werden, über die man sich informieren kann. Wählt man eines dieser Themen (durch Anklicken) aus und betätigt dann die Taste **Start Tutorial**, so wird man von der Tutorial-Funktion durch die Bedienung des Programms geführt. Hierzu "übernimmt" die Funktion den Amiga völlig, steuert auch den Mauszeiger und macht Ihnen damit alle Schritte am Bildschirm vor, die nötig sind, um die vorher ausgewählte Aufgabe durchzuführen. Diese Tutorials sind eine sehr plastische Methode, sich mit der Bedienung des Programms vertraut zu machen. Falls Sie bereits Textverarbeitungssysteme auf anderen Computern kennen (und mit der englischen Sprache etwas vertraut sind) reichen diese Tutorials und die beiden anderen Help-Funktionen wahrscheinlich völlig aus, um Textcraft zu erlernen!

Neben den sechs Menü-Kästchen befinden sich sechs kleine Icons, die besonders wichtige Funktionen des Programms einleiten, wenn man sie anklickt. Sie entsprechen gleichzeitig bestimmten Formen des Maus-Zeigers, die dieser immer dann annimmt, wenn eine dieser Funktionen aktiv ist. Sie können so jederzeit an der Form des Mauszeigers erkennen, in welchem Zustand sich das Programm gerade befindet.¹³

In welchen Zustand die einzelnen Icons den Maus-Zeiger versetzen, erfahren Sie im Detail noch in den folgenden Abschnitten. Bedient werden sie wie ein Mini-Menü. Ein Mausklick **mit der Menü-Taste** aktiviert sie und der Mauszeiger nimmt danach die entsprechende Form an.

6.3.2 Text eingeben

Das Eingeben von Text funktioniert in Textcraft im wesentlichen genauso wie im Notepad oder auch in einem Text-Gadget. Ein senkrechter roter Strich markiert die aktuelle Schreibmarke, also die Stelle, an der neu eingegebener Text erscheint. Diese Schreibmarke kann entweder mit den Cursor- (Pfeil-) Tasten bewegt werden oder durch Anklicken einer Stelle zwischen zwei Buchstaben mit dem Mauszeiger (sofern der Mauszeiger gerade die Form eines kleinen Bleistifts hat). Die Schreibmarke springt dann sofort zu dieser Stelle. Hat der Mauszeiger eine andere Form als den Pinsel, so kann es

¹³ Eine Aufstellung über die Formen und ihre Bedeutung bekommen Sie gezeigt, wenn Sie den Befehl **Quick Reminder** (*Schnelle Erinnerung*) aus dem **Help**-Menü auswählen.

trotzdem sein, daß diese Positionierung der Schreibmarke so funktioniert. Besser ist es aber, zuvor auf das Icon mit dem Bleistift rechts neben den Menüs zu klicken (wodurch der Mauszeiger zum Bleistift wird).



Bild 6 - 12: Der Bleistift

Auch die Korrektur kleiner Tippfehler funktioniert wie im Notepad-Werkzeug. Die <BACKSPACE>-Taste löscht immer den zuletzt eingegebenen Buchstaben bzw. den Buchstaben links von der Schreibmarke. Die -Taste löscht den Buchstaben rechts der Schreibmarke.

Textcraft bietet eine (relativ) komfortable Lösung, nationale Sonderzeichen einzugeben.¹⁴ Hiermit sind vor allem solche Sonderzeichen gemeint, die "über" dem normalen Buchstaben noch einen Akzent tragen. Um solche Buchstaben zu erzeugen, wird eine der beiden <ALT>-Tasten festgehalten und dann eine Taste gedrückt, die dem Akzent ähnlich sieht. Danach ist auf dem Bildschirm nichts zu sehen. Erst wenn dann eine Taste gedrückt wird, die den Buchstaben trägt, der unter dem Akzent stehen soll, erscheint der Buchstabe zusammen mit dem Akzent. (Wollte man z.B. auf der englischen Tastatur ein Ö tippen, müßte man zunächst <ALT><SHIFT> tippen und dann das O.)

6.3.3 Text ändern

Größere Änderungen im Text führt man im allgemeinen jedoch nicht durch Löschen und Neutippen durch (wie man es ja von der Schreibmaschine und TippEx her kennt), sondern mit vielseitigeren elektronischen Hilfsmitteln. Wollen Sie nicht nur ein Wort ändern, sondern z.B. löschen und umstellen, sollten Sie sich dieser Hilfsmittel bedienen.

Diese Hilfsmittel sind die kleinen Icons, die rechts oben am Bildschirm in angedeuteter Form zu sehen (und hoffentlich auch zu erkennen) sind.:

¹⁴ Bei der ersten (amerikanischen) Version des Amiga fallen natürlich auch Umlaute in die Kategorie *nationale Sonderzeichen*. Auch Sie müssen also mit dem beschriebenen "Trick" erzeugt werden.

- eine Schere
- eine Kamera
- eine Tube Klebstoff

Diese drei Hilfsmittel entsprechen in etwa den Befehlen **Cut** (*Ausschneiden*), **Copy** (*Kopieren*) und **Paste** (*Einfügen, Einkleben*), wie wir sie bereits im Programm *Graphicraft* kennengelernt haben.



Bild 6 - 13: *Die Schere*

Wollen Sie z.B. ein Stück eines längeren Textes löschen, so klicken Sie einfach (mit der rechten Taste) auf die Schere, worauf der Mauszeiger zur Schere wird. Klicken Sie nun vor den ersten Buchstaben, der gelöscht werden soll und halten sie die (linke) Maustaste fest.¹⁵ Ziehen Sie nun den Maus-Zeiger hinter den letzten Buchstaben, der gelöscht werden soll. Sobald die Taste losgelassen wird, verschwindet der so selektierte Text und wird zugleich in die Zwischenablage (engl. *clipboard*) gelegt. Um den Text nur in die Zwischenablage zu legen, verwendet man statt der Schere die Kamera.



Bild 6 - 14: *Die Kamera*

Mit der Schere oder der Kamera können Textstellen auch umgestellt oder vervielfacht werden. Hierzu wird der Text erst mit der Schere oder der Kamera in die Zwischenablage befördert und dann die Klebstoff-Tube angeklickt. Klickt man mit der Spitze der Tube (zu der der Maus-Zeiger geworden ist) nun an eine Stelle im Text, so wird der Inhalt der Zwischenablage dort eingefügt. Die Zwischenablage wird dadurch jedoch nicht "verbraucht". Sie kann durch weitere Klicks mit der Klebstoff-Tube noch an mehreren anderen Stellen eingefügt werden, wodurch ein schnelles Vervielfältigen auch großer Textstellen problemlos möglich ist.



Bild 6 - 15: *Die Klebstoff-Tube*

¹⁵ Der heiße Punkt der Schere ist ihre linke Spitze.

6.3.4 Textcraft-Projekte

Nachdem Sie nun die ersten Grundzüge von Textcraft kennengelernt haben, wollen wir nun auch kurz auf das Abspeichern und Ausdrucken Ihrer Werke eingehen. Die Verwaltung von kompletten Textcraft-Projekten geschieht mit dem **Project**-Menü, dessen Inhalt und Funktion nahezu mit dem entsprechenden Menü von Graphicraft identisch ist.

Der Befehl **New Document** dient zum Anlegen eines neuen Textes. Da Textcraft immer nur einen Text bearbeiten kann, muß ein Text, der vielleicht schon bearbeitet wurde, zuvor gesichert werden (s.u.). Nach dem Aufruf von **New Document** erscheint ein Requester, in dem man eine Auswahl verschiedener Dokument-Typen gezeigt bekommt. Voreingestellt ist **Blank Document**, die anderen Optionen werden im Abschnitt über *Text-Formulare* noch etwas erläutert.

Mit **Open Document** kann ein anderes, irgendwann zuvor einmal abgespeichertes Dokument geöffnet werden. Die auf der Diskette zur Verfügung stehenden Texte werden dazu in einem Requester gezeigt, in dem man sich einen davon aussuchen kann.

Save Document dient zum Abspeichern des gerade bearbeiteten Textes auf der Diskette bzw. Festplatte. Hatte das Dokument zuvor schon einen Namen, so wird dieser auch als neuer Name vorgeschlagen. Man kann ihn jedoch auch ändern und so den Text unter einem anderem Namen abspeichern.

Mit **Print Document** wird der gerade bearbeitete Text ausgedruckt. Zuvor erscheint ein Requester, in dem man auswählen kann, ob ein grober Entwurf (**Draft**) oder bereits das endgültige Dokument (**Final**) gedruckt werden soll, ob der Ausdruck mit möglichst kleinen (**Small**), mittelgroßen (**Medium**) oder großen (**Large**) Buchstaben ausgedruckt werden soll und ob auf einzelne Blätter oder Endlospapier gedruckt werden soll. Dieser Requester bleibt während des gesamten Druckvorgangs sichtbar. Durch Betätigung der Taste **Cancel** unten rechts im Requester kann der Druckvorgang jederzeit abgebrochen werden.

Wie ein Ausdruck genau aussieht, hängt natürlich stark vom verwendeten Drucker ab. Je nach Drucker kann es durchaus sein, daß einige der Stil-Optionen (s.u.) oder bestimmte Sonderzeichen nicht gedruckt werden können und nur am Bildschirm sichtbar sind. Die beste Annäherung an das Aussehen des Textes am Bildschirm ist meist mit einem grafikfähigen Nadeldrucker zu erreichen, der aber für viele Anwendungen ein zu computerhaftes Druckbild zeigt. Typenraddrucker bieten zwar ein besseres Druckbild, aber meist

weniger Variationsmöglichkeiten für den Text-Stil und haben oft Schwierigkeiten mit Sonderzeichen.¹⁶

Der letzte Befehl des **Project**-Menüs, **Quit Textcraft**, beendet natürlich das Programm. Existieren aber noch Änderungen im Text, die zuvor nicht gesichert wurden, erscheint ein Requester, in dem man es sich "noch einmal anders überlegen" und ins Programm zurückkehren kann.

6.3.5 Ändern des Text-Stils

Textcraft ist im Vergleich zu ähnlichen Systemen auf anderen Computern ein recht komfortables Textverarbeitungsprogramm. Zur Hervorhebung von Textpassagen können diese nämlich auch **fett**, unterstrichen oder *kursiv* gedruckt werden, und (für Indizes) hoch- oder tiefgestellt werden. Alle diese Stil-Optionen (außer hoch- und tiefstellen) können zudem auch miteinander kombiniert werden.¹⁷

Zum Anwenden der verschiedenen Stil-Mittel benötigt man das **Styles**-Menü und das Pinsel-Icon ganz rechts oben am Bildschirm. Zunächst wählt man aus dem Style-Menü die Stil-Optionen aus, die man verwenden will. Die einzelnen Menü-Punkte im **Styles**-Menü sind "Schalter", d.h. sie werden mit einem kleinen Häkchen versehen, sobald sie aktiv bzw. gültig sind.

Nachdem im **Styles**-Menü alle Optionen abgehakt sind, die man verwenden möchte, klickt man (oder auch vorher) auf den Pinsel und selektiert danach (wie oben schon für die Schere beschrieben) den zu ändernden Text. Sobald man die Maustaste losläßt, erhält der Text die gewünschten Attribute. Der Pinsel bleibt danach übrigens aktiv. Wollen Sie also weitere Passagen mit demselben Stil versehen, so können Sie dies sofort tun, ohne die Stil-Optionen und den Pinsel erst neu auswählen zu müssen.

¹⁶ Man muß hierbei auch bedenken, daß für amerikanische Geräte auch ein Umlaut oder ein 'ß' ein Sonderzeichen ist.

¹⁷ Bedauerlicherweise verfügt Textcraft jedoch nicht über die Möglichkeit, verschiedene Zeichensätze zu verwenden, wie es das Werkzeug Notepad vorführt. Dies liegt vor allem daran, daß diese Zeichensätze nicht mit jedem Drucker zu Papier zu bringen sind.



Bild 6 - 16: Der Pinsel

Diese Möglichkeiten sind zunächst sehr verlockend – vor allem auch, weil sie von einer Schreibmaschine üblicherweise nicht geboten werden. Man sollte aber versuchen, sparsam damit umgehen. Zu intensive Verwendung verschiedener Stile und ihrer Kombinationen verbessern nicht die Lesbarkeit eines Textes, sondern machen ihn im Gegenteil zu unruhig.

6.3.6 Formatieren des Textes

Aber auch in anderer Hinsicht bietet ein Textverarbeitungsprogramm ganz neue Möglichkeiten. Professionell erstellte Texte sind ja z.B. nicht immer nur linksbündig wie Schreibmaschinen-Texte, sondern oft auch rechtsbündig, zentriert oder sogar am Blocksatz. Zwar kann man dies alles auch mit einer Schreibmaschine erreichen (vor allem, wenn sie teuer war), ein Programm kann die entsprechende Buchstabenzählerei aber ungleich leichter und schneller erledigen.

Zur Formatierung von Text dient vor allem das Lineal, das über dem Text-Rahmen zu sehen ist. Dieses Lineal bestimmt das *Absatzformat*. Um zur verstehen, was das Absatzformat ist, muß aber ein wenig weiter ausgeholt werden.

Ein weiterer entscheidender Unterschied der Textverarbeitung zur Schreibmaschine ist der Zeilenvorschub. Bei den meisten Schreibmaschinen muß man ja am Ende jeder Zeile von Hand dafür sorgen, daß eine neue Zeile begonnen wird; entweder mit einem mechanischen Hebel oder (bei elektrischen Schreibmaschinen) mit der <RETURN>-Taste. Diese <RETURN>-Taste gibt es auch auf der Tastatur des Amiga; man benötigt sie jedoch nicht für den Zeilenvorschub. Wenn ein Wort am Ende einer Zeile beim Tippen zu lang wird, um noch in die Zeile hineinzupassen, legt Textcraft es von ganz allein in die nächste Zeile. Sie tippen deshalb im allgemeinen den Text "einfach durch" und achten gar nicht auf den Zeilenwechsel. Nur am Ende eines Absatzes (engl. *Paragraph*) muß die <RETURN>-Taste einmal betätigt werden.

Dieses Schreiben in Absätzen ist eigentlich einfacher als das manuelle Ausführen des Zeilenschubes. Vielen Anfängern der Textverarbeitung wird es aber zweifellos viele Umstellungsprobleme bereiten, weil es eben ganz anders

ist als der Umgang mit der Schreibmaschine. Man sollte sich aber auf alle Fälle zur dieser Arbeitsweise zwingen. Sie ist enorm wichtig für eine einfache Nutzung der sonstigen Möglichkeiten, die die Textverarbeitung auf einem Computer bietet!

Löschen Sie z.B. ein Wort in einem Absatz, wird die entsprechende Zeile dadurch natürlich kürzer. Wird sie kurz genug, so "holt" Textcraft automatisch das erste Wort der folgenden Zeile nach oben und sorgt so für gleichmäßig lange Zeilen — aber nur, wenn zwischen den beiden Zeilen kein <RETURN> eingegeben wurde.

Ein Absatz ist für Textcraft ein eigenständiges "Objekt" mit einem eigenen Format, das anders sein **kann** als das von anderen Absätzen. Mit verschiedenen Gadgets innerhalb des Lineals kann man das Format des Absatzes, in dem sich die Schreibmarke gerade befindet, jederzeit ändern.

Drei kleine Dreiecke am unteren Rand des Lineals bestimmen z.B. den linken und rechten Rand des Absatzes und den linken Rand für die erste Zeile des Absatzes (den sogenannten *Einzug*). Die beiden orangen Dreiecke sind für den rechten und linken Rand "zuständig", das schwarze für den Einzug. Jedes dieser drei Dreiecke kann mit dem Mauszeiger wie ein Icon auf der Workbench ergriffen werden (wobei er die Form einer Hand annimmt) und innerhalb des Lineals bewegt werden. Für präzise Einstellungen können die Teilungsstriche im Lineal verwendet werden, die als Einheit die Breite eines Buchstabens verwenden.

Sind diese Einstellungen erst einmal gewählt, so hält Textcraft sie beim Tippen automatisch ein. Alle im Anschluß daran neu begonnenen Absätze erhalten ebenfalls dieses Format — Sie brauchen es also nicht jedesmal wieder neu einzustellen. Ändert man die Einstellungen jedoch nachträglich, wird der gesamte Absatz automatisch umformatiert. Man braucht sich also nicht schon beim Tippen eines Textes für ein bestimmtes Absatzformat zu entscheiden, sondern kann diese mehr kosmetischen Aufgaben an das Ende der Arbeit verschieben.

Am oberen Rand des Lineals befinden sich sechs kleine Kästchen (Icons), die mit ihrem Aussehen schon in etwa andeuten, wozu sie gut sind. Die beiden Kästchen links von der Mitte dienen zur Auswahl zwischen einzeilig und zweizeilig gedruckten Absätzen. (Zur Hervorhebung eines ganzen Absatzes oder zur Vereinfachung von Korrekturen mit dem Bleistift kann es ja günstiger sein, ihn zweizeilig zu drucken.)

Die vier Kästchen rechts von der Linealmitte ermöglichen es, den Text in einem Absatz linksbündig, rechtsbündig, zentriert oder im Blocksatz (rechts- und linksbündig) zu setzen. Auch für diese Formatierungen sorgt wieder

Textcraft — Sie brauchen sie also **nicht** durch manuelles Einfügen von Leerzeichen zwischen den Buchstaben mühsam zu simulieren.

Eine zweite Möglichkeit, diese Formateinstellungen durchzuführen, ist, den Befehl **Set Ruler Margins** aus dem **Layout**-Menü einen Requester aufzurufen, in dem man sie in etwas anderer Form eingeben kann. Im allgemeinen dürfte das Lineal aber der wohl bequemere Weg sein. Während man mit Einstellungen im Lineal aber im allgemeinen nur einen Absatz umformatieren kann, bietet der Requester die Wahl zwischen der Änderung nur des aktuellen Absatzes (engl. **Current Paragraph**) und aller Absätze im ganzen Text (engl. **Entire Document**).

Sollen mehrere Absätze — aber nicht gleich der ganze Text — umformatiert werden, so kann die Farbrolle (das fünfte Icon rechts neben den Menüs) eingesetzt werden. Hierzu stellt man das Format für zunächst einen Absatz so ein, wie man es haben möchte. Dann klickt man auf die Farbrolle und selektiert nun (wie oben beschrieben) alle Absätze, die das neue Format erhalten sollen. Sobald die (linke) Maustaste losgelassen wird, bekommen alle so ausgewählten Absätze das neue Format.



Bild 6 - 17: Die Farbrolle

Wenn Sie die Formatierungsmöglichkeiten von Textcraft wirklich kennenlernen und ein Gefühl dafür bekommen wollen wie sie wirken, sollten Sie am besten einen schon vorhandenen längeren Text nehmen und diesen umformatieren. Hierfür eignet sich sehr gut das Projekt **Read Me For News**, das ja bei Textcraft gleich mitgeliefert wird. Es enthält allerdings auch wichtige Informationen über die jeweils neueste Programmversion. Deshalb sollten Sie es besser vorher (auf der Workbench) kopieren und die Experimente mit dieser Kopie durchführen.

6.3.7 Seitenformat

Eine weitere wichtige Formatierungsfunktion eines Textverarbeitungsprogramms ist die Anordnung des Textes auf den ausgedruckten Seiten. Textcraft erlaubt es, alle Seiten automatisch mit Seitennummern zu versehen und oben und unten auf jede Seite eine Zeile drucken zu lassen, die zwar auf allen Seiten erscheint, aber nur einmal eingegeben zu werden braucht.

Hierzu dient der Befehl *Set Page Number and Titles* (Setze Seitenzahl und Seitentitel) aus dem Menü *Layout*.. Daraufhin erscheint wieder ein Requester, in dem Sie eine sogenannte *Kopfzeile* (engl. *Header*) und eine *Fußzeile* (engl. *Footer*) eingeben können, die auf jeder Seite des Ausdrucks erscheinen werden. Sowohl die Kopfzeile als auch für die Fußzeile können linksbündig (*Left*), zentriert (*Center*) oder rechtsbündig (engl. *Right*) auf der Seite erscheinen. Dies kann mit jeweils drei kleinen Schaltern, die unter den Text-Gadgets, in denen die Kopf- bzw. Fußzeile eingegeben wird, erscheinen, angegeben werden. Zwischen den Angaben für die Kopf- und Fußzeile befinden sich eine Reihe von Schaltern, die das Aussehen und die Position der Seitenzahl bestimmen. Auch die Seitenzahl, mit der die Numerierung beginnt (engl. *Starting Page Number*) kann festgelegt werden, damit man z.B. sehr lange Texte auch kapitelweise bearbeiten kann.

Neben den diversen Schaltern und Text-Gadgets befindet sich noch ein kleines Rechteck in diesem Requester. In diesem Rechteck wird stets in stilisierter Form gezeigt, wie eine ausgedruckte Seite bei den augenblicklich gültigen Einstellungen aussehen würde.

Bevor ein so formatierter Text aber wirklich ausgedruckt werden kann, benötigt Textcraft noch eine weitere Information. Da der Ausdruck seitenweise erfolgt, muß Textcraft für die Formatierung noch wissen, wieviele Zeilen auf eine Seite passen. Es gibt nämlich eine ganze Reihe verschiedener Papierformate, die Computerdrucker bewältigen können. Zur Einstellung der Seitengröße dient der Befehl *Choose Page Size* (engl. für *Wähle Seitengröße*) im *Layout*-Menü. Eine Reihe von Papiergrößen werden hier bereits vorgeschlagen; es sind jedoch fast alle Größen, die in Europa nicht gebräuchlich sind. Papierlängen, die man hier nicht vorgeschlagen findet, kann man glücklicherweise aber im Text-Gadget hinter *Special* angeben. (Eine gebräuchliche Größe für Endlos-Druckerpapier sind z.B. 72 Zeilen.)

Da es im allgemeinen nicht gut aussieht, wenn sich der Text bis an die obere und untere Kante der Seite erstreckt, können sowohl für den oberen Rand (engl. *Top Margin*) als auch für den unteren Rand (engl. *Bottom Margin*) eine Anzahl von Zeilen angegeben werden, die frei bleiben sollen.

Textcraft berücksichtigt alle diese Einstellungen aus den beiden Requestern *Choose Page Size* und *Set Page Number and Titles* aber nicht nur beim Ausdruck sondern auch bei der Anzeige am Bildschirm. Ein Seitenwechsel wird auch am Bildschirm immer durch eine gestrichelte Linie angedeutet, die quer über den Bildschirm geht. Sie haben so stets die Kontrolle darüber, welcher Teil des Textes auf eine neuen Seite kommt und können Umformatierungen vornehmen, wenn bestimmte Textpassagen zusammen auf einer Seite stehen sollen.

6.3.8 Text-Formulare

Textcraft bietet nicht nur sehr reichhaltige Formatierungsmöglichkeiten. Es gibt auch Hilfestellung bei der Formatierung häufig vorkommender Formen von Dokumenten. Hierzu dienen die Optionen des Requesters, der erscheint, wenn man den Befehl **New Document** aus dem Menü **Project** wählt.

Die erste Option, die in diesem Requester erscheint, ist ja **Blank Document** (*Leeres Dokument*). Darunter folgen jedoch noch eine ganze Reihe weiterer Schalter, die eine deutlich andere Wirkung haben, wenn man sie auswählt und dann den Knopf **OK** betätigt. Die Optionen sind:

- Memorandum
- Geschäftsbrief (engl. **Business Letter**)
- Technischer Bericht (engl. **Technical Report**)
- Geschäftlicher Bericht (engl. **Business Report**)
- Studienarbeit (engl. **Term Paper**)
- Lebenslauf (engl. **Resume**)

Wenn Sie einen dieser Dokumentformen auswählen, werden Sie am Anschluß daran in einer Art Formular nach einigen wesentlichen Bestandteilen des neuen Dokuments gefragt (bei einem Brief z.B. Adressat, Anrede, Datum, Absender, usw.). Wenn Sie dieses Formular dann ausgefüllt haben und wieder die Taste **OK** betätigen, erscheint der normale Textcraft-Bildschirm. Das Dokument ist aber nun nicht leer wie bei Blank Document, sondern enthält bereits die eingegebenen Informationen und einigen zusätzlichen Text. Zudem sind die verschiedenen Bestandteile bereits in einem ansprechenden Format angeordnet, wie es bei der entsprechenden Dokumentenform üblich ist. Sie selbst brauchen nun nur noch den eigentlichen Inhalt des Briefs einzutippen und ihn dann abzuspeichern und/oder auszudrucken.

Bei Berichten haben Sie sogar die Möglichkeit, auch Literaturstellen (für Bücher und Artikel in unterschiedlicher Form) anzugeben und auch diese gleich formatiert zu bekommen. Es kann zwar nur eine Literaturstelle in einem Formular angegeben werden, ihr Format kann jedoch natürlich mit Kamera und Klebstoff problemlos dupliziert werden. Berichte und Studienarbeiten erhalten zudem auch gleich eine besondere Titelseite und eine Seite für eine Zusammenfassung, die ebenfalls bereits speziell formatiert sind.

Diese Formularfunktion erspart Ihnen normalerweise eine Menge Arbeit und hat zudem den Vorteil, daß alle Briefe, Berichte, Memoranden, usw. die Sie so erstellen, ein einheitliches Aussehen haben. Sie sind andererseits in keiner Weise eingeschränkt, da sowohl das Format als auch der Text nachträglich noch beliebig geändert werden können.

Ein Problem besteht zumindest bei der augenblicklichen englischen Version von Textcraft, die aber hoffentlich bald "eingedeutscht" wird. Die Dokumententypen sind stark auf amerikanische Verhältnisse zugeschnitten. Einige davon würde man bei uns in dieser Form nicht verwenden. Zudem sind natürlich auch die Text-Bausteine, die Textcraft selbst einsetzt, wie z.B. Anreden, anglo-amerikanisch (z.B. Mr., Mrs., usw.) und müssen nachträglich geändert werden. Dies ist — zumindest im Moment — ein kleiner Wermutstropfen bei dieser sehr nützlichen Einrichtung von Textcraft.

6.3.9 Textpassagen finden und ersetzen

Eine der Funktionen, für die die elektronische Textverarbeitung geradezu berühmt ist, ist das schnelle Aufsuchen bestimmter Textpassagen und das Austauschen der gefundenen Vorkommnisse gegen einen anderen Text.

Nehmen wir einfach einmal an, Sie hätten einen längeren Brief geschrieben und Ihren Ansprechpartner immer *Herrn Meier* genannt. Auf einmal erfahren Sie, daß *Herr Meier* ein *Doktor Meier* ist, und auf diesen Titel auch gesteigerten Wert legt. Bei einem auf einer Schreibmaschine geschriebenen Brief heißt dies jetzt "Neutippen!"; bei einem mit einer Textverarbeitung erstellten ist jedoch mit ein paar Handgriffen alles wieder in Ordnung und der Brief braucht nur einmal neu ausgedruckt zu werden.

Wollen Sie beim Arbeiten mit Textcraft in einem längeren Text eine bestimmte Formulierung finden, so brauchen Sie dazu nur den Befehl **Find and Replace Text** (engl. für *Text suchen und ersetzen*) aus dem Menü **Extras** aufzurufen. Daraufhin erscheint ein Requester mit zwei Text-Gadgets und mehreren Tasten. Im ersten Text-Gadget (hinter **Find:**) können Sie nun eingeben, welchen Text Sie suchen wollen. Sie brauchen dabei nicht auf Groß- und Kleinschreibung zu achten — solche "unwichtigen Details" ignoriert Textcraft großzügig bei seiner Suche.

Klicken Sie danach auf die Taste **Find Next**, und Textcraft sucht die eingegebene Textpassage im Dokument, wobei die Suche von der Schreibmarke an abwärts geht. (Oberhalb der Schreibmarke findet Textcraft nichts!) Das erste Vorkommen der gesuchten Textpassage wird dann angezeigt. Hierzu wird das Dokument — falls nötig — auch "hinter" dem Requester aufwärts bewegt, damit Sie die Stelle sehen können, wo die gesuchte Passage gefunden wurde. Sie können den Requester nun mit **Cancel** abschließen — worauf die Schreibmarke hinter dem letzten Buchstaben des gefundenen Textes stehen bleibt. Sie können aber auch durch erneuten Druck auf **Find Next** das nächste Vorkommen suchen. (Falls

Textcraft den eingegebenen Text nicht finden kann, zeigt es die Meldung *Couldn't find it.* — zu deutsch *Ich konnte es nicht finden*)

Wollen Sie aber Textstellen nicht nur suchen, sondern sie gegen eine andere Formulierung ersetzen (z.B. *Herr* durch *Dr.*), so müssen Sie den zu ersetzenden Text in das zweite Text-Gadget eingeben (hinter **Replace with:** – zu deutsch *Ersetze durch*). Nachdem Sie mit der Taste **Find Next** das erste Vorkommnis des gesuchten Textes gefunden haben, können sie es durch Druck auf **Replace** dann sofort durch den neuen Text ersetzen lassen. So können Sie nacheinander alle Vorkommnisse der alten Formulierung durch die neue ersetzen lassen. (Mit **Find Next** können Sie allerdings auch Vorkommnisse überspringen und nicht ersetzen lassen.)

Wollen Sie jedoch alle Vorkommnisse des gesuchten Textes durch eine neue Formulierung ersetzen lassen, so müssen Sie statt auf **Find Next** auf **Replace All** (engl für *Alle [Vorkommnisse] ersetzen*) drücken. Textcraft durchläuft dann das gesamte Dokument und tauscht die gesuchte Textpassage aus. Dabei wird in einem kleinen Fenster kontinuierlich angezeigt, wieviele Vorkommnisse schon gefunden (und ersetzt) wurden.

Zum Schluß aber noch eine Warnung vor der Replace-Funktion: Überlegen Sie es sich sehr genau, bevor Sie auf **Replace All** klicken! Sichern Sie vorher am besten einmal das alte Dokument mit dem Befehl *Save Document!*

Es ist nämlich nicht immer möglich, einen solchen *globalen Ersetzungsbefehl* durch den "gegenteiligen" Befehl wieder rückgängig zu machen, wie man vielleicht auf den ersten Blick meint. Warum dies so ist, werden Sie mit ein klein wenig Überlegung — oder einigen Versuchen in dieser Richtung — bestimmt bald selbst herausfinden. (Seine Fehler muß jeder selbst machen!)

6.3.10 Tastaturabkürzungen

Gerade bei der Textverarbeitung kann es sehr uneffizient sein, wenn man andauernd seine Hände von der Tastatur nehmen muß, um mit der Maus ein Icon oder einen Menü-Punkt auszuwählen. Dies hemmt den Arbeitsfluß, obwohl es in der Lernphase mit dem Programm ein großer Vorteil ist.

Die Entwickler von Textcraft haben dies aber glücklicher vorausgesehen und eine Lösung für dieses Problem gleich in das Programm eingebaut. Ähnlich wie in Deluxe Paint kann fast jede Funktion des Programms, die über ein Menü oder einen Mausklick in ein Icon erreicht werden kann, auch durch spezielle Tastendrucke oder Tastenkombinationen erreicht werden.

Für die Menü-Punkte werden dabei die bekannten *Tastatur-Äquivalente* verwendet, wie sie im Kapitel *Intuition* erläutert wurden. Dies ist auch als Gedankenstütze sehr praktisch, weil man immer, wenn man einmal eine Tasten-Kombination vergessen hat, sofort im entsprechenden Menü wieder nachschauen kann.

Ein Klick mit der Menü-Taste auf die sechs Hilfsmittel rechts neben den Menüs (Schere, Kamera, etc.) kann mit den Funktionstasten am oberen Rand der Tastatur "simuliert" werden. Die Tasten F5 bis F10 entsprechen den Funktions-Icons Bleistift bis Pinsel. Diese "einfachen" Abkürzungen sind alle in dem Bild verzeichnet, das man zu sehen bekommt, wenn man den Punkt **Keyboard Shortcuts** aus dem Menü **Help** auswählt.

Darüberhinaus gibt es aber noch eine Vielzahl anderer Abkürzungen z.B. für das Bewegen der Schreibmarke über größere Strecken hinweg und das Bewegen (*Scrollen*) des Textes. Es sind aber zu viele, als daß eine Liste davon an dieser Stelle praktisch wäre.

Ein Tip zu diesem Thema aber trotzdem noch zum Abschluß: Obwohl es zumindest in der augenblicklichen Version des Handbuchs nicht erwähnt wird, kann man auch das Zahlenfeld rechts auf der Tastatur für Bewegungen im Text verwenden. Diese Tasten ergeben also keine Ziffern und Sonderzeichen am Bildschirm sondern bewegen die Schreibmarke oder den Text.

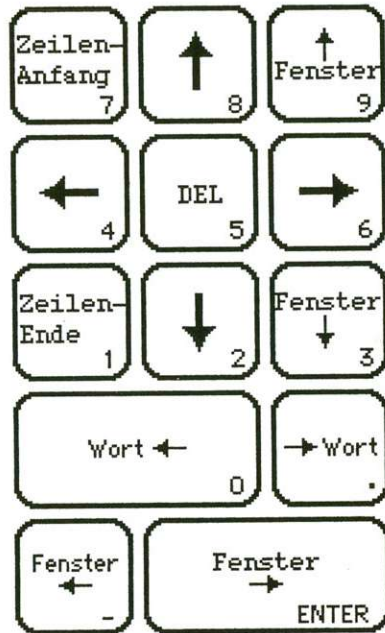


Bild 6 - 18: Funktion des Zahlenfeldes in Textcraft

Experimentieren Sie ein wenig damit herum und Sie werden schnell merken, wie nützlich dieser selten benötigte Teil der Tastatur dadurch auf einmal wird.

6.4 Musicraft

Das Programm Musicraft läßt sich am ehesten als eine Kompositionshilfe und einfacher Synthesizer im Computer beschreiben. Kurze Musikstücke können vierstimmig in musikalischer Notation niedergelegt und dann mit einer Reihe verschiedenster Instrumente gespielt werden, deren Charakteristika mit der Synthesizer-Komponente des Programms manipuliert werden können. Sowohl solche Musikstücke, als auch "Instrumente" können abgespeichert und für die Verwendung in neuen Stücken wieder eingelesen werden.

Musicraft wird aus zwei Gründen nicht ganz so ausführlich beschrieben wie die vorangegangenen Programme. Zum einen lag das Programm bei der Entstehung des Manuskripts noch nicht in einer endgültigen Form vor und

einige Funktionen waren noch nicht realisiert. Zum anderen wird es aber wohl auch viele Besitzer des Amiga geben, die kein Interesse an der Musikerzeugung mit dem Computer haben, vor allem auch deshalb, weil dies einige Vorkenntnisse musikalischer und akustischer Art verlangt, die nicht jeder besitzen dürfte und die nicht mit einem Kapitel aus einem Buch erworben werden können. (Auf dem Bildschirm etwas herumzumalen verlangt da doch deutlich weniger Vorkenntnisse.)

Besonders wegen der mangelnden Endgültigkeit der vorliegenden Version kann es vorkommen, daß einige Eigenschaften des endgültigen Produkts, das der Leser in Händen hält, etwas von den beschriebenen abweichen. Die wesentlichen Grundzüge des Programms werden jedoch gleich bleiben.

6.4.1 Aufbau des Programms

Musicraft besteht aus drei getrennten Komponenten, die auch drei komplett verschiedenen Bildschirmen entsprechen: Einem Kompositionsteil, einer Klaviatur und einem Synthesizer. Zwischen diesen drei Programmteilen kann man mit einem Menü-Befehl hin- und herwechseln. Wie es sich für ein Multitasking-System aber gehört, kann man natürlich in einem Teil des Programms arbeiten, während ein anderer z.B. gerade ein Musikstück spielt. Die Aufteilung des Programms in drei Teile hemmt die Arbeit damit deshalb kaum.

Startet man das Programm, so gelangt man zunächst in den Kompositionsteil, was daran zu sehen ist, daß man ein leeres Notenblatt und eine Menge anderer Hilfsmittel (Gadgets) auf dem Bildschirm sieht.

Wenn man in einen anderen Teil des Programms möchte, so wählt man einen Befehl aus dem ersten Menü **Screens** des Programms. Mit **Keyboard** kommt man zur sogenannten "Tastatur" und mit **Synthesizer** in den Synthesizer. Der jeweils sichtbare Teil des Programms wird im Menü mit einem Haken versehen; zunächst also **Staff** (engl. für *Notenblatt eines Stücks*). (Es dürfte Ihnen aber wohl kaum jemals Schwierigkeiten machen, zu entscheiden, wo Sie sich gerade befinden.) Das Menü **Screens** bleibt in allen Programmteilen, in die Sie damit kommen können, gleich. Alle anderen Menüs werden je nach Programmteil ausgewechselt. Lassen Sie sich davon nicht verwirren; in allen Programmteilen außer dem Kompositionsteil sind es sowieso nur wenige Menü-Befehle, unter denen Sie auswählen können.

Will man Musicraft wieder verlassen, so kann man einfach das Fenster, das gerade zu sehen ist, mit dem Schließ-Gadget schließen. Zumindest die

vorliegende Programmversion 0.8 besitzt keinen Menü-Befehl, mit dem das Programm beendet werden kann.

6.4.2 Der Kompositionsteil

Der Kompositionsteil von Musicraft dient sowohl zum Erstellen eigener Musikstücke als auch zum Abspielen vorhandener. Auf der Musicraft-Diskette befinden sich bereits eine ganze Reihe solcher Musikstücke und in den USA werden sogar bereits Musicraft-Alben, ganze Disketten voller Musik für Musicraft, verkauft. Laden Sie zum Ausprobieren des Programms einfach einmal ein Musikstück mit dem Befehl **Open** aus dem Menü **Scores**. Für Freunde der Pop-Musik befindet sich z.B. ein Auszug des Hits **Axel F.** aus dem Film **Beverly Hills Cop** auf der Diskette, der recht gut gemacht ist.

Unten in der Mitte des Bildschirms befinden sich drei Tasten, auf denen die Bezeichnungen **Play** (engl. für *Spielen*), **Stop** und **Repeat** (engl. für *Wiederholen*) stehen. Nach dem Start des Programms ist zunächst einmal **Stop** aktiv; das Programm wartet. Klicken Sie aber auf **Play**, so wird das zuletzt geöffnete oder von Ihnen eingegebene Musikstück gespielt. Klicken Sie auf **Repeat**, so wird es ständig wiederholt, d.h. nachdem es einmal zu Ende ist, fängt es wieder von vorne an.

Wollen Sie sich die Noten eines längeren Stücks betrachten, die ja nicht alle in das kleine Fenster hineinpassen, so können Sie dazu den Rollbalken verwenden, der das untere Viertel des Bildschirms vom Notenblatt trennt. Wie ein solcher Rollbalken bedient wird, wurde ja bereits mehrfach beschrieben.

Mit Musicraft können recht lange Musikstücke erstellt werden, obwohl auf der Diskette nur ziemlich kurze Beispiele mitgeliefert werden. Experimente haben gezeigt, daß zumindest Stücke von einigen Minuten Dauer kein Problem sind. (Genaue Werte konnten nicht ermittelt werden, da das endgültige Handbuch nicht vorlag.)

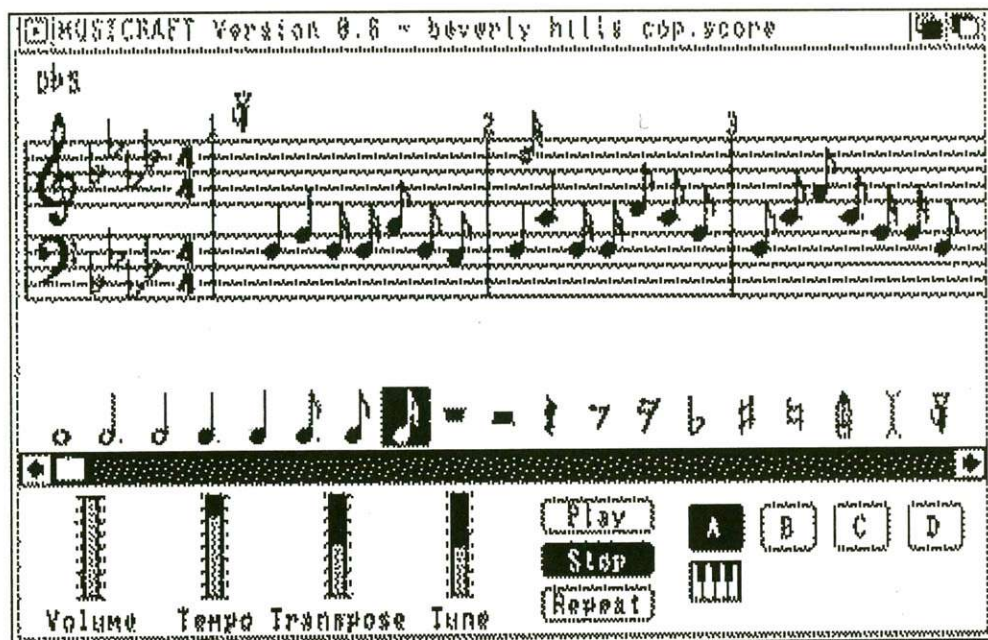


Bild 6 - 18: Das Musicraft-Notenblatt

Musicraft erlaubt das Komponieren und Abspielen von Musikstücken mit maximal vier Stimmen. Sie können diese mit den vier Tasten **A**, **B**, **C** und **D** rechts unten im Fenster, die jeweils einer dieser Stimmen entsprechen, kontrollieren. Diese vier Tasten können insgesamt drei verschiedene Zustände besitzen. Ist eine Taste schwarz, so wird die entsprechende Stimme beim Abspielen des Stücks auch berücksichtigt und kann auch editiert werden (s.u.). Ist eine Taste blau, so wird die Stimme zwar gespielt, kann aber nicht verändert werden. Ist eine Taste weiß, so wird die entsprechende Stimme nicht gespielt, kann nicht verändert werden und wird im Notenblatt auch nicht angezeigt. Die Noten einer Stimme im Notenblatt haben auch immer dieselbe Farbe wie die entsprechende Taste (weiße Noten sind also nicht sichtbar).

Man kann zwar alle vier Stimmen gleichzeitig hören, aber immer nur eine Stimme eines Stücks verändern, was bedeutet, daß nur eine Taste schwarz sein darf, wenn man ein Stück editieren will. Sind zwei oder mehr Tasten

schwarz, so ignoriert das Programm die im folgenden beschriebenen Operationen.

Wenn Sie ein klein wenig von Musik verstehen, sollten Ihnen fast alle Symbole unterhalb des Notenblatts geläufig sein. Mit diesen können Sie ein Stück ändern oder ein neues eingeben. Wenn Sie z.B. eine neue Note eingeben wollen, so brauchen Sie einfach nur auf ein Notensymbol der entsprechenden Länge zu klicken und schon wird der Maus-Zeiger zur Note. Wenn Sie nun auf die Notenlinien zwischen zwei Noten klicken, wird eine solche Note zwischen die schon vorhandenen eingefügt. Die Taktstriche werden dabei automatisch verschoben.

Mit dem Bleistift-Symbol können Sie Noten auch wieder aus einem Stück entfernen. Klicken Sie dazu auf den Bleistift und dann auf die Note, die verschwinden soll. Sie wird dadurch gelöscht, und die rechts davon liegenden Noten rücken automatisch nach links auf.

Das Symbol neben dem Bleistift, das aussieht wie ein in die Länge gezogenes X, dient zum Auswählen größerer Bereiche aus einem Stück, die dann mit den Befehlen aus dem **Edit**-Menü manipuliert werden können. Dieses war in der vorliegenden Version noch nicht realisiert, ermöglicht aber offensichtlich das Löschen, Duplizieren und Wiederholen des selektierten Bereichs und ein Verschieben um einen Halbschritt oder eine Oktave nach oben und unten.

Das letzte Symbol, eine kleine Trompete, dient zum Wechseln des Instruments. Klicken Sie dazu auf die Trompete und dann vor die Note im Notenblatt, die als erste mit dem neuen Instrument gespielt werden soll. Daraufhin erscheint der bereits bekannte Open-Requester, in dem Sie ein neues Instrument öffnen können. Zur Markierung des Instrumenten-Wechsels im Stück erscheint danach oben über den Notenlinien die kleine Trompete. Von dieser Stelle an bis zur nächsten Trompete wird die gerade editierte Stimme dann mit diesem Instrument gespielt.

Während Sie am Komponieren bzw. Ändern einer Komposition sind, können Sie übrigens auch jederzeit mit Melodien herumspielen und sie "antesten". Hierzu drücken Sie entweder kurz auf **Play** oder "klimpern" auch ein wenig direkt auf der Tastatur. Welche Taste, welchen Ton ergibt, können Sie ausprobieren oder es sich im Programmteil **Keyboard** (*Tastatur*) anschauen oder auch ändern (s.u.).

Tonart und Takt, in dem ein Stück gespielt werden soll, kann man mit dem Befehl **Signatures** aus dem Menü **Status** festlegen bzw. ändern. Hier zeigt sich der Vorteil eines Kompositions-Programms gegenüber einem echten Notenblatt. Nachdem sie ein Stück komponiert haben, können Sie den Takt auch nachträglich noch ändern oder es testweise in einer anderen Tonart

spielen lassen. (Wie sinnvoll solche Änderungen sind, ist allerdings eine andere Frage.)

Wollen Sie das Programm verlassen und später an einem Stück weiterarbeiten, so können Sie es mit den Befehlen **Save** oder **Save As** aus dem Menü **Scores** in der aus anderen Programmen inzwischen ja bekannten Weise abspeichern. Haben Sie stattdessen aber einige Änderungen an einem Stück durchgeführt und merken, daß diese überhaupt nicht klingen, so können Sie mit dem Befehl **Revert** zur letzten gesicherten Version zurückkehren oder mit **New** ein neues leeres Notenblatt vorgelegt bekommen. Sowohl bei **New** wie auch bei **Revert** verlieren Sie alle noch nicht gesicherten Änderungen!

6.4.3 Die Tastatur

Der Programmteil von Musicraft, der zweifellos am einfachsten zu beschreiben ist, ist die Tastatur. Sie erscheint, wenn der Befehl **Keyboard** (engl. für *Tastatur*) aus dem Menü **Screens** aufgerufen wird.

Die oberen drei Viertel des nun erscheinenden Bildschirms werden von einer Skizze der Amiga-Tastatur ausgefüllt. Im unteren Viertel ist eine Klavier-Tastatur zu sehen. Auf der oberen Tastatur ist auf jeder Taste in musikalischer Notation verzeichnet, welche Note erklingt, wenn diese Taste gedrückt wird. Tasten, die keinen Ton erzeugen, bleiben weiß.

Die Standard-Belegung der Tastatur ist bereits ganz sinnvoll angeordnet und gut dafür geeignet, Melodien beidhändig zu spielen. Wem Sie jedoch nicht gefällt, oder wer auch auf den unbelegten Tasten eine Note haben möchte, der kann diese Belegung jederzeit ändern. Hierzu wird die zu ändernde Taste einfach mit der Maus angeklickt, worauf diese invertiert wird. Klickt man nun unten in der Klavier-Tastatur auf eine Taste, so erscheint die entsprechende Note auf der so selektierten Taste oben.¹⁸ Auf diese Weise kann man sich die musikalische Tastatur ganz seinen eigenen Bedürfnissen anpassen.

¹⁸ Es ist bedauerlich, daß die deutsche Sprache keine unterschiedlichen Worte für Instrumenten-Tastaturen oder -Tasten und Computer- oder Schreibmaschinen-Tastaturen bzw. -Tasten kennt. Erörterungen, in denen beide Arten von Tastaturen bzw. Tasten vorkommen, sind so zwangsläufig etwas verwirrend.

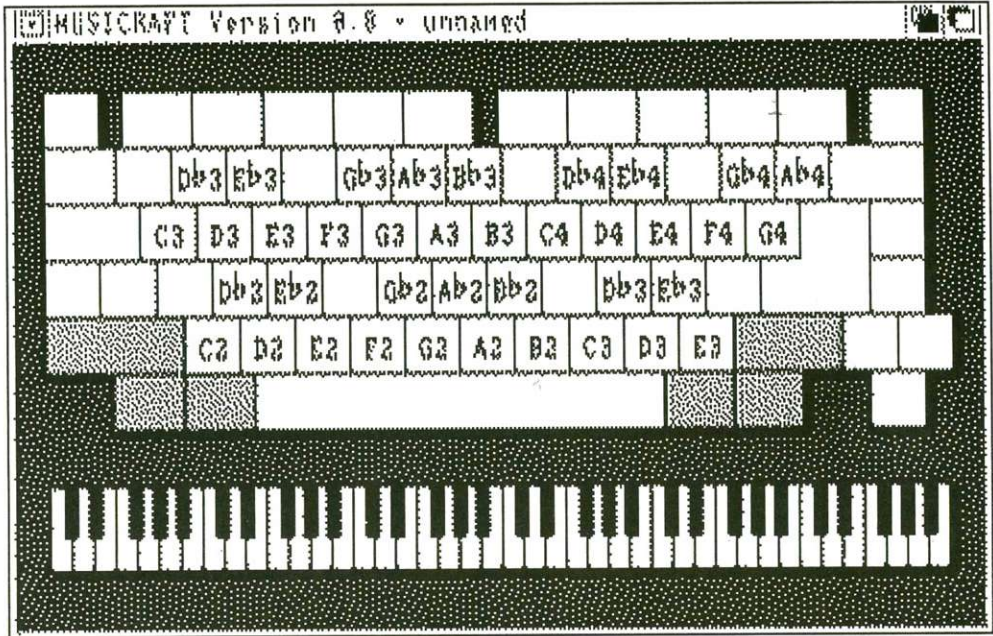


Bild 6 - 19: Die Musicraft-Tastatur

Nachdem man solche Änderungen durchgeführt hat, möchte man natürlich sofort wissen, wie das Spielen auf der neuen Tastatur ist. Hierzu kann man mit dem einzigen Menü-Befehl *Open* im Menü *Instruments* ein Instrument von der Diskette einlesen und es sofort ausprobieren. Wer will, kann so jederzeit auch Violine und Trompete auf seiner Tastatur spielen.

6.4.4 Der Synthesizer

Wesentlich komplexer als die Tastatur ist da schon der Synthesizer, der sichtbar wird, wenn der gleichnamige Befehl aus dem Menü *Screens* ausgewählt wird. Wer sich bereits mit elektronischer Musik auskennt, wird sich aber zweifellos hier zuhause fühlen.

Der Synthesizer von Musicraft ist nur einstimmig, weshalb er für professionelle Anwendungen kaum geeignet sein dürfte. Er ist vor allem für

ein Herumspielen mit den musikalischen Möglichkeiten des Amiga und zur nachträglichen Manipulation der Eigenschaften von Instrumenten gedacht. Je nachdem, ob ein Instrument geladen ist oder nicht, erscheinen auch zwei verschiedene Bilder im Synthesizer-Fenster.

Wenn ein Instrument geladen wurde (entweder in diesem Programmteil oder in **Keyboard**), so kann nur noch die *Hüllkurve* einer Note und das *Vibrato* geändert werden.¹⁹ Ist kein Instrument geladen, so bekommt man ein komplettes Synthesizer-Panel zu sehen, in dem ein Oszillator und ein Modulator (LFO) zur Verfügung stehen, die nach allen Regeln der "Synthesizer-Kunst" miteinander kombiniert werden können.

Der Modulator kann z.B. zur Änderung der Frequenz oder der Lautstärke des Oszillators eingesetzt werden, oder auch als Filter. Wenn man davon absieht, daß nur eine Stimme (ein Kanal) zur Klangerzeugung zur Verfügung steht, wird Musicraft durch diese Möglichkeiten fast schon zu einem vollwertigen Synthesizer.

¹⁹ Wem diese Begriffe nichts sagen, der kann zumindest die Bedeutung der *Hüllkurve* im Kapitel *Klangerzeugung* nachlesen. Prinzipiell gilt aber für den Synthesizer-Teil von Musicraft am stärksten, daß einige Vorkenntnisse verlangt werden.

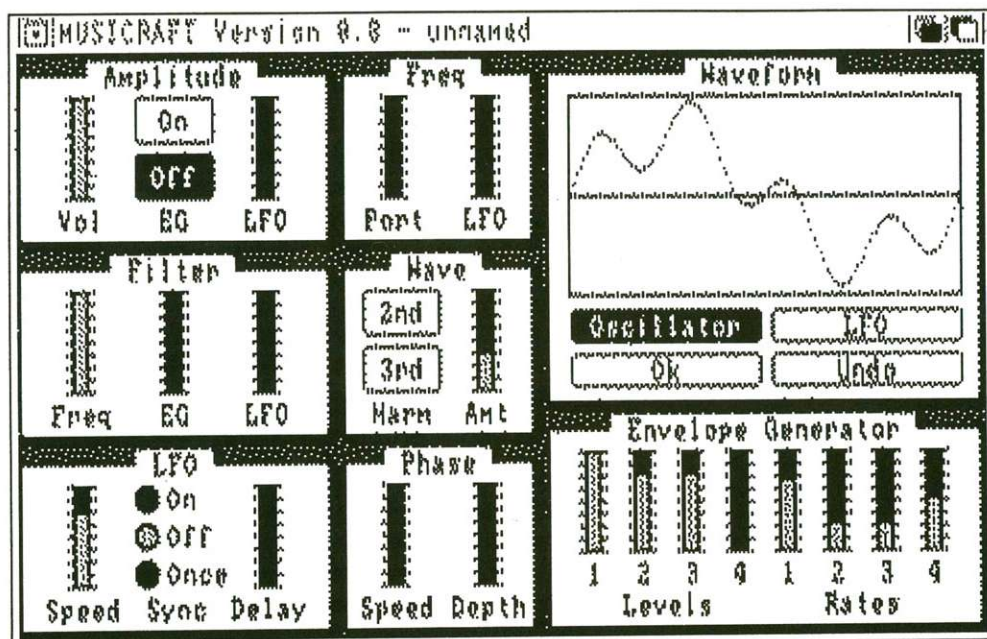


Bild 6 - 20: Der Musicraft-Synthesizer

Wir wollen uns in diesem Buch aber nur mit der einfachsten Synthesizerform intensiver beschäftigen, da die vollständige Version doch mehr etwas für die Profis der elektronischen Musik ist.

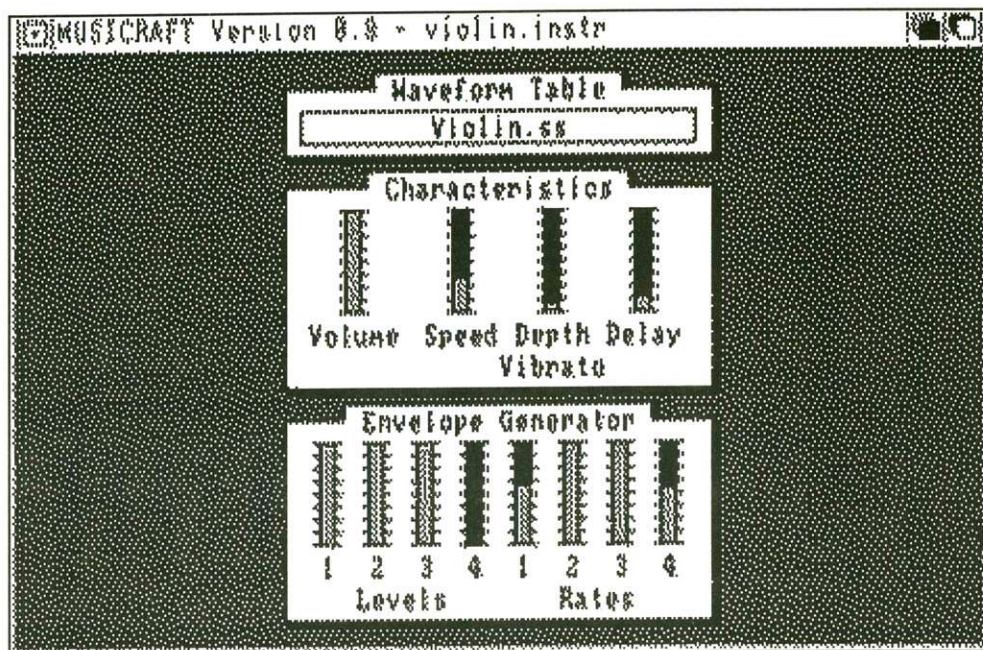


Bild 6 - 21: Ändern der Charakteristika eines Instruments

Die Hüllkurve einer Note setzt sich aus vier Phasen zusammen (die im Kapitel Klangerzeugung ausführlich erläutert werden). Zunächst steigt die Lautstärke des Tons an (*Attack-Phase*), fällt dann etwas ab (*Decay-Phase*), bleibt einen Moment auf einer Art Plateau stehen (*Sustain-Phase*) und klingt dann aus (*Release-Phase*). Diese vier Phasen sind jeweils durch 2 Werte charakterisiert: einer, der angibt, wie laut die Note am Ende der Phase ist und einer, der angibt, wie lange die Phase dauert. Höhe und Länge dieser Zahlen sind bei unterschiedlichen Instrumenten recht verschieden.

Im untersten Kästchen des Synthesizers (*Envelope Generator*) können diese Werte eingestellt werden. Links befinden sich vier Regler für die Höhe bzw. Lautstärke (**Levels**) der vier Phasen und rechts daneben vier Regler für die Dauer (**Rates**) der Phasen. Falls Ihnen das Wort *Hüllkurve* nichts sagt, probieren Sie einfach ein wenig mit diesen Reglern herum und drücken zwischendurch ab und zu einmal auf eine Taste. Sie werden schnell merken, wie diese Einstellungen auf den typischen Klang einer Note wirken.

Im Kästchen darüber (*Characteristics*) können Sie die Lautstärke (*Volume*) des Programms und die Eigenschaften des Vibratos beim ausgewählten Instrument ändern. Letzteres wird durch drei Schieberegler erledigt, die es ermöglichen, Geschwindigkeit (*Speed*), Stärke oder Tiefe (*Depth*) des Vibratos und die Pause (*Delay*), bis er nach Anschlag einer Note einsetzt, zu verstellen.

6.5 Ausblick

Mit dem Programm Musicraft soll dieser kleine Überblick über schon vorhandene Programme für den Amiga erst einmal abgeschlossen werden. Selbst wenn für Sie noch nichts dabei war, trösten Sie sich. Es gibt bereits viele andere Programme für den Amiga, die wir aber nicht rechtzeitig zur Fertigstellung des Buches über "den großen Teich" schaffen konnten. Auch in der Planung und Entwicklung befinden sich noch eine ganze Reihe von Programmen, die die Fähigkeiten des Amiga (hoffentlich) voll ausnutzen werden.

Wie jeder Computer hat natürlich auch der Amiga das Problem, schnell genug "reichlich" Software zu bekommen. Ohne Programme ist ein Computer schließlich zu nichts zu gebrauchen. Bei einem neuen Computer mit völlig neuen Möglichkeiten wird dieses Problem besonders dadurch erschwert, daß sich die Programmierer erst einmal mit diesen Neuheiten vertraut machen müssen. Die Vielfalt bei der Programmauswahl wird deshalb in der ersten Zeit wohl auch beim Amiga etwas vermißt werden. Dies wird aber mit Sicherheit nicht allzu lange anhalten. Im folgenden finden Sie eine Liste der Programme, an denen zur Zeit gearbeitet wird, oder die bereits (zumindest in den USA) erhältlich sind.

Deutsche Programme sind darunter leider noch nicht zu finden. Die deutschen Softwarehäuser hatten den Amiga noch nicht lange genug zur Verfügung, als dieses Buch entstand, um sich ausreichend damit vertraut zu machen. Z.B. eine auf deutsche Verhältnisse zugeschnittene Buchhaltung wird deshalb wohl noch bis Sommer oder Herbst 1986 nicht zu haben sein. Bei vielen anderen Programmen spielt es aber keine sehr große Rolle, ob sie nun in den USA oder hier bei uns entstanden sind, solange die Bedienung "eingedeutscht" wird.

Nun aber die versprochene Liste der Amiga-Programme (Die bereits beschriebenen Programme sind natürlich nicht noch einmal aufgeführt

worden. Auch Programmiersprachen sind in dieser Liste nicht zu finden; sie werden in einem späteren Kapitel angeschnitten.):

AmigaDraw (von der Firma *Aegis Development, Inc.*)

Ein Zeichenprogramm, das im Gegensatz zu den Mal-Programmen *Graphicraft* und *Deluxe Paint* auch für technische Zeichnungen und Skizzen geeignet ist. Zeichnungen können zudem wesentlich größer werden als der Bildschirm.

Deluxe Music Construction Set (von *Electronic Arts*)

Ein Kompositions- und Musikprogramm, ähnlich wie *Musicraft* aber mit wesentlich mehr Möglichkeiten. Über eine zusätzliche MIDI-Schnittstelle kann Musik mit bis zu acht Stimmen (Kanälen) komponiert und ausgegeben werden.

Deluxe Video Construction Set (von *Electronic Arts*)

Ein Programm, mit dem man Bilder aus *Deluxe Paint* und Sound-Effekte aus *Deluxe Music Construction Set* zu bewegten Filmsequenzen mit Ton-Untermalung zusammensetzen kann. Mit einem angeschlossenen Videorecorder können diese Szenen aufgenommen und jederzeit (z.B. bei geschäftlichen Präsentationen) wiedergegeben werden.

Enable (von der Firma *The Software Group*)

Ein integriertes Programmpaket, dessen Teilprogramme im folgenden beschrieben werden. Die Teilprogramme können problemlos Daten austauschen und eng miteinander zusammenarbeiten.

Enable/File (von der Firma *The Software Group*)

Ein Datenbankprogramm, das Teil des integrierten Programmpakets *Enable* (s.o.) ist.

Enable/The Office Manager (von der Firma *The Software Group*)

Ein Organisations- und Planungsprogramm, das Teil des integrierten Programmpakets *Enable* (s.o.) ist.

Enable/Write (von der Firma *The Software Group*)

Ein Textverarbeitungsprogramm, das Teil des integrierten Programmpakets *Enable* (s.o.) ist. *Write* besitzt wesentlich mehr Funktionen als *Textcraft* und ist auch für den professionellen Einsatz und größere Dokumente geeignet.

Financial Cookbook (von *Electronic Arts*)

Ein Programm, mit dem Sie Ihre Haushaltsbuchführung erleichtern, finanzielle Kalkulationen durchführen und unterschiedliche Alternativen für verschiedenen Ausgaben durchrechnen können.

ScoreWriter (von *Cherry Lane Technologies*)

Ein Kompositions- und Musikprogramm, ähnlich wie Musicraft.

Spiele (von *Electronic Arts*)

Die Firma Electronic Arts hat bereits eine ganze Reihe von Spielen für den Amiga fertiggestellt. Darunter sind sowohl klassische Videospiele, als auch Denkspiele und Computer-Abenteuer. Auch der Erholung nach einem langen Arbeitstag steht so nichts mehr im Wege.

Telecraft (von der Firma *Software 66*)

Ein Kommunikationsprogramm, mit dem man in Verbindung mit einem *MODEM*-Zusatzgerät Datenbanken und Mailboxen benutzen und über eine sogenannte *Terminal-Emulation* auch mit Großcomputern kommunizieren kann.

7

Grundlagen des CLI

7.1 Was ist das CLI ?

Bis jetzt haben wir uns in diesem Buch nur mit einer Möglichkeit, den Amiga zu bedienen, beschäftigt: der Workbench. Die Workbench erlaubt es, unter Nutzung der Benutzerschnittstelle Intuition, die Verwaltungsaufgaben, die bei der Benutzung eines Computers anfallen, auf sehr komfortable Weise zu erledigen. Der Inhalt der Computerspeicher, vor allem der Disketten und Festplatten, wird dabei in grafischer Form verdeutlicht. Alle manipulierten Objekte werden in Form kleiner Bilder, der *Icons* dargestellt und die wichtigsten Operationen sind oft durch ein, zwei Mausklicks und Mausbewegungen zu erledigen. Diese freundliche, grafische Darstellung der komplexen Vorgänge unterscheidet den Amiga von vielen anderen Computern, bei denen man die Befehle in Form schwer zu behaltender Kommandos über die Tastatur eingibt und die Ergebnisse dann oft als lange Textlisten zu sehen bekommt.

Der Amiga besitzt aber auch noch eine andere Methode der Bedienung (eine andere "Benutzerschnittstelle", wie der Fachmann sagt): das sogenannte *CLI* kommt der "üblichen" Bedienung eines Computers, wie man es von Computer-Modellen anderer Hersteller kennt, recht nahe. *CLI* steht für *Command Line Interpreter* oder *Command Line Interface* (zu deutsch etwa "Kommandozeilen-Ausführer" oder "Kommandozeilen-Schnittstelle"); die Amiga-Dokumentation (und auch die Belegschaft von Commodore Deutschland) ist sich über den genauen Ursprung der Abkürzung etwas

uneins. Ich werde im folgenden immer nur die Abkürzung CLI verwenden — sie ist einfach wesentlich kürzer als einer der beiden kompletten Namen. Die Kommunikation über diese Schnittstelle spielt sich über die Tastatur und über ein Fenster ab, das sich verhält, als wäre es ein Computer-Terminal, das nur Text darstellen kann. Die Grafikmöglichkeiten des Amiga liegen hier völlig brach!

"Wozu dies alles? Wozu die vielen Möglichkeiten des Amiga ungenutzt lassen?" werden Sie jetzt vielleicht fragen. Aber auch diese zweite Art, den Amiga zu bedienen, hat ihre Berechtigung, wie wir in den folgenden Kapiteln noch sehen werden. Die Workbench hat einige Einschränkungen und Umständlichkeiten, die Ihnen jetzt wahrscheinlich noch nicht auffallen werden. Vielleicht werden Sie diese Einschränkungen sogar nie bemerken oder nicht als solche empfinden. Bei einem täglichen Gebrauch und bei speziellen Anwendungen, wie z.B. der Programmierung, können sie aber bald lästig werden. Deshalb gibt es das CLI.

Das CLI ist ein Programm (fast) wie jedes andere. Computer-Wissenschaftler würden es eine *Shell* (zu deutsch "Schale") nennen. Startet man es, so legt es sich wie eine Schale (daher der Name) um den Betriebssystemkern des Amiga. Dieser Kern ist ein recht komplexes Software-Gebilde, mit dem zu kommunizieren den wenigsten Anwendern leicht fallen dürfte. Für alle, die nicht in der binären Computersprache denken, ist eine Shell da. Die Befehle des Benutzers werden von einer solchen Shell aus ihrer menschenverständlichen und einigermaßen freundlichen Form in die Sprache überführt, die der Computer versteht. Genauso werden die Ergebnisse der Benutzerbefehle, die vom Betriebssystemkern kommen, zunächst von der Shell empfangen und in eine Sprache übersetzt, die der menschliche Anwender verstehen kann. Bei der Workbench ist dies eine bildliche Sprache, beim CLI eine etwas schwierigere, textuelle Sprache. Aber auch sie ist erlernbar — leichter als die meisten menschlichen Sprachen.

Prinzipiell haben die meisten Computer eine "bevorzugte" Shell, mit der ein Anwender "normalerweise" zu tun hat. Beim Amiga ist es die Workbench. Bei vielen Computern ist es aber auch problemlos möglich, sie durch ein anderes Programm zu ersetzen oder zu ergänzen. Das bekannte Computer-Betriebssystem UNIX™, von dem Sie vielleicht schon gehört haben, aber auch der Amiga nutzen diese Möglichkeit. Der Anwender kann stets die Shell wählen, die ihm am besten gefällt oder die für eine bestimmte Aufgabe am besten geeignet scheint.

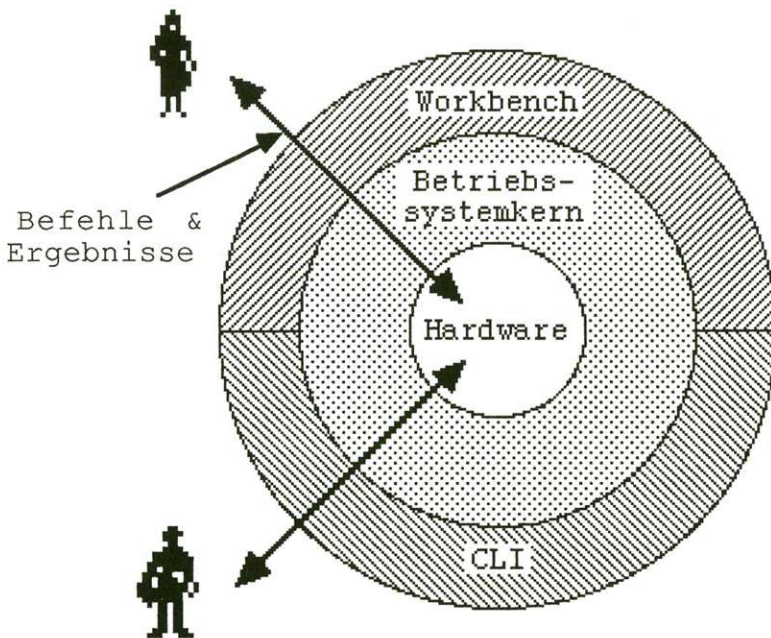


Bild 7 - 1: *Es kann mehrere Programme geben, die einem helfen, einen Computer zu bedienen*

Das CLI ist mehr für den fortgeschrittenen Benutzer gedacht, der schon eine Weile mit dem Amiga gearbeitet hat. Wer gerade erst seinen Amiga gekauft hat und die verschiedenen Optionen der Workbench schon verwirrend genug findet, der sollte das Studium dieses Buchteils vielleicht auf einen späteren Zeitpunkt verschieben.

7.2 Das Einschalten des CLIs

7.2.1 Vorarbeiten

Bevor man mit den ersten Versuchen mit dem CLI beginnt, sollte man sich zunächst am besten eine zweite Kopie der originalen Workbench-Diskette, die

dem Amiga beigelegt ist, auf eine neue leere Diskette machen. Wie dies innerhalb der Workbench gemacht wird, wird im zweiten Kapitel dieses Buches beschrieben. Wenn die Kopie fertig ist, beschriftet man sie sinnigerweise mit dem Namen "CLI-Diskette" und gibt ihr auch in der Workbench den Namen "CLI-Diskette" (mit dem Befehl *Rename* im Menü *Workbench*). Diese Diskette werden wir für den Rest dieses Buchteils für unsere Experimente verwenden. Die erste Kopie der Workbench-Diskette kann man solange erst einmal beiseite legen. Sind Sie mit dem Kopieren fertig, so nehmen Sie bitte beide Disketten aus den Laufwerken und starten den Amiga neu (mit der Tastenkombination <CTRL>+<A>+<A>). Schieben Sie dann die neu kopierte Diskette ins interne Laufwerk. Für den Rest dieses und den größten Teil der folgenden Kapitel nehmen wird an — sofern nicht ausdrücklich etwas anderes erwähnt wird — **daß Sie nur mit dieser Diskette arbeiten**. Sind sie stolzer Besitzer eines zweiten Laufwerks oder einer Festplatte (beides eine sehr sinnvolle Anschaffung), so bleibt beides vorläufig unberücksichtigt. Nun aber genug der Vorreden. Kommen wir zum CLI selbst.

7.2.2 Sichtbarmachen des CLI

Zunächst ist es gar nicht so einfach, das CLI zu aktivieren. Nach dem Einschalten des Amiga und dem Einschieben der Workbench-Diskette erscheint ja sofort die Workbench und ein "CLI" ist weit und breit nicht zu sehen. Wer sich aber einmal das Hauptfenster des Programms *Preferences* genau angesehen hat, der wird einen Schalter namens "CLI" in der unteren Hälfte am rechten Rand dieses Bildschirms entdecken. Er befindet sich direkt unter dem Schalter "Text", mit dem man zwischen 60- und 80-spaltigem Text innerhalb der Workbench wählen kann. Der Schalter CLI steht normalerweise auf *Off* (*Off* ist Orange und *On* Blau). Klicken Sie nun auf *On*, das daraufhin blau werden sollte, und verlassen Sie dann Preferences mit dem Knopf *SAVE*. Wenn Sie zurück in der Workbench sind, also wieder Disketten- und Schubladen-Icons erscheinen und die roten Lämpchen an beiden Diskettenlaufwerken erloschen sind, starten Sie bitte ihren Amiga noch einmal neu (wieder mit der Tastenkombination <CTRL>+<A>+<A>).¹

Sobald Sie jetzt wieder zurück in der Workbench sind, öffnen Sie das Disketten-Icon (das jetzt "CLI-Diskette" heißen sollte, wenn Sie sich eine

¹ Dieser Neustart sollte eigentlich nicht unbedingt nötig sein, und es kann sein, daß Sie auch ohne ihn auskommen. Wenn Sie experimentierfreudig sind, können Sie auch ohne Neustart fortfahren.

Kopie ihrer Workbench-Diskette angelegt haben, wie oben beschrieben). Nachdem das Fenster der Diskette sich geöffnet hat, öffnen Sie nun bitte die Schublade **System**. Bei einer nicht modifizierten Originaldiskette ist es die dritte Schublade von oben. Im Fenster, das sich nun öffnet, sollte ein Icon namens "CLI" erscheinen.² Dies ist das Programm CLI, die zweite Benutzerschnittstelle des Amiga, mit der wir uns im Rest dieses Buchteils befassen werden.

7.2.3 Starten des CLI und erste Schritte

Starten Sie nun CLI (indem Sie es öffnen) und ein neues Fenster erscheint auf dem Bildschirm. Oben links in diesem Fenster erscheint der Text

1>

Dies ist der sogenannte "Prompt" des CLI, der immer dann erscheint, wenn das CLI bereit ist, einen neuen Befehl von Ihnen entgegenzunehmen und auszuführen.

Spätestens jetzt wird es Zeit für Sie, die Tastatur unter der Systemeinheit Ihres Amiga hervorzuholen, falls Sie dies nicht schon längst getan haben. Denn innerhalb des CLI-Fensters zählt nur die Tastatur und die Maus hat keine Funktion mehr. Sie können mit ihr zwar noch wie gewohnt die Position und Größe des Fensters ändern, ansonsten befinden Sie sich nun aber in einer Umgebung, die in gewisser Hinsicht vorsintflutlich ist und in der die Maus nicht zählt. Benutzer fast aller anderen Computer-Modelle müssen sich allerdings fast immer so ähnlich mit ihren Computern herumschlagen, deswegen sollten Sie sich nicht entmutigen lassen. Manches ist zwar lästig aber dafür sind mit dieser Art der Computerbedienung andere Dinge auch wieder einfacher.

Tippen Sie nun Ihren ersten Befehl an das CLI ein:

1>**dir**

Während Sie auf der Tastatur die Buchstaben "d", "i" und "r" tippen, erscheinen diese auf dem Bildschirm und die Schreibmarke wandert weiter

² Sehen Sie kein Icon namens **CLI**, starten Sie bitte noch einmal das Programm **Preferences** und vergewissern Sie sich, daß der Schalter **CLI** auf **On** steht. Danach führen Sie noch einmal einen Neustart durch.

nach rechts.³ Das ganze ist dem Eintippen eines neuen Namens für ein Icon in der Workbench (nach dem Befehl **Rename**) sehr ähnlich. Mit der <Backspace>-Taste können Sie, falls Sie sich vertippt haben, den letzten Buchstaben löschen — andere Editiermöglichkeiten gibt es aber nicht. Auch die Pfeil-Tasten und die -Taste können — im Gegensatz z.B. zu **Rename** — nicht verwendet werden. (Es handelt sich **nicht** um ein Text-Gadget.)

Tippen Sie nun <RETURN> zum Abschluß ihrer Eingabe. Das CLI liest den von Ihnen eingegebenen Befehl und versucht ihn auszuführen. In diesem Fall war es der Befehl **dir**, der eine Liste von Datei- und Schubladen-Namen der internen Diskette ausgeben sollte. Haben Sie den Befehl richtig getippt, so "rollt" diese Liste nun vor ihren Augen vorbei — manchmal ein, manchmal zwei Namen in einer Zeile. Währenddessen läuft das Diskettenlaufwerk andauernd und die rote Lampe flackert, denn das CLI muß die Namen der Dateien natürlich von der Diskette lesen. Ist die Liste am unteren Rand des Fensters angekommen, bewegt sich der Fensterinhalt um eine Zeile nach oben (die obere Zeile verschwindet dabei). Die nächsten Namen tauchen dann in der untersten, jetzt leeren, Zeile auf.⁴ Ist die Liste komplett, so erscheint wieder der Prompt:

1>

Manche der Namen, die in der Liste erscheinen, werden Ihnen bekannt vorkommen. Es sind dieselben Namen, die auch einige Icons im Diskettenfenster auf der Workbench tragen. Viele Namen aber werden neu sein, und das zeigt Ihnen schon einen wesentlichen Unterschied zwischen der Workbench und dem CLI. Im CLI können Sie **alle** Dateien und Schubladen⁵ sehen, die auf der Diskette liegen. In der Workbench sehen Sie nur die, von denen die Entwickler des Amiga meinen, daß Sie sie sehen sollten. Auch die Datei (das Programm) CLI war schon immer im Ordner **System** vorhanden,

³ Erscheint das "d" nicht sofort im Fenster, nachdem Sie es gedrückt haben, klicken Sie bitte kurz mit der Maus ins Innere des Fensters und versuchen es noch einmal. Erscheint dann immer noch nichts, so dürfte Ihre Tastatur defekt sein.

⁴ Diesen Vorgang nennt man "Rollen" oder auch "Scrollen" des Fensters. Sie haben ihn ja schon bei den Workbench-Fenstern kennengelernt. Anders als bei diesen Fenstern ist alles, was über den oberen Fensterrand hinaus verschwunden ist, aber unwiederbringlich verloren. Sie können nicht zurückrollen, wie in der Workbench. .

⁵ Das CLI nennt Schubladen *Directories* bzw. *Dateiverzeichnisse*; mehr dazu aber im nächsten Kapitel.

sichtbar wird es erst in dem Moment, wenn der entsprechende Schalter in den **Preferences** umgelegt wird.

Nun noch ein paar Worte zur Form, bevor wir mit der Beschreibung des CLI fortfahren. Wie auch jetzt schon dreimal, werden häufiger Beschreibungen von Eingaben auftauchen, die sie im CLI-Fenster machen sollen und natürlich auch den zugehörigen Ausgaben des CLIs. Um nicht unnötig Verwirrung zu stiften "für welche Texte, wer zuständig ist", werden alle Inhalte des CLI-Fensters in diesem Buch in einem anderen Schriftbild erscheinen. Im Gegensatz zu diesem Text, den sie gerade lesen, wird dieses Schriftbild dem einer Schreibmaschine ähnlich sehen. Handelt es sich bei einem solchen Text um eine Ausgabe des CLIs, erscheint er in normalem Schriftstil (z.B. "1>"), handelt es sich um Eingaben, die Sie auf der Tastatur machen sollen, wird er fett gedruckt (z.B. "**dir**").

Sie haben nun Ihre ersten Schritte im CLI getan, haben es gestartet und den **dir**-Befehl kennengelernt. Bevor Sie weitere Befehle des CLI kennenlernen und ausprobieren können, müssen wir im nächsten Kapitel zunächst einen kleinen Exkurs in die Struktur einer Diskette bzw. einer Festplatte oder eines anderen Massenspeichermediums machen, "wie das CLI es sieht". Es geht dabei vor allem Dateien und Dateiverzeichnisse, zwei Begriffe die Sie bestimmt schon einmal gehört haben, wenn Sie in der Vergangenheit schon mit anderen Computern zu tun hatten.

7.2.4 Verlassen des CLI

Wenn Sie zwischendurch auch einmal wieder in die Workbench wollen oder das CLI endgültig verlassen wollen, bevor Sie das folgende Kapitel durcharbeiten, haben Sie mehrere Möglichkeiten:

Wenn Sie nur vorübergehend etwas in der Workbench machen wollen, und danach wieder zurück ins CLI wollen, ist es nicht nötig, dieses endgültig zu verlassen. Der Amiga besitzt ja ein Multitasking-Betriebssystem, was in diesem Fall bedeutet, daß beim Start des CLI die Workbench nicht beendet wurde. Die Workbench ist nach wie vor — neben dem CLI — aktiv. Klicken Sie einfach mit dem linken Mausknopf in eins der Fenster der Workbench (dies sind z.B. Fenster, in denen der Inhalt einer Diskette oder einer Schublade gezeigt wird). Sind die Fenster der Workbench nicht zu sehen, weil das CLI-Fenster sie alle verdeckt, so können Sie das CLI-Fenster natürlich auch wie jedes andere Fenster mit dem entsprechenden Gadget nach hinten legen.

Sie werden, sobald Sie "zurück in der Workbench" sind, bemerken, daß sich der Inhalt der Titelzeile des Bildschirms ändert. Wenn Sie den rechten Mausknopf niederdrücken, müssen nun auch wieder die Menüs der Workbench in der Menüzeile erscheinen. Sie können jetzt beliebige Aktionen in der Workbench durchführen und durch einen einfachen Mausklick wieder in das CLI zurückkehren.⁶

Eine andere Möglichkeit, das CLI zu verlassen, ist etwas permanenter. Sie beendet das *Programm* CLI wirklich und schließt auch das entsprechende Fenster. Falls das CLI-Fenster im Moment nicht aktiv ist, machen Sie es aktiv, indem sie mit der Maus kurz in das Fensterinnere klicken. Tippen sie dann hinter dem Prompt:

```
1>endcli
```

Daraufhin erscheint eine kurze Meldung im Fenster, die aber im allgemeinen zu schnell verschwindet, als daß man sie lesen könnte, und das CLI-Fenster wird geschlossen. Die einzige Möglichkeit, jetzt wieder ins CLI zu gelangen, ist, explizit das Icon **CLI** in der Schublade **System** zu öffnen.⁷ Der Befehl **endcli** ist also recht endgültig und sollte nur angewendet werden, wenn man sich sicher ist, das CLI in nächster Zeit nicht mehr zu benötigen. Er kann aber auch dann sehr sinnvoll sein, wenn der vom CLI benötigte Speicherplatz dringend für ein anderes Programm gebraucht wird. Falls Sie einen Amiga mit nur 256K Speicherausbaubesitz besitzen, ist diese Überlegung naturgemäß besonders wichtig.

7.2.5 Aufrufen von CLI-Befehlen

Wie oben schon am Beispiel **dir** zu sehen war, bestehen CLI-Befehle immer aus einem (meist kurzen) Wort, das Sie eingeben müssen, damit das CLI eine bestimmte Operation ausführt. Welche Befehle Ihnen normalerweise zur Verfügung stehen, erfahren Sie noch in aller Ausführlichkeit im übernächsten

⁶ Für Experimentierfreudige: Eine recht spektakuläre Möglichkeit, zu zeigen, daß Workbench und CLI wirklich unabhängig voneinander und gleichzeitig aktiv sein können ist, während der Ausführung eines CLI-Befehls in die Workbench zu gehen und dort z.B. ein Icon von einem Fenster in ein anderes zu legen. Der Befehl **dir** im CLI ist hierfür recht gut geeignet, da seine Ausführung meist eine ganze Weile dauert.

⁷ Dies gilt natürlich nur, falls das auf diese Weise beendete das einzige war, das im Moment aktiv war. Existieren noch weitere CLI-Fenster, so können Sie natürlich diese verwenden.

Kapitel. (Eine Warnung vorweg: es sind deutlich mehr als auf der Workbench!)

Das CLI hat jedoch gar keinen "festen" Befehlssatz. Wenn Sie ein Befehls - wort eingeben, so sucht es in Wirklichkeit nach einem Programm, das diesen Befehl ausführen kann, lädt dieses in den RAM-Speicher (sofern es gefunden wurde) und startet es. Indem Sie die Befehls-Programme auf Ihrer CLI-Diskette löschen, umbenennen oder neue hinzufügen, ändern Sie also den Befehlssatz des CLIs. Diese Programme werden allerdings nicht auf der ganzen Diskette, sondern nur in ganz bestimmten Dateiverzeichnissen (Schubladen) gesucht (und hoffentlich gefunden).

Im Gegensatz zur Workbench startet das CLI ein solches Befehls-Programm aber normalerweise nicht als parallelen Task, d.h. Sie können nicht gleichzeitig mit einem solchen Befehl arbeiten und dem CLI neue Befehle erteilen, wie es auf der Workbench je üblich ist. Das CLI läuft in Wirklichkeit zwar weiter, es wartet jedoch immer auf die Beendigung eines Befehls, bevor es einen neuen akzeptiert. Das ist aber nicht ganz so schlimm, wie es sich anhört, da die CLI-Befehle fast alle sehr kleine Programme sind, die auch recht schnell mit ihrer Aufgabe fertig sind.

Sollten Sie es aber sehr eilig haben, können Sie auch schon einen neuen Befehl eingeben, bevor der vorhergehende ganz fertig ist. Während Sie dieses Befehl eintippen, macht der gerade arbeitende Befehl eine Pause und fährt erst dann fort, wenn Sie die <RETURN>-Taste drücken. Der neue Befehl wird dann erst einmal gespeichert und sofort ausgeführt, sobald der vorhergehende fertig ist.

Den Effekt, der eintritt, wenn Sie einen neuen Befehl eintippen, solange der vorhergehende noch nicht fertig ist, können Sie auch dazu "mißbrauchen", einen lange andauernden Befehl kurz anzuhalten. Probieren Sie das einmal mit dem *dir*-Befehl. Tippen Sie *dir* und dann <RETURN> . Sobald dann die ersten Zeilen ausgegeben wurden, drücken Sie kurz auf die <Leertaste> und die Ausgabe hält an. Das CLI hält den *dir*-Befehl jetzt an und läßt ihn nicht weiter arbeiten. Sie können nun in aller Ruhe die schon ausgegebenen Zeilen betrachten. Wenn Sie wollen, daß der *dir*-Befehl fortfährt, drücken Sie ein oder zweimal auf die <BACKSPACE>-Taste, um das Leerzeichen, das Sie eingegeben haben, wieder zu löschen. Sobald die Schreibmarke wieder am linken Rand des Fensters angekommen ist, setzt der *dir*-Befehl seine Arbeit fort.

Diese Unterbrechungsmöglichkeit ist besonders bei Befehlen wichtig, die lange Listen von Daten auf dem Bildschirm ausgeben und/oder, wenn Sie Ihr CLI-Fenster sehr klein gemacht haben. (Denken Sie daran: Text der einmal

am oberen Ende des CLI-Fensters verschwunden ist, ist unwiederuflich verloren!)

7.3 CLI-Tasks

Bisher haben wir ja nur gesehen, wie man dem CLI einen Befehl gibt, dann wartet, bis dieser fertig ist und dann einen neuen eintippt. Dies nutzt die Fähigkeiten eines Multitasking-Betriebssystems wie AmigaDOS kaum aus. Es gibt deshalb auch die Möglichkeit, vom CLI aus mehrere Programme parallel zu bearbeiten.

7.3.1 Mehrere CLI-Fenster

Die erste dieser Möglichkeiten ist die Verwendung mehrerer CLIs. Es hält uns ja nichts davon ab, nach dem Öffnen des ersten CLIs zurück in die Workbench zu gehen, um das CLI-Icon noch einmal zu öffnen. Dies ergibt natürlich ein zweites CLI-Fenster, in dem wir Befehle parallel zum ersten erteilen können. Wenn im ersten z.B. ein "länglicheres" Programm läuft (z.B. der Compiler für eine Programmiersprache), kann man nun einfach mit der Maus auf das zweite Fenster klicken und dort andere Befehle eingeben.

Neue CLI-Fenster können aber auch von einem CLI aus eröffnet werden. Hierzu dient der Befehl *newcli* (*neues CLI*).⁸ Wenn ein solches zweites oder drittes CLI erscheint, fällt Ihnen bestimmt auf, daß der Prompt in diesen Fenstern etwas anders aussieht. Statt 1> steht dort nun 2> oder 3>. Normalerweise zeigt ein jedes CLI nämlich vor dem Größer-Zeichen seine "laufende Nummer" an. Diese Nummer wird aufsteigend von 1 bei der Entstehung neuer CLIs "vergeben" und ändert sich auch dann nicht, wenn andere CLIs gelöscht werden. Man kann dadurch immer eindeutig identifizieren, mit welchem CLI man es zu tun hat.

⁸ Wie Sie bereits beim Erzeugen eines neuen CLIs dessen Fenster auf die gewünschte Größe und Position bringen können, erfahren Sie im übernächsten Kapitel bei der Auflistung aller CLI-Befehle.

7.3.2 Hintergrund-Befehle

Eine vielleicht noch praktischere Methode, das Multitasking des Amiga zu nutzen, sind die *Hintergrund-Befehle*. Wenn wir einen Befehl aufrufen wollen, von dem wir genau wissen, daß er zwar lange dauert, aber keine weiteren Tastatureingaben mehr benötigt und nur wenig Bildschirmausgaben macht, so brauchen wir dazu kein neues CLI (Jedes zusätzliche Fenster bedeutet zusätzlichen Speicherbedarf). Wir lassen ihn mit dem Befehl **run** (lauf los) *im Hintergrund* laufen.

Diesen Befehl kann man an den Anfang jeder Befehlszeile schreiben und er führt den darauf folgenden Befehl als separaten Task parallel zum CLI aus. Zur Kommunikation mit Ihnen verwendet aber auch dieses im Hintergrund laufende Programm dasselbe Fenster wie das CLI im *Vordergrund*. Bei Programmen, die beide Ein- und Ausgaben im Fenster machen wollen, kann dies sehr verwirrend wirken. Falls Sie einen Eindruck von dem Chaos haben wollen, der dann am Bildschirm entstehen kann, tippen Sie einmal **run dir** und sofort danach wieder **dir**. Für die Hintergrund-Verarbeitung eignen sich wirklich nur Befehle, die keine weitere Interaktion mit dem Anwender benötigen.

Wenn Sie genau wissen, daß Sie mehrere Befehle hintereinander aufrufen wollen, so können Sie alle hinter dem **run** eingeben. Nachdem Sie den ersten Befehl komplett eingegeben haben, tippen Sie dazu ein '+' ans Ende der Befehlszeile und dann erst <RETURN>. Der erste Befehl wird nun nicht sofort ausgeführt! Das Plus-Zeichen ist ein Signal für das CLI, zu warten, da noch weitere Befehle folgen. Es bewirkt praktisch, daß die erste Zeile "logisch verlängert" wird, also für das CLI gar keine neue Zeile ist. Auf diese Weise können Sie nahezu beliebig viele "wirkliche Zeilen" zu einer "logischen Zeile" verketteten.

Erst wenn Sie <RETURN> drücken, ohne unmittelbar davor ein Plus-Zeichen eingegeben zu haben, beginnt das CLI, die eingegebenen Befehle auszuführen. Einer nach dem anderen wird gesucht, geladen und aufgerufen. Leider fährt das CLI auch dann noch mit der Ausführung der restlichen Befehle fort, wenn "weiter vorne" ein Fehler aufgetreten ist. Dies macht die gleichzeitige Eingabe mehrerer Befehle zu einer etwas riskanten Sache. Nehmen wir einmal an, Sie hängen zwei Befehle auf diese Art hintereinander. Der erste kopiert eine Datei auf eine andere Diskette, der zweite löscht anschließend das Original. Wenn beim Kopieren ein Fehler auftritt, fährt das CLI trotzdem fort und löscht das Original. Sie "stehen dann da" und haben nun weder das Original noch eine Kopie. Falls es das einzige Exemplar einer wichtigen Datei war, kann wochenlange Arbeit verloren sein!

Obwohl die Verkettung bzw. Verlängerung von Befehlszeilen mit dem Plus-Zeichen eine sehr praktische Sache ist, sollten Sie sie sehr vorsichtig anwenden — zumindest dann, wenn Befehle dabei sind, die direkt oder indirekt destruktiv wirken können!

Das selbe gilt sinngemäß aber auch für die Eingabe neuer Befehle, solange der vorhergehende noch nicht beendet ist (s.o.). Auch hierbei kontrolliert das CLI nicht, ob Fehler eintreten, sondern führt den neuen Befehl "blindgläubig" aus.

7.4 Beeinflussung laufender Programme

Gelegentlich kann es nötig werden, in ein laufendes Programm einzugreifen. Eine Möglichkeit solchen Eingreifens, die Veranlassung einer kurzen Programmpause und das darauf folgende Fortsetzen eines Programms, haben wir ja schon kennengelernt. Sie können aber Programme auch vom CLI aus abbrechen. Dies kann z.B. dann sinnvoll sein, wenn Programme oder Programmfolgen "aus dem Ruder laufen", z.B. in Folge eines Bedienungsfehlers Ihrerseits.

Zum Abbrechen von Programmen dienen sogenannte Control-Kombinationen, bei denen Sie die <CTRL>-Taste festhalten und dann eine andere Taste drücken müssen.

So bricht z.B. <CTRL>-C das gerade laufende Programm ab und <CTRL>-D bricht eine Kommandofolge ab, die mit dem *execute*-Befehl gestartet wurde, der in einem folgenden Kapitel noch näher erläutert wird. Es gibt auch noch die Abbruchbefehle <CTRL>-E und <CTRL>-F, die aber sehr selten (von Programmierern) verwendet werden.

Zu diesen Abbruchmöglichkeiten über die Tastatur muß man jedoch sagen, daß kein Programm gezwungen wird, darauf zu reagieren. Alle diese Control-Kombinationen setzen nur ein internes Signal, um das sich ein Programm kümmern kann, aber nicht muß. Gut geschriebene Programme prüfen diese Signale aber regelmäßig und halten sich auch daran.

8

Dateien und Dateiverzeichnisse

Viele der Konzepte, die wir in der Workbench kennengelernt haben, haben im CLI ein etwas anderes Aussehen. Dies wird vielleicht im ersten Augenblick etwas verwirrend sein; bald wird der Leser aber die Zusammenhänge verstehen und sehen, daß es sich wirklich um dieselben Dinge handelt, die nur in einer etwas anderen "Verkleidung" daherkommen.

8.1 Grundlagen

Die wichtigsten Konzepte des CLI sind wohl Dateien, Dateiverzeichnisse und Geräte. Wer schon andere Computer kennt, dem werden diese Begriffe bereits vertraut sein. Wenn Sie bisher aber nur die Workbench des Amiga kennen, soll Ihnen der folgende Abschnitt einen ungefähren Eindruck von diesen Worten in "Computerchinesisch" geben.

8.1.1 Dateien

Dateien haben wir bereits im ersten Teil dieses Buches kennengelernt, der sich mit der Workbench beschäftigte. Programme (Werkzeuge) und Dokumente (Projekte) sind z.B. Dateien. Während eine Bezeichnung wie "Programm" aber schon etwas über die Aufgabe des entsprechenden Objekts sagt, ist "Datei" etwas abstrakter. Eine Datei ist einfach eine Ansammlung von Daten, die zu einem zusammenhängenden Objekt, eben einer Datei,

zusammengefaßt wurden. Was diese Daten sind oder wie sie strukturiert sind, kümmert das Betriebssystem des Amiga, AmigaDOS, dabei nicht. Egal, ob es sich um ein Programm oder um einen Brief an einen Geschäftsfreund handelt, AmigaDOS sieht nur eine Folge von Daten, die auf der Diskette oder einer Festplatte abgespeichert werden müssen.

Die Disketten oder Festplatten des Amiga sind zunächst einmal auch eine Folge von Daten (Bytes), die aus technischen Erwägungen zu größeren *Blöcken* zusammengefaßt werden. Diese Blöcke haben fortlaufende Nummern. Gäbe es keine Dateien, so müßte man jedesmal, wenn man Daten auf einer Diskette speichern will, angeben, in welchem Block die Daten abgelegt werden sollen. Wenn die Daten länger sind als ein Block, so müssen sie sich auf einen zweiten, dritten, vierten, usw. Block erstrecken. In welchen Blöcken diese zusammengehörigen Daten liegen, muß man sich natürlich auch merken.

Das Konzept der Datei enthebt uns (bzw. ein Programm) von diesen Bit- und Byte-Manipulationen und dem "unwichtigen" Verwaltungskram. Für Programme ist eine Datei eine Folge von Bytes (Buchstaben), die einen eindeutigen Namen besitzt. An diese Folge kann man hinten weitere Daten anhängen, wahlweise ein bestimmtes Byte aus der Folge lesen, oder auch modifizieren, usw.. Das Betriebssystem des Amiga, AmigaDOS, sorgt für die Zuordnung dieser Bytes in Blöcke auf der Diskette oder Festplatte; es übernimmt den Verwaltungskram, wie wir auch von einer guten Sekretärin oder einem Sekretär erwarten, daß sie alle einzelnen Blätter, Formulare, etc. aus den verschiedenen Schränken eines Büros hervorsuchen, wenn wir *die Akte Meier* anfordern.

8.1.2 Dateiverzeichnisse

Wenn Dateien einzelnen Akten entsprechen, so sind Dateiverzeichnisse die Aktenordner unseres elektronischen Büros. Die Workbench zeigt die Dateiverzeichnisse dann ja auch als Schubladen — in denen man sich eben Aktenordner vorstellen kann.

Wenn man sehr viele einzelne Akten verwalten muß, ist es aber unpraktisch, sie alle in einem großen Stapel zu verwalten. Bestimmte Akten werden dann sehr schwer zu finden sein und zudem bekommen wir Benennungsprobleme. Wenn wir z.B. auch persönliche Korrespondenz mit dem Kunden Meier führen, so wollen wir manchmal diese persönlichen Briefe erhalten und manchmal die Rechnungen an ihn, wenn wir *die Korrespondenz mit Meier* sehen wollen. Es wäre unpraktisch, dann alles auf den Tisch gelegt zu

bekommen und sich die gewünschten Dinge wieder selbst herauswählen zu müssen.

Haben wir aber mehrere Aktenordner, wie beispielsweise einen, der die persönliche und einen der die geschäftliche Korrespondenz mit Personen enthält, so können wir diese ganz klar trennen. Bevor wir die "Korrespondenz mit Meier" anfordern müssen wir dann aber auch dazusagen, aus welchem Aktenordner sie kommen soll — aus dem Aktenordner "Persönliches" oder "Geschäftliches".

Genauso können Dateien geordnet werden. Statt einfach alle Dateien "nebeneinander" auf eine Diskette zu legen, können wir auf ihr mehrere Dateiverzeichnisse anlegen und die Dateien auf diese verteilen. Es kann danach durchaus zwei Dateien namens "Meier" geben — aber in verschiedenen Dateiverzeichnissen. Beide Dateien können problemlos unterschieden werden, indem man den Namen des Dateiverzeichnisses vor den Dateinamen schreibt, wenn man einen Dateinamen benötigt. Die beiden Teilnamen werden dabei durch einen Schrägstrich getrennt. Liegt eine Datei namens *Meier* in den beiden Dateiverzeichnissen *Geschäftliches* und *Persönliches* so lauten die "vollständigen" Namen der beiden Dateien dementsprechend:

Persönliches/Meier bzw. *Geschäftliches/Meier*.

8.1.3 Das hierarchische Dateisystem

Werden unsere Aktenmengen noch größer, so können die Aktenberge auch trotz Aktenordner unübersichtlich werden. Wieso sollten wir dann nicht auch die Aktenordner wieder gliedern? Genau das ist im CLI auch möglich. Dateiverzeichnisse können in Dateiverzeichnissen liegen und selbst natürlich auch wieder Dateiverzeichnisse enthalten. Dateiverzeichnisse entsprechen also auch in dieser Hinsicht den Schubladen der Workbench. Wie diese alle anderen Arten von Objekten mit Ausnahme von Disketten enthalten können, können auch Dateiverzeichnisse alle Arten von CLI-Objekten enthalten — einschließlich anderer Dateiverzeichnisse.

Ein anderer möglicher Vergleich wäre ein Buch: Wie ein gutes Buch eine vielstufige Gliederung hat, kann auch eine gut aufgeräumte Diskette und vor allem eine ja noch viel größere Festplatte eine vielstufige Hierarchie von Dateien und Dateiverzeichnissen haben, die es uns erleichtert, uns unter den vielen Dateien auf ihr zurechtzufinden.

Eine solche Hierarchie darf aber auch nicht zu unübersichtlich werden. Es kann sonst zu extrem langen Dateinamen kommen und schwieriger werden, eine bestimmte Datei, deren Namen wir nicht mehr genau wissen, "irgendwo ganz unten" in der Hierarchie zu finden, als beim Arbeiten ganz ohne Dateiverzeichnisse. Glücklicherweise gibt es im CLI aber auch das Konzept des aktuellen Dateiverzeichnisses, in dem Dateien gesucht werden, wenn deren Namen kein Name eines Dateiverzeichnisses vorangestellt wird. Mehr dazu aber weiter unten.

8.1.4 Geräte

Noch "tiefer" als die Konzepte von Dateien und Dateiverzeichnissen liegt im AmigaDOS das Konzept des "Gerätes". Wenn wir einen Computer bedienen, haben wir ja mit einer ganzen Reihe von Geräten zu tun. Dazu gehören z.B. der Computer selbst, der Bildschirm, die Tastatur, die Diskettenlaufwerke, der Drucker, usw. Dies sind die physikalischen Geräte, mit denen man zu tun hat. Daneben gibt es aber noch logische Geräte, die AmigaDOS helfen, die Dateien zu finden, die es für den eigenen Betrieb benötigt (s.u.).

8.1.4.1 Physikalische Geräte

Alle diese Geräte muß man vom CLI mit einem Namen ansprechen können, wenn man sie für einen Befehl benötigt. (In der Workbench haben wir es oft leichter: auf eine Diskette kann man z.B. einfach zeigen.) AmigaDOS und das CLI behandeln solche Geräte dabei wie spezielle Dateien. Es "erkennt" Geräte aber daran, daß hinter ihrem Namen immer ein Doppelpunkt steht. So ist es z.B. möglich, eine Datei auf das Drucker-Gerät zu kopieren (wodurch sie ausgedruckt wird). Dies geht genauso wie das Kopieren einer Datei. Dazu muß statt eines Dateinamens im Kopierbefehl nur der Gerätenamen **PRT:** für Printer (zu deutsch: Drucker) angegeben werden.

Geräte sind aber wirkliche spezielle Dateien, weil man sie meist nicht löschen kann (wie soll man einen Drucker löschen?), sie manchmal nicht wieder lesen kann (wie soll man Daten von einem Drucker lesen?) und eine Reihe anderer Operationen, die mit Dateien möglich sind, auch nicht oder nur eingeschränkt verwendet werden können.

Eine weitere wichtige Funktion von Geräten ist, bei Dateinamen angeben zu können, auf welcher der angeschlossenen Disketten sich die Datei befinden soll. Dazu muß der Gerätenamen des Diskettenlaufwerks dem Namen vorangestellt werden. Die Laufwerksnamen beim CLI lauten dabei **DF0:**,

DF1:, **DF2:** bis **DF3:**; Festplatten erhalten die Bezeichnungen **DH0:** und **DH1:**.

Statt **Persönliches/Meier** müßte man z.B. **DF0:Persönliches/Meier** schreiben, wenn man eine Datei auf der Diskette im internen Laufwerk meint und **DF1:Persönliches/Meier**, wenn man eine Datei auf der Diskette im externen Laufwerk meint.

Genauso wie es aber ein aktuelles Dateiverzeichnis gibt, gibt es auch immer ein aktuelles Laufwerk, auf dem Dateien gesucht werden, deren Name kein Laufwerk vorangeht. Im allgemeinen wird deshalb nur selten ein Laufwerksname benötigt.

8.1.4.2 Logische Geräte

AmigaDOS — und damit das CLI — kennen neben den physikalischen Geräten auch noch logische Geräte oder noch besser gesagt: logische Diskettenlaufwerke. Dies sind Diskettenlaufwerke, die in Wirklichkeit Dateiverzeichnissen entsprechen. Auf diesen logischen Disketten sucht das CLI immer Dateien, deren Name der Benutzer eingetippt hat, die aber im aktuellen Dateiverzeichnis nicht zu finden sind.

Z.B. enthält das logische Diskettenlaufwerk **C:** die Programme, die die Befehle des CLI, die wir im nächsten Kapitel noch kennenlernen werden, ausführen. Tippt man einen Befehlsnamen ein, und im aktuellen Dateiverzeichnis befindet sich kein Programm dafür (was meistens der Fall sein dürfte), so sucht das CLI dieses Programm stattdessen auf dem "Laufwerk" **C:**. Es gibt natürlich kein spezielles Diskettenlaufwerk für solche Programme, sondern sie befinden sich üblicherweise im Dateiverzeichnis **C** der Startdiskette. Das CLI speichert intern aber die Assoziation ab, daß die Dateien des Laufwerks **C:** im Dateiverzeichnis **<Startdiskette>:C** zu finden sind. Diese Startdiskette (die Workbench-Diskette, mit der der Amiga gestartet wurde) selbst entspricht dem logischen Laufwerk **SYS:**. Statt den eventuell langen Namen der Startdiskette oder den Laufwerksnamen zu verwenden, kann man bei der Angabe der Namen von Dateien, die auf der Startdiskette liegen sollen, immer auch **SYS:** sagen.

Dies spart manchmal Tipparbeit. (Die Namen einiger anderer logischer Laufwerke sind bedeutend kürzer als die entsprechenden Pfadnamen zu den Dateiverzeichnissen.) Ein viel bedeutenderer Vorteil ist aber, daß die Zuordnungen von logischem Laufwerk zu wirklichem Dateiverzeichnis geändert werden können. Wenn Sie z.B. die Startdiskette (Workbench) aus dem Diskettenlaufwerk herausnehmen und einen CLI-Befehl eintippen, dann schaut das CLI auf dem logischen Laufwerk **C:** nach dem entsprechenden

Programm. Dazu muß aber die Diskette wieder eingelegt und eine andere entfernt werden. Das ist lästig! Befinden sich aber dieselben CLI-Befehle auf einer anderen Diskette, die sowieso gerade in einem Laufwerk ist, so können wir (mit einem entsprechenden Befehl) sagen: "Bitte, liebes CLI, das logische Laufwerk **C:** ist jetzt im Dateiverzeichnis **DF1:programme** zu finden!". Besitzen wir gar eine Festplatte, so kann man mit einer ähnlichen Zuordnung erreichen, daß diese als Startdiskette aufgefaßt wird — und damit viele Vorgänge viel schneller ablaufen.

8.1.5 Zusammenhänge

Zum Abschluß dieser kleinen Einführung in die grundlegenden Konzepte des CLI noch ein paar Worte zum Zusammenhang zwischen Dateiverzeichnissen und Workbench-Schubladen bzw. zwischen Dateien und Projekt-Icons oder Werkzeug-Icons. Jede Workbench-Schublade entspricht immer auch einem Dateiverzeichnis, das wir auch im CLI oder von einem Programm aus ansprechen können. Jedes Projekt (Dokument) und jedes Werkzeug (Programm), das in einem Fenster der Workbench erscheint, ist immer auch eine Datei. Der Name eines Schubladen-Icons ist dabei auch immer gleich dem Namen des Dateiverzeichnisses und der Name des Projekt-Icons gleich dem Dateinamen.

Umgekehrt gilt diese Beziehung aber nicht. Schon auf der Standard-Workbench befinden sich eine ganze Reihe von Dateiverzeichnissen (und natürlich auch Dateien darin), die in der Workbench nicht zu sehen sind. Dies sind Dateien, von denen die Entwickler des Amiga meinen, daß sie der "gemeine Benutzer" nicht zu sehen braucht, bzw. gar nicht sehen darf, damit er nicht irgendwelchen Quatsch damit anstellt. Diese Einstellung ist so falsch nicht. Eine wichtige Datei, die man nicht sieht, kann man nicht löschen.

Erst wenn man sich in das CLI begibt, bekommt man alle Dateien und Dateiverzeichnisse über die entsprechenden CLI-Befehle zu sehen. Die Entwickler des Amiga waren offensichtlich der Meinung, daß Benutzer, die sich bis zum CLI vorgearbeitet haben, bereits soviel Wissen und Verantwortungsbewußtsein (und Vorsicht !) erworben haben, daß man ihnen selbst das Innerste des Amiga "ausliefern" kann.

In der Workbench werden Dateien und Dateiverzeichnisse erst dann sichtbar, wenn es eine Datei gleichen Namens aber mit der zusätzlichen Endung ".info" gibt. Ein Icon namens **Meier** taucht also erst dann in der Workbench auf, wenn es auch eine Datei namens **Meier.info** gibt. (Sie können das sehr leicht mit dem **dir**-Befehl, den wir ja bereits in Aktion gesehen haben,

feststellen.) Zu allen Namen, die aufgelistet werden und die Sie auch auf der Workbench sehen, gibt es ein Pendant mit dem angehängten ".info". Diese zusätzliche Datei enthält unter anderem das Icon der Datei und weitere Verwaltungsinformationen, die nur Workbench benötigt. Mehr dazu erfahren Sie im Kapitel *Tips für das CLI*.

8.2 Anwendung

Nachdem Sie jetzt einen ungefähren Eindruck davon bekommen haben, was Dateien, Dateiverzeichnisse und Geräte sind, wollen wir nun kurz darauf eingehen, wie man diese typischerweise bei der Arbeit gebraucht.

8.2.1 Gebrauch von Dateiverzeichnissen

Der Umgang mit Dateiverzeichnissen ist besonders wichtig für alle Arbeiten im CLI. Hier ist besonders das Konzept des *aktuellen Dateiverzeichnisses* zu nennen. Wenn dieses geschickt gewählt wird, kann man sich enorm viel Tipparbeit sparen.

8.2.1.1 Das aktuelle Dateiverzeichnis

Mit dem CLI-Befehl **cd** (für *change current directory*; setze aktuelles Dateiverzeichnis) kann ein beliebiges Dateiverzeichnis zum *aktuellen Dateiverzeichnis* erklärt werden. Tippt man später Dateinamen ein, so sucht das CLI die Dateien ausgehend von diesem Dateiverzeichnis. Wenn man z.B. die beiden folgenden Befehle eingibt, so versucht das CLI die Datei *texte/privat/brief1* auf der Diskette im interenen Laufwerk (*df0:*) zu finden und dann auf dem Bildschirm auszugeben (*type* ist ein Befehl, mit dem man Textdateien am Bildschirm betrachten kann).

```
1>cd df0:texte/privat
1>type Brief1
```

Vor jeden Dateinamen setzt das CLI immer den Namen des aktuellen Dateiverzeichnisses, um den *vollständigen Namen* (Pfadnamen; s.u.) der Datei zu erhalten. Man kann sich einer Datei, die in einem Dateiverzeichnis "weit unten" in der Hierarchie liegt, auch schrittweise nähern. Die folgenden

Befehlsfolgen z.B. leisten auf verschiedene Art alle dasselbe wie die beiden Befehle oben.

```
1>type df0:texte/privat/Brief1
```

```
1>cd df0:
```

```
1>type texte/privat/Brief1
```

```
1>cd df0:
```

```
1>cd texte
```

```
1>type privat/Brief1
```

```
1>cd df0:
```

```
1>cd texte
```

```
1>cd privat
```

```
1>type Brief1
```

Betrachten wir einmal die letzte Befehlsfolge. Dabei wird zunächst das aktuelle Dateiverzeichnis auf das interne Laufwerk gesetzt. (Jeder Laufwerksname ist immer auch ein Dateiverzeichnis; s.u.) Dann wird in diesem Laufwerk das Dateiverzeichnis *texte* aktuell, dann in diesem Dateiverzeichnis das Unterverzeichnis *privat*. Und schließlich wird aus diesem die Datei *Brief1* am Bildschirm ausgegeben.

Die verschiedenen Namen hinter dem *type*-Befehl sind verschiedene Formen, ein und dieselbe Datei zu bezeichnen. Der *vollständige Name* (man sagt oft dazu auch *Pfadname*) der bewußten Datei lautet *df0:texte/privat/Brief1*. Der eigentliche Name der Datei der sogenannte *Dateiname*, ist einfach nur *Brief1*. Die verschiedenen anderen Namen sind *Teilpfadnamen* der Datei, ausgehend vom aktuellen Dateiverzeichnis. Der Begriff Pfad kommt daher, daß man AmigaDOS praktisch immer mitteilen muß, auf welchem Weg oder Pfad es durch die verschiedenen Dateiverzeichnisse zu einer Datei findet.

8.2.1.2 Die Wurzel des Dateiverzeichnisses

Mit dem Befehl *cd* wird neben dem aktuellen Dateiverzeichnis immer auch das *aktuelle Laufwerk* gesetzt. Es ist natürlich das Laufwerk, das dieses Dateiverzeichnis enthält. Das aktuelle Laufwerk erhält immer dann eine Bedeutung, wenn bei einem Dateinamen festgelegt werden muß, auf welcher Diskette die entsprechende Datei liegen soll, bzw. in welchem Laufwerk diese Diskette liegt.

Jede Diskette enthält natürlich ein oberstes Dateiverzeichnis, in dem immer dann eine Datei oder ein anderes Dateiverzeichnis liegt, wenn sie auch auf der

Workbench im Diskettenfenster und nicht in einer Schublade erscheinen würde. Dieses Dateiverzeichnis ist die sogenannte *Wurzel*, in der alle Pfadnamen beginnen. Die Wurzel hat keinen eigenen Namen, sondern trägt den Namen der Diskette, wie er auf der Workbench unter dem Disketten-Icon auftauchen würde. Heißt die Diskette z.B. **Werkbank**, so könnte man das oberste Dateiverzeichnis dieser Diskette unter dem Namen **Werkbank:** ansprechen, als wäre die Diskette ein Gerät (s.o.). Es spielt dabei keine Rolle, in welchem Laufwerk sich die Diskette befindet! Wenn sie im Moment gar nicht eingelegt ist, fordert AmigaDOS Sie mittels eines Requesters auf, diese einzulegen.

Wollen Sie hingegen das oberste Diskettenverzeichnis der Diskette ansprechen, die sich im externen Laufwerk befindet, egal wie diese Diskette heißt, so können Sie den Namen **DF1:**, den Gerätenamen des externen Laufwerks, verwenden. Bei Disketten, die sich in einem angeschlossenen Laufwerk befinden, gibt es also immer zwei Möglichkeiten, ihr oberstes Dateiverzeichnis zu benennen: den Diskettennamen oder den Gerätenamen des Laufwerks.

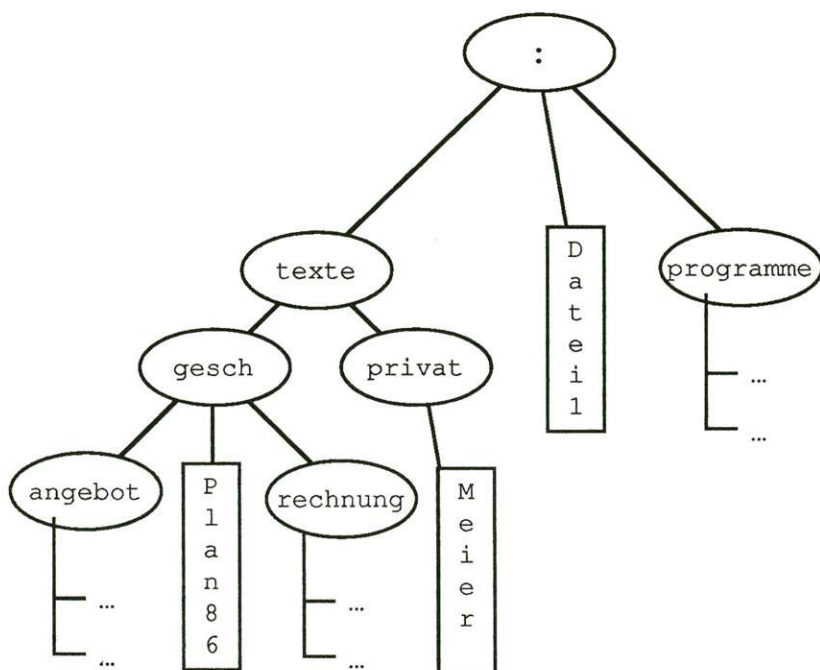


Bild 8 - 1: Der Dateienbaum

Die obige Abbildung zeigt einen solchen Dateienbaum, wie er auf einer Diskette existieren könnte. Dateiverzeichnisse sind in diesem durch Ovale, Dateien durch Rechtecke gekennzeichnet. (Einige Zweige des Baums sind aus Platzgründen nur durch Pünktchen angedeutet.) Es entspricht einer allgemein gültigen Konvention, solche Dateienbäume "auf dem Kopf", also mit der Wurzel oben darzustellen. Lassen Sie dich dadurch nicht verwirren — es ist einfach so üblich.

Die Schreibarbeit beim Namen des aktuellen Laufwerks können Sie sich sparen. Die Wurzel des aktuellen Laufwerks heißt einfach nur ":" (Doppelpunkt). Der vollständige Pfadname der Datei **Plan86** aus Bild 8–1 wäre also **:texte/gesch/Plan86**.

8.2.1.3 Auf- und Absteigen im Dateibaum

Im Moment dürfte noch nicht ganz klar sein, wie man Dateien anspricht, die sich nicht im aktuellen Dateiverzeichnis oder in einem "tiefer liegenden" Dateiverzeichnis befinden. Im Abschnitt 8.2.1.1 war ja die Rede davon, daß das CLI immer den Namen des aktuellen Dateiverzeichnisses vor jeden Dateinamen setzt, den wir eintippen. Auf diese Weise könnte man zwar alle Dateien und Dateiverzeichnisse benennen, die hierarchisch tiefer liegen, aber nie die, die hierarchisch höher liegen.

Die Aussage von oben stimmt aber (glücklicherweise) nicht ganz, wie Sie sich inzwischen vielleicht auch gedacht haben. Die echte Regel lautet:

Das CLI setzt den Namen des aktuellen Dateiverzeichnisses vor einen eingetippten Dateinamen, sofern dieser nicht mit einem Laufwerks- oder Diskettenamen beginnt!

Steht in einem Datei- oder Dateiverzeichnisnamen ein Doppelpunkt, so nimmt das CLI an, daß davor der Name eines Laufwerks (eventuell auch eines logischen Laufwerks) oder einer Diskette steht und dahinter der vollständige Pfadname der Datei oder des Dateiverzeichnisses folgt. Steht der Doppelpunkt an erster Stelle des Namens, so wird der dahinter folgende Text als der vollständige Pfadname einer Datei im aktuellen Laufwerk genommen.

Deswegen geht man als Anwender üblicherweise so vor, daß man Dateinamen oder Teilpfadnamen für alle Dateien verwendet, die hierarchisch unterhalb des aktuellen Dateiverzeichnisses liegen und den vollständigen Pfadnamen für alle Dateien, die hierarchisch höher oder in einem anderen Zweig liegen. Die Wichtigkeit der richtigen Wahl des aktuellen Dateiverzeichnisses dürfte auch deshalb jetzt wohl klarer sein.

Die folgende Abbildung zeigt wieder einen Dateienbaum. In diesem ist das aktuelle Dateiverzeichnis durch ein fettes Oval markiert und zwei Dateien sind durch ein fettes Rechteck hervorgehoben.

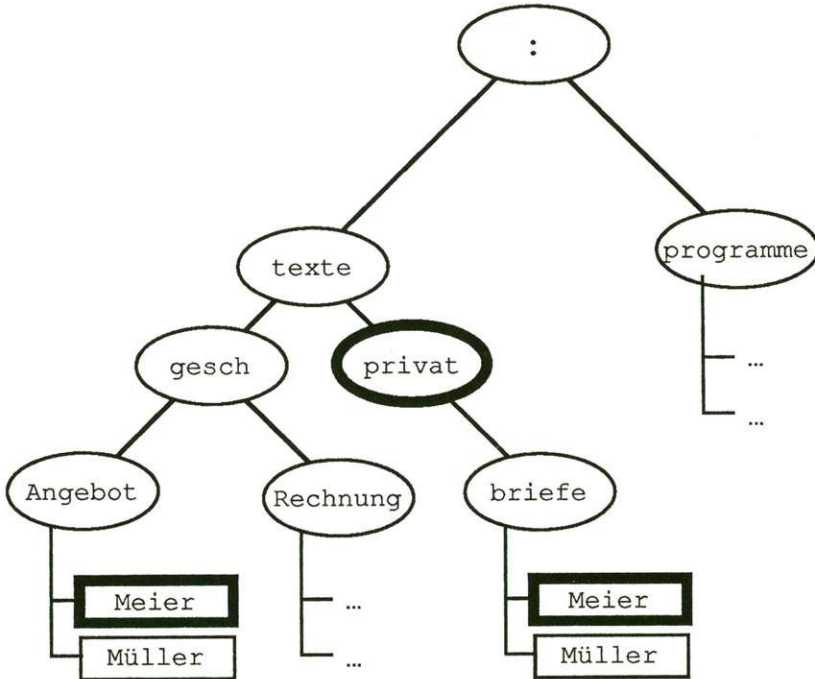


Bild 8 - 2: Namenskonventionen im Dateienbaum

Die Namen, die man für diese Dateien im CLI eintippen müsste, würden **:texte/gesch/angebot/Meier** und **briefe/Meier** lauten (beim aktuellen Dateiverzeichnis **:texte/privat**).

In sehr verästelten Dateibäumen kann es aber unpraktisch sein, so vorzugehen. Wenn der Weg von der Wurzel zur gesuchten Datei sehr weit ist, **kann** es weniger Tipparbeit bedeuten, ihn ausgehend vom aktuellen Dateiverzeichnis auch für Dateien zu bilden, die hierarchisch übergeordnet sind oder in einem anderen Zweig liegen.

Auch dies ist möglich. Ähnlich wie das Zeichen **:** zur Wurzel der Dateiverzeichnisse auf einer Diskette führt, und man einen Namen dann ausgehend von dieser Wurzel angeben kann, gibt es ein Zeichen, mit welchem man im Baum nur "eine Stufe höher" kommt. Schreibt man zu Beginn eines Dateinamens einen Schrägstrich, so entspricht dies dem Dateiverzeichnis, in dem das aktuelle Dateiverzeichnis liegt. Dies kann man verwenden, um im

Baum "höher zu steigen" oder auch um zunächst höher zu steigen und dann wieder in einem Nachbar-Ast "abzusteigen".

Betrachten wir dazu noch einmal Bild 8-2. Die Datei **Meier** im Dateiverzeichnis **angebot** konnte ja nur erreicht werden, indem man mit **:texte/gesch/angebot/Meier** im Baum ganz noch oben ging. Mit der besonderen Funktion des Schrägstrichs könnten wir aber auch schreiben: **/gesch/angebot/Meier**. Der Schrägstrich führt uns zunächst in das Dateiverzeichnis **texte** über dem aktuellen Dateiverzeichnis. Von diesem aus geht es dann ins Dateiverzeichnis **gesch**, dann nach **angebot** und in diesem schließlich findet sich die Datei **Meier**. Der Schrägstrich kann besonders, wenn sich über dem aktuellen Dateiverzeichnis bereits eine sehr komplexe Hierarchie befindet, viel Tipparbeit sparen.

Indem zu Beginn eines Dateinamens mehrere Schrägstriche geschrieben werden, kann man auch mehrere Stufen im Baum "nach oben klettern".

8.2.2 Gebrauch von Gerätenamen

Gerätenamen werden Sie — mit wenigen Ausnahmen — wahrscheinlich nicht so oft benötigen, wie die Namen von Dateien und Dateiverzeichnissen. Am häufigsten werden Sie wohl die Namen von Diskettenlaufwerken und des Druckers verwenden.

Wenn Sie Ihren Amiga einschalten, kennt AmigaDOS insgesamt 13 physikalische und 7 logische Geräte.

8.2.2.1 Physikalische Geräte

AmigaDOS kennt insgesamt 13 physikalische Geräte. Die folgende Tabelle enthält eine Aufstellung ihrer Namen und jeweils ein Stichwort zu Ihrer Bedeutung.

Geräte-Name	Bedeutung
NIL:	Leeres Pseudogerät.
SER:	Serielle Schnittstelle
PAR:	Parallele Schnittstelle
PRT:	Drucker
CON:	Konsole (Bildschirm und Tastatur)
RAW:	Spezielle Konsole
DF0:	Internes Diskettenlaufwerk
DF1:	1. externes Diskettenlaufwerk.
DF2:	2. externes Diskettenlaufwerk.
DF4:	3. externes Diskettenlaufwerk.
DH0:	1. Festplattenlaufwerk.
DH1:	2. Festplattenlaufwerk.
RAM:	Pseudo-Diskettenlaufwerk im RAM-Speicher.

Tabelle 8 - 1: *Physikalische Geräte des Amiga*

Wofür Sie die Namen der Diskettenlaufwerke benötigen, haben Sie bereits im vorangegangenen Abschnitt erfahren. Der vollständige Pfadname einer Datei beginnt immer mit dem Namen der Diskette/Festplatte oder mit dem Namen des Disketten-/Festplattenlaufwerks. Im Betriebssystem des Amiga sind insgesamt vier Diskettenlaufwerke (DF0: bis DF3) und zwei Festplattenlaufwerke vorgesehen (DH0: und DH1:). (Sie können diese natürlich nur dann in einem Pfadnamen verwenden, wenn sie auch angeschlossen sind.)

Ein ganz besonderes Diskettenlaufwerk ist auch noch RAM:. Das Gerät RAM: simuliert ein Diskettenlaufwerk bzw. eine Diskette im RAM-Speicher des Amiga. Da diese Diskette keine beweglichen Teile benötigt, ist sie natürlich extrem schnell. Wenn Daten auf diese Disketten geschrieben oder von ihr gelesen werden, werden sie in Wirklichkeit nur im Speicher hin- und hergeschauelt. Wenn Sie dieses superschnelle Laufwerk, welches beim Start zunächst leer ist, verwenden, brauchen Sie nur Dateien in das Dateiverzeichnis RAM: zu kopieren (nicht vergessen, jeder Laufwerksname ist immer auch die Wurzel eines Dateienbaums). Diese Dateien können Sie dann so verwenden, als lägen Sie auf einer "ganz normalen Diskette".

NIL: ist ebenfalls ein ganz besonderes "Gerät". Wie oben ja bereits angedeutet, können Gerätenamen manchmal wie Dateinamen für Befehle verwendet werden. D.h.: man kann Daten statt in eine Daten zu einem Gerät schicken oder von diesem holen. Manchmal kann es aber auch sinnvoll sein, z.B. für Testzwecke, Daten "ins Nichts" zu schicken. Dazu dient NIL:. Kopiert man z.B. eine Datei zum Gerät NIL:, so verschwinden die Daten auf

Nimmerwiedersehen in diesem Gerät. NIL: wird wahrscheinlich nur von fortgeschrittenen Benutzern und auch von diesen nur sehr selten benötigt.

Etwas "bodenständiger" sind da schon die Geräte SER:, PAR: und PRT:. SER: bezeichnet die serielle Schnittstelle (Steckdose) an der Rückseite des Amiga, PAR: die parallele Schnittstelle. Sie können verschiedene Geräte an diese Stecker anschließen, die sowohl Daten empfangen, als auch senden können. Dies können z.B. Drucker und Geräte zur Telekommunikation (Modems) sein. Die Software des Amiga sorgt dafür, daß sich solche Geräte wie Dateien verhalten, um etwas einfacher damit umgehen zu können. Je nachdem, was für ein Gerät Sie an welcher Schnittstelle angeschlossen haben, können SER: und PAR: sowohl zum Lesen als auch zum Schreiben von Daten verwendet werden. Von Ausnahmen abgesehen, werden Sie diese beiden Gerätenamen aber trotzdem sehr selten benötigen, da mit den Geräten, die Sie dort anschließen, meist auch Programme kommen, die sich um solche Details selbst kümmern.

Das Gerät PRT: hingegen werden Sie häufiger verwenden. PRT: steht für *printer* und bezeichnet den Drucker, unabhängig davon, an welche Schnittstelle Sie ihn angeschlossen haben. Kopieren Sie eine (Text-) Datei nach PRT:, so wird diese ausgedruckt. PRT: kümmert sich dabei automatisch um solche Feinheiten wie welcher Drucker angeschlossen ist, wie schnell man mit ihm kommunizieren kann, ob er farbig drucken kann, usw.. Verwenden Sie stets PRT:, wenn Sie einen Text ausdrucken wollen und nicht SER: oder PAR:! PRT: kann nur Daten empfangen und keine abgeben — im Gegensatz zu SER: und PAR:, die, wenn die entsprechende Geräte angeschlossen sind (keine Drucker), auch Daten senden können.

Wenn Sie vom CLI aus eine Datei nach PRT: schicken, sollten Sie aber darauf achten, daß es wirklich eine sogenannte "reine Textdatei" ist, die nur druckbare Buchstaben und Zeilenvorschübe enthält. Der Text-Editor *ed*, der in einem folgenden Kapitel beschrieben wird, erzeugt z.B. solche Dateien. Erstellen Sie mit einem speziellen Textverarbeitungsprogramm einen Text, in dem Sie verschiedene Farben und/oder Zeichensätze und Größen verwenden, so müssen Sie diesen im allgemeinen auch innerhalb des Textverarbeitungsprogramms ausdrucken. PRT: versteht solche Dateien nicht und liefert deshalb völligen Quatsch auf dem Papier.

Die Geräte CON: und RAW: entsprechen den "Geräten" des Amiga, mit denen Sie wohl am meisten zusammenarbeiten: der Tastatur und dem Bildschirm. Zusammengefaßt nennt man Tastatur und Bildschirm in der Computerbranche meist *Konsole*. Wenn Sie eine Textdatei nicht ausdrucken, sondern zunächst einmal am Bildschirm betrachten wollen, so können Sie sie nach CON: schicken. Das Gerät RAW: verhält sich im wesentlichen gleich wie CON:, reagiert aber auf Steuerzeichen in besonderer Weise. RAW: ist

eigentlich nur für Programmierer interessant und soll deshalb hier nicht weiter besprochen werden.¹

CON: hat eine Besonderheit, die dieses Gerät von anderen unterscheidet. Da der Amiga ja alle Ein- und Ausgaben in Fenstern macht, reicht es nicht, eine Datei nach CON: zu schicken, um sie zu sehen. Wir müssen zusätzlich erst angeben, wie das Fenster aussehen soll, in dem die Datei gezeigt werden soll. Die folgende Befehlszeile schickt die Datei *text* z.B. in ein neues Fenster mit dem Namen **Fenster-B**, das oben links am Bildschirm erscheint und 400 Punkte breit sowie 100 Punkte hoch ist.

```
1>copy text TO CON:20/10/400/100/Fenster-B
```

Die allgemeine Form für die Angabe eines CON:-Gerätes lautet:

CON:X/Y/Breite/Höhe/Fenster-Titel

Dabei ist *X* der Abstand des linken Fensterrandes vom linken Bildschirmrand, *Y* der Abstand vom oberen Bildschirmrand, **Breite** die **Breite** des entstehenden Fensters, **Höhe** entsprechend die Höhe und **Fenster-Titel** der Name, der oben links im neuen Fenster erscheinen soll. Enthält dieser Name Leerzeichen, so muß der gesamte Gerätenamen mit Gänsefüßchen eingerahmt werden. Das Gerät mit der Bezeichnung "**CON:10/10/100/100/der Text**" ist ein Beispiel dafür, wie dies aussehen könnte.

Im folgenden Kapitel wird der Befehl **newcli** erläutert, der ein neues CLI-Fenster erzeugt, in dem Sie unabhängig vom ersten Fenster Befehle eingeben können. Dieser Befehl benötigt die Angabe des Gerätes, von dem aus das CLI seine Befehle empfängt und auf dem es seine Meldungen ausgeben kann. Dies wird normalerweise ein CON:-Gerät sein. Sie können so durch die Angabe der entsprechenden Teile des vollständigen CON:-Namens schon beim Erzeugen des neuen CLI-Fensters angeben, wo es erscheinen soll und wie groß es sein soll.²

¹ Wenn Sie eine experimentierfreudige Natur besitzen, können Sie die Eigenschaften von RAW: selbst herausfinden, indem Sie in den folgenden Beispielen einfach immer CON: durch RAW: ersetzen. Eine Warnung aber vorweg: Wenn Sie nicht sehr vorsichtig dabei sind, kann es bei solchen Experimenten durchaus nötig werden, den Amiga neuzustarten!

² Es kann, für fortgeschrittene Benutzer, durchaus interessant sein, ein CLI auch für andere Geräte zu erzeugen. So ist es möglich, CON: durch RAW: zu ersetzen (was nur sehr selten sinnvoll ist); es kann sogar sinnvoll sein, **newcli SER:** einzugeben, wenn dieses neue CLI von einem anderen Computer aus gesteuert werden soll.

Wichtig ist bei der Verwendung von CON: auch, daß dabei immer ein **neues Fenster** geöffnet wird, dessen Eigenschaften man ja auch angeben muß. Dies kann Vorteile haben, wie man vielleicht beim newcli-Befehl gesehen hat. Man kann z.B. auch den Inhalt einer Datei in einem neuen Fenster zeigen, wobei der Inhalt des CLI-Fensters unbeeinträchtigt bleibt. Dies Verhalten ist aber nicht unbedingt immer das, was man möchte.

Will man, daß ein Befehl genau das Fenster für Ein- oder Ausgaben nimmt, in dem man selbst gerade mit dem CLI arbeitet, so darf man nicht das Gerät CON: verwenden. Das aktuelle CLI-Fenster wird immer durch das Zeichen * (Stern) symbolisiert. Nehmen wir das Beispiel von oben, bei dem eine Text-datei am Bildschirm ausgegeben wird. Soll dies nicht in einem neuen, sondern im aktuellen Fenster geschehen, so müßte der Befehl lauten:

```
1>copy text TO *
```

8.2.2.2 Logische Geräte

Während physikalische Geräte — wie z.B. PRT: und CON: — meist verwendet werden, als wären es Dateien, in denen man Daten ablegen oder daraus lesen kann, werden logische Geräte fast immer wie Dateiverzeichnisse verwendet. Diese "Zwiespältigkeit" findet man bei den Gerätenamen der Diskettenlaufwerke ebenfalls, die einerseits natürlich Geräte bezeichnen, meist aber nur die Wurzel des Dateienbaums auf der Diskette in diesem Laufwerk. Deshalb nennt man logische Geräte oft auch *logische Laufwerke*.

Welchem Dateiverzeichnis ein logisches Laufwerk zugeordnet ist, kann man zudem jederzeit mit Hilfe des CLI-Befehls **assign** ändern. Dies geht bei physikalischen Geräten natürlich nicht; DF0: ist immer das interne Laufwerk. Die Hauptfunktion logischer Laufwerke ist es, dem Betriebssystem des Amiga zu sagen, wo es bestimmte Dateien findet, die für den Betrieb benötigt werden. Hätte man festgelegt, daß diese Dateien immer in bestimmten Dateiverzeichnissen auf bestimmten Disketten zu finden sein müssen, müßte man Einbußen an Flexibilität hinnehmen. Beim Start werden die logischen Laufwerke automatisch bestimmten Dateiverzeichnissen auf der Start-Diskette zugeordnet, in denen die jeweiligen Dateien zu finden sind. Danach kann man diese Zuordnungen aber ändern, wie man möchte.

Die Zeichensätze, die im Werkzeug **Notepad** zur Verfügung stehen, werden z.B. auf dem logischen Laufwerk FONTS: gesucht. Normalerweise ist dieses Laufwerk dem Dateiverzeichnis **fonts** auf der Startdiskette zugeordnet. Hat man aber eine spezielle Diskette, die nur mit Zeichensätzen gefüllt ist so kann man vor dem Aufruf von **Notepad** einfach folgenden Befehl geben:

1>**assign FONTS: FontDisk:Spezial**

In Folge dieses Befehls wird *Notepad* die Zeichensätze, die es im Menü *Fonts* anbietet, dann auf der Diskette *FontDisk* im Dateiverzeichnis *Spezial* suchen. Auf ähnliche Weise kann man auch andere Zuordnungen im Betriebssystem seinen Bedürfnissen anpassen.

Die folgende Tabelle gibt einen Überblick über die logischen Geräte, die standardmäßig vom AmigaDOS benutzt werden, sowie darüber, welchen Dateiverzeichnissen sie beim Start zugeordnet werden und wozu sie dienen.

Gerätename	Dateiverzeichnis	Aufgabe
SYS:	StartDisk:	
C:	StartDisk:c	(Befehle des CLI)
L:	StartDisk:l	(System-Libraries)
S:	StartDisk:s	(Kommandofolgen)
LIBS:	StartDisk:libs	(Libraries)
DEVS:	StartDisk:devs	(Geräte-Treiber)
FONTS:	StartDisk:fonts	(Zeichensätze)

Tabelle 8 - 2: *Logische Geräte des Amiga*

Eine zentrale Bedeutung bei diesen logischen Geräten nimmt SYS: ein. Dies ist der logische Diskettenname für die Diskette, mit der der Amiga gestartet wurde (die sog. Workbench-Diskette). Der wirkliche Name dieser Diskette wird in der Tabelle immer mit *StartDisk* bezeichnet. Benötigt man irgendwelche Dateien oder Dateiverzeichnisse auf der Startdiskette, so kann man immer den Namen SYS: für das Wurzel-Dateiverzeichnis verwenden, egal in welchem Laufwerk die Diskette liegt, oder ob sie überhaupt in einem Laufwerk ist.

C: ist das logische Laufwerk, in dem die Programme gesucht werden, die die CLI-Befehle (*commands*) ausführen (jeder dieser Befehle ist in Wirklichkeit ein einzelnes Programm). Wenn Sie einen CLI-Befehl aufrufen, und das CLI findet das entsprechende Programm nicht im aktuellen Dateiverzeichnis, so sucht es auf dem Gerät C: danach. Nehmen wir an, Sie haben auf einer anderen Diskette als der Startdiskette einen vollständigeren Satz von CLI-Befehlen, der sich dort ebenfalls im Dateiverzeichnis *c* befindet. Der folgende Befehl sorgt dann dafür, daß das CLI immer, wenn Sie einen Befehl eintippen, dort nach dem entsprechenden Programm sucht (allerdings überhaupt nicht mehr auf der Startdiskette):

1>assign C: AndereDisk:c

Die Geräte L: und LIBS: enthalten Programmbibliotheken (engl. *Libraries*) die vom System bzw. von anderen Programmen genutzt werden können. Diese Bibliotheken werden erst dann für Sie interessant, wenn Sie den Amiga programmieren. Es ist selten nötig, deren Zuordnungen zu einem Dateiverzeichnis zu ändern.

Genauso befinden sich auf dem logischen Laufwerk DEVS: (für *devices* oder zu deutsch *Geräte*) die *Geräte-Treiber*. Geräte-Treiber enthalten, ähnlich wie Bibliotheken, kleine Programme, die von anderen Programmen benötigt werden. In diesem Fall helfen diese kleinen Programme, die verschiedenen Geräte, die man an die Schnittstellen an der Rückseite Ihres Amiga anschließen kann, zu bedienen. Dies können ja sehr verschiedene Geräte sein, die man beim Schreiben der Betriebssystem-Software nicht schon alle vorhersehen konnte. Deshalb ist es sinnvoller, deren Steuerung auf solche Geräte-Treiber zu verlagern, die man problemlos austauschen kann, wenn Sie einmal ein neues Gerät bekommen.

Das logische Laufwerk FONTS: wurde oben bereits erläutert. Auf dem Laufwerk S: schließlich befinden sich die *Kommandofolgen* (*Sequenzen*). Kommandofolgen bieten die Möglichkeit, eine Folge von CLI-Befehlen zu einem neuen Befehl zusammenzufassen. Wenn Sie eine solche Kommandofolge aufrufen und das CLI findet sie nicht im aktuellen Dateiverzeichnis, so sucht es auf dem Gerät S: danach. Mehr zu Kommandofolgen erfahren Sie in einem der folgenden Kapitel.

Viele der üblichen Zuordnungen von logischen Laufwerken zu Dateiverzeichnissen werden Sie kaum jemals ändern müssen. Es kann jedoch in einem Fall nötig werden, sie alle zu ändern. Falls Sie sehr viel RAM-Speicher in Ihrem Amiga oder gar eine Festplatte besitzen, so kann es sehr sinnvoll sein, diese als "Startdiskette" zu verwenden (obwohl Sie natürlich nicht von dem entsprechenden Gerät starten können). Dies ist vor allem deshalb "sehr sinnvoll", weil dann alle Vorgänge wesentlich schneller ablaufen. Eine Festplatte und erst recht das RAM-Laufwerk sind Disketten von der Geschwindigkeit her weit überlegen. Die folgende Befehlsfolge würde das Gewünschte erreichen (sie nimmt an, daß Sie eine Festplatte, DH0: besitzen).³

³ Falls Sie wirklich stolzer Besitzer einer Festplatte sind, sollten Sie sich diese Folge aus Gründen der Arbeitersparnis natürlich in einer Kommandofolge abspeichern.

```
assign SYS:      DH0:
assign C:        DH0:c
assign L:        DH0:l
assign S:        DH0:s
assign LIBS:     DH0:libs
assign DEVS:     DH0:devs
assign FONTS:    DH0:fonts
```

Sie können danach die wirkliche Startdiskette aus dem Laufwerk nehmen und werden sie nie wieder benötigen, solange der Amiga eingeschaltet bleibt.

8.3 Ein-/Ausgabe-Umleitung

Ein weiteres sehr wichtiges Konzept des CLI ist die sogenannte Ein-Ausgabe-Umleitung (*I/O redirection*). Viele Befehle des CLI lesen Ihre Eingaben — sofern sie überhaupt welche benötigen — von einer Datei und schreiben alle Ausgaben in eine andere. Normalerweise werden die Eingaben dabei von der Tastatur erwartet und die Ausgaben gehen ins CLI-Fenster. (Auch die Tastatur und das CLI-Fenster sind ja eine Datei — oder besser: ein Gerät, CON:.)

Den Programmen ist es jedoch in der Praxis egal, von welchen Dateien bzw. Geräten sie Ihre Eingaben erhalten und wo sie ihre Ausgaben hinschicken. Deshalb kann man bei den meisten Programmen diese Dateizuordnungen gefahrlos umlegen, z.B. auf den Drucker oder eine Diskettendatei. Sie können auf diese Weise z.B. den Inhalt eines Dateiverzeichnisses mit *dir* auf den Drucker schicken oder in einer Datei abspeichern, die Sie nachträglich mit einem Textverarbeitungsprogramm bearbeiten können.

Um einem Befehl eine andere Eingabedatei zuzuweisen als die Tastatur, müssen Sie sofort hinter dem Befehlsnamen ein Kleinerzeichen (<) und danach den Namen der Datei oder des Gerätes schreiben, von dem die Eingaben kommen sollen. Genauso können Sie alle Ausgaben eines Befehls auf eine Datei oder ein Gerät schicken, indem Sie hinter den Namen des Befehls eine Größerzeichen (>) und dann den Namen der Datei bzw. des Geräts schreiben.

Nehmen wir einmal an, es gäbe einen Befehl namens *date* (es gibt ihn tatsächlich), der seine Eingabe von der Tastatur erwartet und seine Ausgaben in das CLI-Fenster schickt. Tippen Sie aber die folgende Befehlszeile, so holt

er seine Eingaben von der Datei ***Input*** und druckt seine Ausgaben (auf einem angeschlossenen Drucker) aus.

1>**date <Input >PRT:**

Den vollen Nutzen der Ein-/Ausgabe-Umleitung können Sie wahrscheinlich aber erst abschätzen, wenn Sie wirklich täglich mit dem CLI umgehen. Sie läßt sich sehr vielseitig einsetzen und ist deshalb eine Möglichkeit, die seit einigen Jahren in keinem neuen Betriebssystem für Computer mehr fehlen darf.⁴

⁴ Wie übrigens viele der fortschrittlichen Elemente des CLI und von AmigaDOS ist auch die Ein-/Ausgabe-Umleitung vom Betriebssystem UNIXTM abgeschaut worden.

9

Die Kommandos des CLI

In diesem Kapitel wird eine Liste der wichtigsten Befehle aufgestellt, (Kommandos), die man in einem CLI-Fenster geben kann. Sie ist insofern unvollständig, als sie nur diejenigen Befehle umfaßt, die von einem "normalen Anwender" benötigt werden. Befehle, die speziell für Programmierer oder Systementwickler geschaffen wurden, werden nicht aufgeführt. Informationen darüber sind in den Unterlagen, die mit den entsprechenden Programmiersprachen kommen, zu finden. Eine vollständige Liste enthält das *AmigaDOS User's Manual*, das Ihnen Ihr Amiga-Händler beschaffen kann.

9.1 Überblick über die CLI-Kommandos

Alle CLI-Kommandos sind in Wirklichkeit kleine Programme. Anders als bei anderen Computern und deren Betriebssystemen, sind keine Befehle in das Programm **CLI** "eingebaut". Hat man eine Zeile mit einem Befehl an das CLI eingegeben, so nimmt das CLI das erste Wort dieser Zeile und sucht nach einem Programm gleichen Namens. Findet es dieses Programm, so wird es in den Speicher geladen und gestartet. Der Rest der Zeile, hinter dem Namen des Programms, wird als Parameter an das Programm weitergegeben und es bleibt diesem überlassen, was es damit tut.

Das CLI sucht nur an ganz bestimmten Stellen nach solchen Programmen. Zunächst einmal wird im aktuellen Dateiverzeichnis nach einem Programm entsprechenden Namens gesucht. Wird dort keines gefunden, so wird noch

das "logische Gerät" namens **C:** durchsucht.¹ Wird es auch dort nicht gefunden, so gibt es eine Fehlermeldung — sonst geschieht nichts. Das logische Gerät **C:** wird beim Start automatisch dem Dateiverzeichnis **<Startdiskette>:c** zugewiesen.² Diese Zuordnung kann aber mit dem Befehl **assign**, der weiter unten noch ausführlich erklärt wird, jederzeit geändert werden. Solange sie aber nicht geändert wird, wird der Amiga Sie jedesmal, wenn Sie einen CLI-Befehl eintippen und die Startdiskette nicht mehr in einem Laufwerk steckt, dazu auffordern, die Startdiskette einzulegen.

Nach diesen Erklärungen dürfte es wohl klar sein, daß das CLI eigentlich nicht über einen festen Satz von Kommandos verfügt. Löscht man eines der Programme im logischen Gerät **C:**, so verliert man ein mögliches Kommando. Schreibt man selbst ein Programm und legt es ins logische Gerät **C:**, so wird es dadurch automatisch ein CLI-Kommando. Insbesondere wenn Sie mit unterschiedlichen Disketten starten kann es sein, daß Ihnen unterschiedliche Kommandos zur Verfügung stehen. Doch damit genug der verwirrenden Möglichkeiten.

Die im folgenden noch beschriebenen Kommandos stellen den "Standardsatz" von Befehlen dar, so wie er Ihnen beim Kauf des Amiga auf der Workbench-Diskette mitgeliefert wird. Solange Sie keine Modifikationen daran vornehmen, stehen Ihnen all diese Befehle (und einige wenige andere, hier nicht beschriebene) zur Verfügung.

9.2 Format der CLI-Kommandos

Alle CLI-Kommandos verfügen über sich ähnelnde Aufruf-Formate. Am Anfang jeder Zeile, die einen CLI-Befehl enthält, steht der Name des Befehls. Dieser Name besteht immer aus einem kurzen Wort, das im allgemeinen einem (englischen) Verb gleich oder ähnlich ist. Hinter diesem Wort folgt

¹ Directories und logische Geräte wurden ausführlich bereits im letzten Kapitel erläutert. Bei Unklarheiten schauen Sie bitte dort noch einmal nach. Diese beiden Begriffe werden für dieses Kapitel laufend wieder benötigt.

² **<Startdiskette>** muß dabei natürlich gegen den Namen der Diskette ersetzt werden, mit der Sie den Amiga gestartet haben. Haben Sie die Anweisungen zu Beginn dieses Buchteils befolgt, so steht für **Startdiskette** immer **CLI-Diskette**.

meist eine Reihe sogenannter *Parameter* oder *Operanden*, die dem Befehl genauer sagen, was er zu tun hat. Ein Beispiel für einen solchen Befehl mit Parametern ist das CLI-Kommando *dir*, das wir ja bereits kennen. *Dir* ist der Name des Befehls und der Name des Dateiverzeichnisses, das diesem Namen folgen kann, ist ein Parameter.

Parameter sagen dem Befehl, auf welchen Operanden (Dateien, Dateiverzeichnissen, Texten, usw.) er arbeiten soll und wie (falls es mehrere Möglichkeiten gibt). Diese Parameter werden in den im nächsten Abschnitt folgenden Format-Beschreibungen als *Platzhalter* beschrieben. Ein solcher Platzhalter ist immer ein kurzer Begriff, der in spitze Klammern gesetzt wird, (z.B. *<dateiname>*). Ein Platzhalter soll nicht wirklich eingetippt werden, so wie er angegeben wurde, sondern gegen den wirklichen Parameter ersetzt werden. So muß z.B. *<dateiname>* gegen einen echten Dateinamen ersetzt werden.

Wie beim *dir*-Befehl zu sehen, können Parameter *optional* sein, d.h. sie **können, müssen aber nicht** angegeben werden. Wird der entsprechende Parameter nicht angegeben, geht der Befehl von einer (hoffentlich sinnvollen) Voreinstellung dafür aus. Der *dir*-Befehl nimmt dann z.B. an, daß der Benutzer eine Liste der Dateien im aktuellen Dateiverzeichnis sehen will. Ist ein Parameter notwendig, muß er also zur Befehlsausführung eingegeben werden, so wird sein Platzhalter in den folgenden Beschreibungen, wie der Befehlsname selbst, immer **fett** gedruckt. Überhaupt gilt für die Format-Beschreibungen immer, daß fettgedruckte Texte unbedingt eingegeben werden müssen (wobei eventuell Platzhalter gegen andere Texte zu ersetzen sind). Alle anderen Texte sind optional anzugeben.

Solange ein Befehl nur einen möglichen Parameter hat, ist die ganze Sache recht einfach. Der Text, der hinter dem Namen des Befehls folgt, ist der Parameter. Steht hinter dem Namen kein Text mehr, ist der Parameter eben leer. Kann ein Befehl aber mehrere Parameter bekommen, wird die ganze Sache etwas schwieriger. Der Befehl *copy* z.B. benötigt zum Kopieren einer Datei zwei Parameter: welche Datei er kopieren soll, und wie die Kopie heißen soll.

Der Rest der Zeile hinter dem Wort *copy* selbst muß also in zwei Teile gespalten werden. Der erste ist der Name der zu kopierenden Datei, der zweite der Name der Kopie. Damit die Teilungsstelle nicht gar so willkürlich gewählt werden muß, kennt das CLI sogenannte *Schlüsselworte* (engl. *keywords*). Ein Schlüsselwort kann entweder alleine — als eine Art besonderer Parameter — auftreten oder es leitet einen Parameter ein und trennt ihn von den vorangehenden Parametern. Der Befehl *copy* kennt z.B. das Schlüsselwort *TO*, das den Namen einleitet, den die Kopie der Datei

tragen soll. Die folgende Zeile legt eine Kopie der Datei **Text1** unter dem Namen **Kopie** an:

```
1>copy Text1 TO Kopie
```

TO ist eine Schlüsselwort, das ganz eindeutig den Beginn des zweiten Parameters kennzeichnet. Schlüsselworte, die unbedingt so eingetippt werden müssen, wie sie angegeben sind, werden in den folgenden Beschreibungen immer in Großbuchstaben geschrieben, um sie von den Platzhaltern zu unterscheiden, die gegen andere Texte ersetzt werden sollen. Falls sie unbedingt notwendig sind, werden Sie, genauso wie unbedingt zu ersetzende Platzhalter, **fett** gedruckt. Im allgemeinen gilt, daß das Schlüsselwort eines unbedingt nötigen Parameters auch unbedingt nötig ist. Es gibt jedoch Ausnahmen. Manchmal kann man Schlüsselworte für bestimmte Parameter wegfallen lassen, wenn der Befehl auch ohne sie eindeutig ist. Diese Schlüsselworte werden dann nicht fett gedruckt.

9.3 Die wichtigsten CLI-Kommandos

Es folgt nun eine Liste, die Erläuterungen aller der CLI-Kommandos enthält, die Sie als "normaler Benutzer" wahrscheinlich benötigen werden. Der Übersichtlichkeit wegen beginnt jede Erklärung auf einer neuen Seite. Dies dient nicht dazu — wie böswillige Menschen vielleicht behaupten würden — die Seitenzahl dieses Buches zu erhöhen, sondern wirklich nur, Ihnen die Arbeit damit möglichst leicht zu machen.

Alle folgenden Seiten dieses Abschnittes folgen einem festen Format. Oben auf jeder Seite steht der Name des Befehls, danach folgt eine detaillierte Beschreibung des Aufruf-Formats, eine Kurzbeschreibung von Sinn und Zweck des Befehls und schließlich eine etwas ausführlichere Erläuterung für die Anwendung.³ Die Erläuterungen bleiben absichtlich etwas knapp und nüchtern und sind nur mit wenigen kurzen Beispielen versehen. Eine

³ Die in der folgenden Liste angegebenen Formate entsprechen nicht immer den "wirklichen" Formaten. Sie sind aus Verständnisgründen teilweise vereinfacht worden und spiegeln die gebräuchlichsten Formen der Verwendung wieder. Komplette Formatangaben sind selbstverständlich im *AmigaDOS User's Manual* zu finden; meist bieten sie aber nicht mehr Möglichkeiten für Kommandos, sondern nur andere Schreibweisen, die dasselbe leisten wie die hier aufgeführten.

ausführliche Erläuterung jedes der fast 50 Befehle mit Beispielen könnte nämlich alleine schon ein Buch füllen. Mit ein wenig Experimentierfreude werden Sie aber auch die komplexeren Befehle mit Sicherheit rasch anwenden können!

Das Format der Befehlsbeschreibungen soll zwei sehr unterschiedliche Benutzungsweisen des Kapitels erleichtern. Einmal das sequentielle Lesen und Experimentieren, mit dem Sie ja hoffentlich gleich beginnen werden. Gleichzeitig sollen die folgenden Beschreibungen aber auch als Nachschlagwerk für den täglichen Gebrauch des CLIs dienen. Hierfür dürften oft die Beschreibungen des Aufruf-Formats der einzelnen Kommandos genügen.

;

Format: <Befehl>; <Kommentar>**Zweck:** dient der Möglichkeit, erläuternde Kommentare hinter einen CLI-Befehl anzufügen.**Erläuterung:**

Das Semikolon ist ein Zeichen für das CLI, alles dahinterstehende zu ignorieren. Schreibt man hinter einen kompletten CLI-Befehl ein Semikolon und danach einen Text, so hat dieser keine Wirkung. Auf diese Weise kann der entsprechende Befehl mit einem Kommentar versehen werden. Dies ist nicht besonders wichtig, wenn man den Befehl gerade eintippt – schließlich weiß man in diesem Augenblick, was man will – wird aber sehr wichtig, wenn man eine Folge von Befehlen zu einer *Kommandofolge* zusammenfaßt. Mehr über Kommandofolgen im folgenden Kapitel *Die Automatisierung des Amiga* und bei der Erläuterung des Befehls *execute*.

> und <

Format: Befehl >datei2 <datei1

Zweck: dient zum "Umleiten" der Ein- und Ausgaben eines Befehls.

Erläuterung:

Genau wie schon das Semikolon sind auch diese beiden Zeichen kein Befehl im eigentlichen Sinne. Sie modifizieren nur den Befehl, mit dem zusammen sie verwendet werden.

Viele Programme arbeiten nach dem klassischen Prinzip der Datenverarbeitung. Sie lesen ihre Eingaben von einer bestimmten Datei, verarbeiten sie und schreiben die Ergebnisse dann in eine andere Datei. Beide "Dateien" sind standardmäßig die Tastatur und das CLI-Fenster, die ja zusammen die Datei (genauer das logische Gerät) *CON*: bilden. Wenn ein Programm weitere Eingaben erwartet, so müssen sie also über die Tastatur kommen und die Ausgaben werden in das CLI-Fenster geschrieben.

Möchte man aber, daß ein Programm seine Eingaben von einer anderen Datei beziehen soll, so schreibt man direkt hinter dem Namen des Befehls < und dann den Namen der Datei. Genauso kann man > und dann den Namen einer Datei schreiben, worauf der Befehl seine Ausgaben nicht mehr ins CL-Fenster, sondern in diese Datei schickt.⁴ Beide Symbole können sowohl einzeln als auch kombiniert verwendet werden.

Benötigt ein Befehl allerdings mehr als nur eine Ein- und eine Ausgabedatei, so helfen diese Möglichkeiten natürlich nicht mehr weiter. Solche Fälle werden dann von den einzelnen Befehlen in teilweise unterschiedlicher Weise gelöst.

⁴ Diesen Vorgang nennt man Ein-/Ausgabe-Umlenkung (oder auf neudeutsch "I/O redirection"). Er wurde durch das Betriebssystem UNIX™ populär und ist heute bei vielen Betriebssystemen zu finden.

assign

Format: *assign* <Name> <Dir>

Zweck: dient zum Ändern, Abprüfen oder Erzeugen logischer Geräte.

Erläuterung:

Mit dem Befehl *assign* ist es möglich, die Zuordnung von logischen Geräten (logischen Laufwerken) zu Dateien oder Dateiverzeichnissen zu ändern.

Ruft man *assign* ohne Parameter auf, bekommt man eine Liste aller logischen Geräte und der entsprechenden Zuordnungen gezeigt.

Ruft man *assign* auf und tippt dahinter nur einen Namen, so wird dieses logische Gerät gelöscht. **Dies ist sehr gefährlich bei den standardmäßig definierten logischen Geräten! Bei diesen sollte niemals versucht werden, die Zuordnungen zu löschen!**

Der "Normalfall" ist ein Aufruf von *assign* mit zwei Parametern. Der erste ist der Name eines logischen Gerätes, der zweite der Name einer Datei oder meist eines Dateiverzeichnisses, das mit diesem logischen Gerät verbunden werden soll. Der Name eines logischen Gerätes sollte immer mit einem Doppelpunkt enden! Handelt es sich um ein schon vorhandenes logisches Gerät, so wird die Zuordnung geändert. Ansonsten definiert *assign* ein neues logisches Gerät, mit dem man sich leicht Tipparbeit ersparen kann.

Ein Beispiel:

```
assign akt: df0:markus/texte/neu
```

```
list akt:
```

ist eine praktische Befehlsfolge, wenn man Dateien aus dem Dateiverzeichnis *df0:markus/texte/neu* häufig braucht, aber dieses Verzeichnis nicht zum aktuellen Dateiverzeichnis machen kann.

cd

Format: *cd* <dir>

Zweck: dient zum Ändern und Abprüfen des aktuellen Dateiverzeichnisses.

Erläuterung:

Um dem Anwender Tipparbeit zu ersparen, kennt der Amiga das Konzept eines aktuellen Dateiverzeichnisses. Tippt man einen "einfachen" Dateinamen, so hängt das CLI intern immer den Namen des aktuellen Dateiverzeichnisses davor, um so den kompletten Namen der Datei zu bilden. Hat man beispielsweise eine Datei mit dem kompletten Namen **D1:alpha/text/brief1** und das aktuelle Dateiverzeichniss heißt **D1:alpha/text**, so braucht man nur **brief1** zu tippen, um die Datei genau zu bezeichnen. Der Befehl **cd** (für *change current directory* oder *ändere das aktuelle Dateiverzeichnis*) ändert den Namen, den das CLI als aktuelle Dateiverzeichnis auffaßt.

Tippt man nur den Befehl **cd** ohne Parameter, so wird der Name des aktuellen Dateiverzeichnisses angezeigt.

Tippt man **cd** und dahinter den Namen eines Dateiverzeichnisses, so wird dies das aktuelle Dateiverzeichnis. Dabei muß man beachten, daß das CLI vor diesen Namen normalerweise natürlich erst den Namen des jetzt noch aktuellen Dateiverzeichnisses setzt. Will man übergeordnete oder gar auf anderen Disketten liegende Dateiverzeichnisse aktuell machen, muß man deren Namen komplett angeben oder andere "Tricks" anwenden, die ausführlich im Kapitel *Dateien und Dateiverzeichnisse* erklärt sind.

copy

Format: *copy <name2> TO <name2> ALL QUIET*

Zweck: dient zum Kopieren einzelner Dateien und nahezu beliebiger Gruppen von Dateien.

Erläuterung:

Die meistverwendete Möglichkeit des *copy*-Befehls ist sicherlich, einzelne Dateien oder die Inhalte kompletter Dateiverzeichnisse zu kopieren.

Die Befehlsform *copy dateiname1 TO dateiname2* kopiert den kompletten Inhalt der Datei *dateiname1* in die Datei namens *dateiname2*. Existierte *dateiname2* schon, geht der alte Inhalt verloren! Existierte er noch nicht, so wird eine neue Datei dieses Namens angelegt. Dateinamen können dabei sowohl einfach sein (unter Nutzung des aktuellen Dateiverzeichnisses), als auch komplett angegeben werden. *Dateiname2* kann sinnvollerweise auch der Name eines Gerätes sein. Kopiert man beispielsweise eine Datei zum Gerät *PRT*: (dem Namen des Druckers), so bewirkt dies, daß die entsprechende Datei ausgedruckt wird.

Die Befehlsform *copy dir1 TO dir2* kopiert alle Dateien, die sich im Dateiverzeichnis *dir1* befinden, in das Dateiverzeichnis *dir2*. Damit dies funktioniert, muß das Verzeichnis *dir2* bereits existieren; es wird durch diesen Befehl auf keinen Fall automatisch angelegt. Bei dieser Befehlsform werden nur echte Dateien kopiert. Dateiverzeichnisse, die sich in *dir1* befinden, und die Dateien in diesen bleiben unberücksichtigt. Schreibt man ans Ende der Zeile aber noch das Schlüsselwort *ALL*, also *copy dir1 TO dir2 ALL*, werden auch alle darin enthaltenen Dateiverzeichnisse und Dateien mitkopiert.

Wenn auf diese Weise mehrere Dateien durch einen *copy*-Befehl kopiert werden, protokolliert der Befehl dies, indem er immer den Namen der gerade bearbeiteten Datei ins CLI-Fenster schreibt. Man hat so einen stetigen Überblick darüber, wie weit der Kopiervorgang schon ist. Selbst wenn der Vorgang wegen eines Fehlers (wenn z.B. die Zieldiskette voll ist) abbricht, weiß man genau, bei welcher Datei dies geschehen ist. Will man — aus welchen Gründen auch immer — dieses Protokoll nicht sehen, so kann man das Schlüsselwort *QUIET* (engl. für *still* oder *ruhig*) hinter den Befehl schreiben.

Eine letzte Möglichkeit, den *copy*-Befehl zu verwenden, besteht darin, statt dem ersten Dateiverzeichnis ein Muster anzugeben, in das die Dateinamen

passen müssen, um kopiert zu werden. Der Befehl ***copy #?.BAK TO df0:bak*** kopiert alle Dateien im aktuellen Dateiverzeichnis, die mit der Buchstabenkombination ***'BAK'*** enden in das Dateiverzeichnis ***bak***. Solche Muster können auch bei anderen Befehlen verwendet werden, die mehrere Dateien gleichzeitig verarbeiten können. Wie solche Muster genau aussehen können, wird im folgenden Abschnitt dieses Kapitels detailliert beschrieben.

date

Format: *date* <datum> <zeit>

Zweck: dient zum Anzeigen oder Ändern der vom System gespeicherten Zeit und des Datums.

Erläuterung:

Der Befehl *date* wird wohl am häufigsten dafür verwendet, die Systemzeit nachzustellen. In der Grundausbaustufe hat der Amiga ja keine eingebaute, batteriegespeiste Uhr — deshalb wird dieses Nachstellen nach jedem Einschalten wieder nötig. Beim Start liest der Amiga zunächst einmal das Datum und die Uhrzeit der letzten Modifikation an der Startdiskette und verwendet dieses als aktuelle Zeit. Wer Wert darauf legt, daß die Datumsangaben, die ihm z.B. bei Anwendung des *list*-Befehls (s.u.) gezeigt werden, wirklich stimmen, der sollte deshalb nach jedem Start die Uhr nachstellen. Die Verwendung des CLI-Befehls *date* ist meist wesentlich schneller als das Einstellen der Zeit mittels *Preferences*, wie es innerhalb der Workbench üblicherweise gemacht wird. Wer also üblicherweise das CLI benutzt, sollte nicht den Umweg über Preferences wählen.

Mittels *date* können sowohl Datum als auch Uhrzeit getrennt oder beide gleichzeitig eingestellt werden. So setzt *date 06-sep-86* das intern gespeicherte Datum auf den 6. September 1986. *Date 07:04* setzt die Uhrzeit auf 7 Uhr 4, wobei das Datum unverändert bleibt. Hingegen ändern *date 06-sep-86 07:04* oder *date 07:04 06-sep-86* Datum und Uhrzeit auf den 6. September 7 Uhr 4. Die Reihenfolge, in der Uhrzeit und Datum eingegeben werden, spielt keine Rolle. *Date* erkennt die Uhrzeit am Doppelpunkt, das Datum am Bindestrich.

Tippt man nur *date* ein, so wird die gespeicherte Uhrzeit und das Datum ausgegeben.

delete

Format: *delete* <namenliste> *ALL QUIET Q*

Zweck: dient zum Löschen einzelner Dateien und nahezu beliebiger Gruppen von Dateien.

Erläuterung:

Das Kommando *delete* ist dem *copy*-Befehl vom Format her recht ähnlich.

In der Form *delete datei1 datei2 datei3* kann man bis zu 10 Dateien angeben, die gelöscht werden sollen.

Mit der Form *delete dirname* können alle Dateien im Dateiverzeichnis *dirname* entfernt werden. Schreibt man ans Ende des Befehls noch das Schlüsselwort *ALL*, so werden auch alle Dateiverzeichnisse innerhalb von *dirname* und die Dateien darin gelöscht. Statt des Namens eines Dateiverzeichnisses kann auch ein Muster angegeben werden, in das die Namen der zu löschenden Dateien passen müssen. So löscht der Befehl *delete #?.BAK* z.B. alle Dateien im aktuellen Dateiverzeichnis, die mit der Buchstabenkombination ".BAK" enden. Solche Muster können auch bei anderen Befehlen verwendet werden, die mehrere Dateien gleichzeitig verarbeiten können. Wie solche Muster genau aussehen können, wird im folgenden Abschnitt dieses Kapitels detailliert beschrieben.

Wenn mehrere Dateien durch einen *delete*-Befehl gelöscht werden, protokolliert der Befehl dies, indem er immer den Namen der gerade bearbeiteten Datei ins CLI-Fenster schreibt. Will man — aus welchen Gründen auch immer — dieses Protokoll nicht sehen, so kann man das Schlüsselwort *QUIET* (engl. für *still* oder *ruhig*) oder nur das Kürzel *Q* hinter den Befehl schreiben.

dir

Format: *dir* <dirname> *OPT a d i*

Zweck: dient zum Anzeigen des Inhalts von ganzen Disketten und Dateiverzeichnissen.

Erläuterung:

Der Befehl **dir** ist die wichtigste Möglichkeit, mit der man einen Überblick über den Inhalt von Dateiverzeichnissen gewinnen kann. Detailliertere Informationen über einzelne Dateien lassen sich mit dem Befehl **list** gewinnen.

Gibt man den Befehl **dir** ohne den Namen eines Dateiverzeichnisses ein, so wird der Inhalt des aktuellen Dateiverzeichnisses aufgelistet. Folgt auf **dir** der Name eines Dateiverzeichnisses, so wird dessen Inhalt aufgelistet. Dabei werden stets zunächst die weiteren Dateiverzeichnisse, die sich darin befinden und dann die reinen Dateien in alphabetisch sortierter Form gezeigt. Die Liste der Dateiverzeichnisse ist einspaltig und hinter jedem Namen steht noch einmal in Klammern der Hinweis 'dir'. Die Auflistung der Dateien erfolgt zweispaltig.

Das Schlüsselwort **OPT** erlaubt eine weitere Modifikation des **dir**-Befehls. Hinter **OPT** können eine Reihe von Buchstaben folgen, die die gewählte Option bestimmen. Die Option **A** bewirkt, daß **dir** nicht nur die Namen aller untergeordneten Dateiverzeichnisse, sondern auch deren Inhalte anzeigt. Jede der so entstehenden Listen ist genauso aufgebaut, wie bei einem "einfachen" **dir**-Befehl, nur ist sie etwas nach rechts eingerückt. Der Befehl **dir df0: OPT a** gibt Ihnen also eine Liste aller auf der Diskette im eingebauten Laufwerk befindlichen Dateien aus. Die Option **D** bewirkt, daß nur Dateiverzeichnisse angezeigt werden. Die Option **I** erlaubt eine "interaktive Auflistung" von Dateiverzeichnissen, die im folgenden noch näher erklärt wird.

Wird bei einem **dir**-Befehl die **I**-Option (für interaktiv) gewählt, so wird dem Benutzer jeder Name eines Dateiverzeichnisses oder einer Datei einzeln gezeigt und dann ein Fragezeichen ausgegeben. Der Benutzer kann nun durch die Eingabe eines Buchstabens bestimmen, wie weiter vorzugehen ist:

Drücken Sie nur die <**RETURN**>-Taste, so wird Ihnen der nächste Name gezeigt. Tippen Sie **Q**, so wird der **dir**-Befehl abgebrochen. Wird der Name eines Dateiverzeichnisses angezeigt und Sie tippen **E** (für *enter* oder auf deutsch *einsteigen*), so werden der Inhalt dieses Dateiverzeichnisses

bearbeitet. Befinden Sie sich in einem untergeordneten Dateiverzeichnis und geben Sie **B** (für *back* bzw. *zurück*) ein, so wird im darüberliegenden Dateiverzeichnis an der Stelle weitergemacht, wo der **E**-Befehl gegeben wurde. Mit **E** und **B** kann man sich auf diese Weise durch den Baum der Dateiverzeichnisse "hangeln". Tippt man hinter dem Fragezeichen **DEL** (nicht die -Taste sondern die Buchstaben D, E und L), so wird die gerade angezeigte Datei gelöscht. Auch Dateiverzeichnisse können auf diese Weise gelöscht werden, aber nur, wenn sie zuvor vollständig geleert wurden (durch Löschen aller Dateien und Dateiverzeichnisse in ihnen). Wird der Name einer Datei angezeigt und man tippt **T**, so wird der Inhalt dieser Datei im CLI-Fenster ausgegeben.⁵ Hat man genug vom Dateiinhalte gesehen und möchte die Ausgabe deshalb abbrechen, so muß man die <CTRL>-Taste festhalten und gleichzeitig **c** tippen.

Sind Sie sich während eines interaktiven **dir**-Befehls jemals im Unklaren darüber, welche Möglichkeiten Sie gerade haben, tippen Sie hinter dem Fragezeichen einfach noch ein Fragezeichen. Eine Liste der im Augenblick möglichen Eingaben wird Ihnen dann gezeigt.

⁵ Dies ist natürlich nur dann sinnvoll, wenn es sich beim Inhalt dieser Datei um einen Text handelt. Ansonsten können unvorhersehbare Schwierigkeiten auftauchen.

diskcopy

Format: *diskcopy* <diskname1> *TO* <diskname2>

Zweck: dient zum Kopieren kompletter 3,5-Zoll-Disketten (für allem für Sicherungskopien).

Erläuterung:

Der Befehl *diskcopy* legt eine komplette Kopie einer Diskette auf einer anderen Diskette an. Der alte Inhalt der zweiten Diskette wird dabei vollständig zerstört. Dies ist besonders für das regelmäßige Anlegen von Sicherungskopien sehr sinnvoll. Bei Anwendung von *diskcopy* ist ein vorheriges Formatieren der Diskette (z.B. mit dem CLI-Befehl *format*) nicht nötig! Für die Kopie kann deshalb sowohl eine schon verwendete wie auch eine ganz neue, unbenutzte Diskette verwendet werden.

Genau wie der entsprechende Befehl in der Workbench kann auch *diskcopy* den Kopiervorgang entweder mit einem oder zwei Diskettenlaufwerken bewältigen.

So kopiert *diskcopy df0: TO df1:* z.B. den kompletten Inhalt der Diskette, die sich in diesem Moment im internen Laufwerk befindet, auf die Diskette, die sich in diesem Moment im externen Laufwerk befindet. Dies dürfte die übliche Vorgehensweise sein, wenn Sie stolzer Besitzer eines externen Laufwerks sind. Bevor der Kopiervorgang wirklich beginnt, fordert der Befehl aber noch einmal dazu auf, die korrekten Disketten in beide Laufwerke einzulegen und dann erst <RETURN> zu drücken.

Besitzen Sie nur ein Diskettenlaufwerk (das eingebaute), so wird die übliche Anwendungsform des Befehls für Sie wohl *diskcopy df0: TO df0:* sein. Genau wie beim Diskettenkopieren mit nur einem Laufwerk in der Workbench werden Sie nun immer wieder im Wechsel aufgefordert, die Original-Diskette oder die Diskette, die die Kopie aufnehmen soll, einzulegen. Eine gute Beschriftung beider Disketten ist für diesen Fall sehr wichtig und erleichtert Ihnen die Arbeit.

Egal, ob Sie ein oder zwei Diskettenlaufwerke besitzen, ist es in jedem Fall wichtig, Vorkehrungen zu treffen, damit nicht die Original-Diskette durch einen Bedienungsfehler zerstört wird. Der Diskcopy-Befehl ist weitgehend gegen solche Bedienungsfehler geschützt aber sie wissen ja: "Sicher ist sicher !" Deshalb schützen Sie bitte die Original-Diskette vor einem Überschreiben, indem Sie das Schreibschutzplättchen in eine Position bringen, bei der ein kleines Loch

entsteht. (Genaue Angaben dazu finden Sie im ersten Teil dieses Buches.) Die Diskette, die die Kopie aufnehmen soll, darf natürlich nicht schreibgeschützt sein.

Die auf einem dieser beiden Wege erzeugte Kopie wird für Sie nicht vom Original zu unterscheiden sein.⁶ Sie enthält alle Daten des Originals und trägt auch denselben Namen. Falls die Original-Diskette irgendwann einmal fehlerhaft wird, können Sie getrost die Kopie verwenden. Dies gilt leider nur in dem Fall, in dem der Hersteller der Programme, die sich eventuell auf dem Original befinden, diese nicht durch besondere Tricks vor einem Kopieren geschützt hat. Diese Unsitte der Software-Hersteller ist bei anderen Computern schon weit verbreitet und wird Ihnen gewiß auch auf dem Amiga bald das Leben unnötig schwer machen.

⁶ Das Betriebssystem kann die beiden Disketten allerdings als unterschiedlich erkennen. Die Zeit der letzten Schreiboperation wird nämlich auf der Kopie vermerkt und kann zur eindeutigen Identifikation einer Diskette verwendet werden.

echo

Format: *echo "beliebiger Text"*

Zweck: dient zum Ausgeben von Texten im CLI-Fenster (besonders in Exec-Dateien).

Erläuterung:

Der Text, der hinter *echo* (in Anführungsstrichen) angegeben ist, wird als Folge des Befehls im CLI-Fenster ausgegeben. Dies ist im allgemeinen nur sinnvoll in Exec-Dateien (Kommandofolgen) und bei Befehlen, die im Hintergrund ausgeführt werden.⁷ Man kann den Text allerdings auch auf diese Weise zu einem Gerät schicken. Die Befehlsfolge *echo >PRT: "Hallo Drucker !"* gibt den Text "Hallo Drucker !" z.B. auf einem eventuell angeschlossenen Drucker aus.

⁷ Vergleiche für mehr Informationen die Beschreibung des Befehls *execute* und *run*, sowie das nächste Kapitel.

ed

Format: *ed* <dateiname> *SIZE* <n>

Zweck: dient zum Anlegen und Modifizieren von Dateien, die druckbare Texte enthalten.

Erläuterung:

Gibt man den Befehl *ed*, so gelangt man in in einen einfachen "Editor". Ich verwende das Wort "Editor" und nicht etwa "Textverarbeitungsprogramm", um dieses einfache Programm von komfortableren Werkzeugen — wie z.B. *TextCraft* — abzusetzen. Der Editor *ed* kennt keine verschiedenen Zeichensätze, kennt keine Möglichkeit zum Unterstreichen oder Fettdrucken von Texten oder für die seitenweise Bearbeitung von Texten. Er ist nur für die Bearbeitung von reinen Textdateien geeignet. Diese werden z.B. typischerweise für die Programmierung oder auch als Exec-Dateien (Kommandofolgen) benötigt. Eine genaue Beschreibung von *ed* ist im Kapitel *Der Texteditor "ed"* zu finden.

Gibt man den Befehl *ed*, und dahinter den Namen einer schon existierenden Datei, so wird das Programm *ed* gestartet und die Datei eingelesen, damit sofort mit ihrer Bearbeitung begonnen werden kann. Existiert diese Datei noch nicht, so wird eine neue Textdatei dieses Namens angelegt.

Normalerweise können mit *ed* Texte bis zu einer maximalen Größe von weniger als 40 000 Anschlägen (ca. 20 Schreibmaschinenseiten) bearbeitet werden. Dies reicht für die einfachen Anwendungen, für die *ed* gedacht ist, üblicherweise völlig aus. Benötigt man mehr Platz, so kann man hinter den Namen der zu bearbeitenden Datei noch das Schlüsselwort *SIZE* und die Anzahl der Anschläge schreiben, mit denen *ed* höchstens "fertig werden" muß.⁸

⁸ Hier kann man natürlich auch kleinere Zahlen als 40 000 eingeben, wenn die zu bearbeitende Datei sehr klein ist und der Platz im Speicher des Amiga sehr eng ist, da noch andere Programme (Tasks) gleichzeitig aktiv sind.

edit

Format: *edit* <dateiname1> TO <dateiname2>

Zweck: dient zum Anlegen und Modifizieren von Dateien, die druckbare Texte enthalten.

Erläuterung:

Der Befehl *edit* aktiviert einen sehr altmodischen, aber auch sehr leistungsfähigen Texteditor.⁹ Sie werden ihn im Normalfall nie benötigen und den anderen Editor *ed* auch als wesentlich angenehmer empfinden. Benötigen Sie jedoch die Möglichkeiten von *edit*, so finden Sie eine ausführliche Beschreibung im *AmigaDOS User's Manual*.

⁹ Es handelt sich dabei um einen sogenannten "Zeileneditor", wie sie früher auf wesentlich primitiveren Computern weit verbreitet waren. Sie sind unhandlich zu bedienen und erlauben immer nur die Bearbeitung einer aktuellen Zeile. Der wesentliche Vorteil von *edit* gegenüber *ed* liegt allerdings darin, daß *ed* über die Tastatur bedient werden muß und die Bearbeitung eines Textes deshalb nicht mit einer Exec-Datei automatisiert werden kann. *Edit* kann seine Eingaben auch von einer Textdatei empfangen und so automatisch bestimmte Operationen mit einer großen Datei durchführen.

endcli

Format: *endcli*

Zweck: dient zum Beenden eines aktiven CLI-Tasks.

Erläuterung:

Der Befehl *endcli* beendet den CLI-Task, in dem der Befehl gegeben wurde. Das entsprechende Fenster wird geschlossen.

Mit diesem Befehl sollte man sehr vorsichtig umgehen! Er sollte nur erteilt werden, wenn "im Hintergrund" hinter dem CLI-Fenster noch die Workbench sichtbar ist oder noch ein oder mehrere andere CLI-Fenster offen sind. Ist das nicht der Fall, "hängt" der Amiga — er nimmt dann keinerlei Befehle über die Tastatur oder die Maus mehr an.

execute

Format: *execute* <kommandofolge> <argumentliste>

Zweck: dient zum Ausführen einer Liste von CLI-Kommandos, die in einer Textdatei gespeichert sind.

Erläuterung:

Der Befehl *execute* dient zum Automatisieren des Amiga. Diese Möglichkeit, längere Aufgabenfolgen ohne manuelles Zutun ablaufen zu lassen, ist einer der wesentlichen Vorteile des CLI gegenüber der Workbench. Hierzu werden die CLI-Befehle, die nacheinander ausgeführt werden sollen, in einer Textdatei gespeichert. Die Ausführung aller dieser Befehle kann mit *execute dateiname* "angestoßen" werden. Dabei ist *dateiname* natürlich der Name der Textdatei, die die CLI-Befehle enthält.

Die Möglichkeiten von *execute* werden ausführlich im Kapitel *Die Automatisierung des Amiga* erläutert.

failat

Format: *failat* <n>

Zweck: dient zum Ändern des Abbruch-Levels, welches erreicht werden muß, bis eine Kommandofolge abgebrochen wird.

Erläuterung:

Der Befehl *execute* führt eine Folge von CLI-Befehlen aus, die in einer Textdatei gespeichert sind. Tritt bei der Ausführung eines der gespeicherten Befehle ein Fehler auf, so bricht die Ausführung der gesamten Folge sofort ab; die restlichen Befehle werden nicht mehr aufgerufen. Dies ist nicht immer wünschenswert. Manchmal möchte man, daß nur bei sehr schwerwiegenden Fehlern abgebrochen wird oder man möchte den Fehler selbst erkennen und innerhalb der Kommandofolge darauf reagieren. In diesem Fall muß man den Abbruch-Level, ab dem Kommandofolgen bei einem Fehler abgebrochen werden, heraufsetzen.

Hierzu dient der Befehl *failat*, der diesen Level auf die Zahl setzt, die dahinter angegeben wird. Der Befehl *failat 20* setzt den Abbruch-Level z.B. auf 20 hoch, so daß nur noch relativ schwere Fehler die aktuelle Kommandofolge unterbrechen können. Auf Wunsch können sogar noch höhere Werte verwendet werden. Das Amiga-Betriebssystem unterscheidet grob zwischen drei Levels:

- 5 für Warnungen (außergewöhnliche Situation aber kein Fehler)
- 10 für normale Fehler
- 20 und größer für besonders schwerwiegende Fehler, die nur vom Benutzer behoben werden können. (Der normale Level ist 10.)

Mehr dazu aber im Kapitel *Die Automatisierung des Amiga* .

filenote

Format: *filenote* <dateiname> COMMENT <Kommentar>

Zweck: dient zum Anlegen und Ändern eines Kommentars, der mit einer bestimmten Datei verbunden ist.

Erläuterung:

Das Betriebssystem des Amiga kann zu jeder Datei einen kurzen Kommentar von maximal 80 Anschlägen verwalten. Dieser Kommentar kann vom Benutzer völlig frei gewählt werden. Er wird im allgemeinen zusätzliche Informationen über den Inhalt der Datei enthalten, die über das hinausgehen, was man in einem Dateinamen unterbringen kann. So könnte man z.B. vermerken, daß es sich bei einer bestimmten Datei um die 3. Version handelt und ähnliches. Der Kommentar zu einer Datei kann auf Wunsch mit dem CLI-Befehl *list* (s.u.) betrachtet, jedoch nicht geändert werden, mit dem Befehl *filenote* geändert, aber nicht betrachtet werden, und in der Workbench mit dem Menübefehl *Info* sowohl betrachtet als auch geändert werden.

Die Anwendung von *filenote* ist simpel. Man tippt *filenote* und dann den Namen der Datei, deren Kommentar geändert werden soll und dahinter den neuen Kommentar. Handelt es sich nur um ein Wort, kann man es einfach eintippen. Handelt es sich um mehrere Worte, die durch Zwischenräume und/oder Sonderzeichen getrennt sind, muß man den Kommentar in Anführungszeichen setzen.

Der Befehl *filenote* hat einige merkwürdige Eigenschaften. Eine neue Datei hat zunächst keinen Kommentar. Wird aber — z.B. beim Kopieren — eine alte Datei mit einem neuen Inhalt überschrieben, so bleibt der alte Kommentar dieser Datei erhalten (der ja wahrscheinlich jetzt nicht mehr zutrifft). Es wird also nicht der Kommentar der Datei verwendet, die kopiert wurde, sondern der Kommentar der Datei, die durch den Kopiervorgang zerstört wurde.

format

Format: *format DRIVE <drivename> NAME <diskname>*

Zweck: dient zum Formatieren und Initialisieren einer unbenutzten oder zum Löschen einer alten Diskette.

Erläuterung:

Bevor eine Diskette oder eine Festplatte verwendet werden kann (bevor Dateiverzeichnisse und Dateien darauf angelegt werden können), muß sie formatiert werden. Bei diesem Vorgang wird der amorphen Struktur des Speichermediums, das ja nur eine Fläche magnetisierbaren Materials ist, ein Format "aufgezwungen". Die Diskette besteht danach aus einer Folge von konzentrischen Spuren (engl. *tracks*), und jede dieser Spuren besteht aus einer bestimmten Anzahl von Blöcken oder Sektoren (engl. *sectors*).¹⁰ Nach dieser rein magnetischen Formatierung muß auch noch das zunächst leere Dateiverzeichnis angelegt und initialisiert werden, das die Wurzel der Datei-Hierarchie auf dieser Diskette oder Festplatte bildet.

Diese beiden Aufgaben erfüllt für 3,5-Zoll-Disketten der CLI-Befehl *format*. Seine Anwendung ist recht unmißverständlich. Er benötigt zwei Parameter, die beide von einem Schlüsselwort eingeleitet werden. Der erste gibt an, in welchem Diskettenlaufwerk sich die zu formatierende Diskette befindet (i.a. wohl *df0:* oder *df1:*), der zweite gibt den Namen an, den die frisch formatierte Diskette dann tragen soll.

Wichtig beim Befehl *format* ist, daß er nur 3,5-Zoll-Disketten formatiert. Für 5,25-Zoll Disketten oder Festplatten — sofern Sie solche besitzen — müssen Sie einen anderen Befehl verwenden, der ihnen hoffentlich beim Erwerb dieser Zusatzgeräte mitgeliefert wird.

Format ist nur dann nötig, wenn Sie eine leere Diskette benötigen, um auf ihr einzelne neue Dateien anzulegen. Wollen Sie eine Diskette kopieren (mit *diskcopy*), so brauchen sie keine neue Diskette, die die Kopie aufnehmen soll, zu formatieren. Das erledigt *diskcopy* gleich mit!

¹⁰ Genaue Angaben über das Format sind im *AmigaDOS Technical Reference Manual* zu finden, das Ihnen Ihr Amiga-Händler besorgen kann.

if

Format: *if* <bedingung>

Zweck: dient zum bedingten Ausführen von CLI-Befehlen innerhalb von Kommandofolgen.

Erläuterung:

Den Befehl *if* werden Sie nie direkt dem CLI erteilen. Er ist nur in Kommandofolgen sinnvoll. Hinter *if* folgt immer eine Bedingung, die das CLI überprüft, wenn es diesen Befehl erhält. Trifft die Bedingung zu, so werden die folgenden Befehle in der Kommandofolge bearbeitet, bis der Befehl *else* oder der Befehl *endif* aufgerufen wird. Trifft die Bedingung nicht zu, sucht das CLI den Befehl *else* oder *endif* und fährt hinter diesem mit der Ausführung der Befehle in der Kommandofolge fort.

Mit den Bedingungen (Schlüsselworten) *WARN*, *ERROR* und *FAIL* kann der Fehler-Level abgeprüft werden (siehe *failat*). *WARN* trifft zu, wenn dieser Level ≥ 5 ist, *ERROR*, wenn er ≥ 10 ist und *FAIL*, wenn er ≥ 20 ist. Mit diesen Bedingungen können Kommandofolgen vom Erfolg vorangehender Befehle abhängig gemacht werden.¹¹

Die Bedingung *EXISTS* <dateiname> trifft zu, wenn eine Datei des angegebenen Namens existiert. Die Bedingung <name1> *EQ* <name1> trifft dann zu, wenn die beiden Texte gleich sind. Unterschiede zwischen Groß- und Kleinschreibung werden bei diesem Vergleich ignoriert. Setzt man schließlich das Schlüsselwort *NOT* vor eine der beschriebenen Bedingungen, so wird deren Ergebnis umgekehrt.

Mehr dazu aber im Kapitel *Die Automatisierung des Amiga*.

¹¹ Damit solche Bedingungen sinnvoll sind, muß unbedingt der Fehler-Level mit *failat* hochgesetzt werden. Ist er z.B. kleiner als 10, so wird das CLI nie bis zu einer Bedingung *FAIL* kommen, weil die Ausführung der Kommandofolge schon vorher abgebrochen wird.

info

Format: *info*

Zweck: dient zur Information über den Zustand des gesamten Dateisystems.

Erläuterung:

Der Befehl *info* gibt eine Liste aus, in der für jede im Moment vorhandene Diskette, Festplatte usw. die folgenden Informationen ausgegeben werden:

- das Laufwerk, in dem sich die Diskette/Festplatte befindet,
- die Größe der Diskette (in KByte)
- wieviel KByte darauf belegt sind,
- wieviel KByte darauf frei sind,
- welcher Prozentsatz belegt ist,
- natürlich der Name der Diskette/Festplatte und
- einige weitere technische Informationen

install

Format: *install* <drivename>

Zweck: macht eine formatierte Diskette zur startfähigen Diskette.

Erläuterung:

Der Befehl *install* ist nur für fortgeschrittene Anwender gedacht. Mit ihm werden die Informationen auf eine Diskette geschrieben, die der Amiga zum starten benötigt, nachdem er die Kickstart-Diskette schon gelesen hat. Falls nicht noch weitere Vorkehrungen getroffen werden, ist die so entstehende Diskette aber reichlich unbrauchbar. Bevor Sie den Befehl anwenden, sollten Sie mehr über die Dateien wissen, die Sie für die verschiedenen Zwecke benötigen, als Ihnen dieses Buch vermitteln kann. Vom (vorsichtigen) Experimentieren kann und soll durch diese warnenden Worte aber niemand abgehalten werden.

join

Format: *join* <namensliste> *AS* <neuename>

Zweck: verbindet bis zu 15 alte Dateien zu einer neuen Datei.

Erläuterung:

Der Befehl *join* (engl. für *verbinden*, *verknüpfen*) dient zur Verknüpfung von Dateien. Dies kann aus den verschiedensten Gründen heraus wichtig sein. Eine solche Verknüpfung funktioniert aber nur bei Dateien, die nur aus einer Folge gleichförmiger Daten bestehen, z.B. Textdateien. Mit *join* kann man aus mehreren kurzen Text-Dateien eine lange machen. Bei Dateien mit komplexem Format, wie z.B. GrafiCraft-Bildern, ist ein *join* meist nicht möglich.

Um *N* Dateien (maximal 15) zu einer neuen zu verbinden, tippt man einfach *join*, danach die *N* Dateinamen und schließlich *AS* und einen neuen Namen. Der Befehl *join text1 brief AS neutext* verbindet z.B. die beiden Dateien namens *text1* und *brief* zu einer neuen Datei namens *neutext*..

Beachten Sie bitte, daß die neue Datei nicht denselben Namen tragen kann, wie eine der darin zu verschmelzenden Dateien!

lab

Format: *lab* <text>

Zweck: setzt eine Marke in einer Kommandofolge.

Erläuterung:

Der Befehl **lab** markiert die Zeile einer Kommandofolge, an der er steht, als das mögliche Ziel eines entsprechenden **skip**-Kommandos. Der Befehl **skip** **marke** in einer Kommandofolge bewirkt so z.B., daß das CLI alle Befehle zwischen diesem und einem späteren **lab** **marke** ignoriert bzw. überspringt.

Mehr dazu aber im Kapitel *Die Automatisierung des Amiga* .

list

Format: *list* <name> *P* <muster> *KEYS* *NODATES*
S <text> *SINCE* <datum> *UPTO* <datum> *QUICK*

Zweck: gibt eine Menge von Informationen über eine Datei oder eine Gruppe von Dateien aus.

Erläuterung:

Der Befehl *list* kann sehr ausführliche Informationen über eine Datei anzeigen. Dazu gehören Name und Größe (in Bytes) der Datei, eventuelle Schreib-/Lese-Schutzflags, das Datum und die Uhrzeit der letzten Modifikation und schließlich der Kommentar (s. den CLI-Befehl *filenote*). Verschiedene Optionen steuern, welche Informationen ausgegeben werden und für welche Dateien sie ausgegeben werden.

Gibt man *list* ohne den ersten Parameter *name* an, so werden Informationen über alle Dateien im aktuellen Dateiverzeichnis aufgelistet (sofern nicht andere Optionen diese Menge einschränken). Ist *name* ein Dateiname, so werden nur Informationen über diese Datei ausgegeben. Ist *name* der Name eines Dateiverzeichnisses, so werden Informationen über alle Dateien dieses Dateiverzeichnisses aufgelistet (sofern nicht andere Optionen diese Menge einschränken).

Gibt man das Schlüsselwort *KEYS* mit an, so erhält man eine technische Information¹² zu jeder Datei, mit der man defekte Dateien eventuell wiederherstellen kann. Das Schlüsselwort *NODATES* unterdrückt die Ausgabe mit Datums- oder Uhrzeit-Informationen zur jeweiligen Datei. Das Schlüsselwort *QUICK* bewirkt, daß nur noch die Namen der Dateien und Dateiverzeichnisse ausgegeben werden und reduziert deshalb *list* weitgehend auf die Funktion, die auch der CLI-Befehl *dir* erfüllt.

Mit den Schlüsselworten *P*, *S*, *SINCE* und *UPTO* kann die Menge der Dateien aus einem Dateiverzeichnis, über die Informationen ausgegeben werden, eingeschränkt werden. Gibt man *P* und ein *Muster* ein, so werden nur Dateien berücksichtigt, die in dieses Muster passen.¹³ Gibt man *S* und einen Text an, so werden nur Dateien berücksichtigt, die in ihrem Namen den

¹² Es handelt sich dabei um die Nummer des *file-* oder *directory-header blocks*, dessen Bedeutung im *AmigaDOS Technical Reference Manual* näher erläutert wird.

¹³ Weitergehende Informationen über Muster sind im nächsten Abschnitt dieses Kapitels zu finden.

bewußten Text enthalten. **SINCE** und eine Datumsangabe bewirken, daß nur Dateien aufgelistet werden, die seit diesem Datum (engl. *since*) noch modifiziert wurden. **UPTO** und eine Datumsangabe schränkt die Ausgabe auf Dateien ein, die spätestens bis (engl. *upto*) diesem Datum noch modifiziert wurden und danach nicht mehr.

Der Befehl **list** kennt so viele Optionen, daß Sie am besten ein wenig damit experimentieren sollten. Vergessen Sie auch nicht, daß Sie die Ausgabe, also die Informationen über die Dateien, auch (mit **>**) auf eine Datei schicken und sie dort in Ruhe betrachten können (z.B. mit **ed**).

loadwb

Format: *loadwb*

Zweck: Startet die Workbench, falls sie noch nicht aktiv ist.

Erläuterung:

Unter normalen Umständen wird beim Starten des Amiga die Workbench aktiv und bleibt immer aktiv, auch während andere Programme gestartet werden und laufen. Man kann diesen Start der Workbench jedoch verhindern und beim Start gleich in das CLI gelangen. Möchte man dann trotzdem wieder in der Workbench arbeiten, so dient dazu der Befehl *loadwb*.

Mehr dazu aber im Kapitel *Tips für das CLI*.

makedir

Format: *makedir* <dir>

Zweck: dient zum Anlegen eines neuen, untergeordneten Dateiverzeichnisses.

Erläuterung:

Der Befehl *makedir* ist immer dann nötig, wenn Sie Dateien in ein neues Dateiverzeichnis legen wollen. Vorher muß dieses neue Dateiverzeichnis (engl. *directory*) explizit mit *makedir* angelegt werden. Dies ist eine Sache die Sie nicht oft genug machen können, um ihre Dateien gut gegliedert zu halten.¹⁴

makedir NeuVerz legt ein neues Dateiverzeichnis namens *NeuVerz* an, das dem aktuellen Dateiverzeichnis untergeordnet ist. Wollen Sie ein neues Dateiverzeichnis in einem anderen als dem aktuellen Dateiverzeichnis anlegen, so können Sie dafür die selben Methoden anwenden wie beim Identifizieren von Dateien in "fremden" Dateiverzeichnissen (vollständige Namen usw.).

Gelöscht werden können Dateiverzeichnisse genau wie Dateien mit dem Befehl *delete*. Sie müssen zuvor aber vollständig leer sein!

¹⁴ Ausführliche Informationen zu Dateiverzeichnissen und deren Nutzen und Philosophie finden Sie im Kapitel *Dateien und Dateiverzeichnisse*.

newcli

Format: *newcli* <fenster>

Zweck: dient zum Erzeugen eines neuen CLI-Tasks in einem neuen Fenster.

Erläuterung:

Der Befehl *newcli* startet, sofern noch genügend Speicherplatz frei ist — einen neuen Task, in dem auch das CLI läuft. Dieser Task erhält ein eigenes Fenster, in dem sich die Kommunikation mit dem Benutzer abspielt.

Tippt man *newcli* ohne Parameter ein, so erhält man ein Standardfenster für die Kommunikation mit dem neuen CLI-Task. Dieses Fenster kann danach beliebig verschoben, vergrößert und verkleinert werden. Möchte man aber sofort ein Fenster passender Größe und Position haben, so kann man dies dem Befehl durch einen Parameter mitteilen. Dieser hat die Form *CON:<x>/<y>/<breite>/<höhe>/<titel>*.¹⁵ Er enthält im wesentlichen (durch Schrägstriche getrennt), den Abstand vom oberen Bildschirmrand, vom unteren Bildschirmrand, die Breite und die Höhe des Fensters und dessen Titel. Enthält der Titel Leerzeichen oder Sonderzeichen, sollte der ganze Parameter in Anführungszeichen gesetzt werden.

Die Zeile *newcli "CON:10/30/600/100/Neues CLI"* erzeugt zum Beispiel einen neuen CLI-Task, dessen Fenster etwa das obere Bildschirm Drittel einnimmt und den Namen *Neues CLI* in seiner Titelleiste trägt.

¹⁵ *CON:* teilt dem Betriebssystem des Amiga mit, daß sich das Gerät *CON:* (Tastatur und Bildschirm) um dieses Fenster kümmern soll. Ein anderes Gerät kann sich sinnvollerweise aber kaum um ein CLI-Fenster kümmern und sollte hier deshalb nicht angegeben werden.

prompt

Format: *prompt* <text>

Zweck: dient zum Ändern der Meldung, mit der das CLI dem Benutzer mitteilt, daß es bereit ist, den nächsten Befehl entgegenzunehmen.

Erläuterung:

"Normalerweise" ist der Prompt, mit dem das CLI dem Benutzer mitteilt, daß es bereit ist, den nächsten Befehl entgegen zu nehmen, eine Zahl und das Größer-Zeichen >. Die Zahl dient dazu, verschiedenen CLI-Fenster zu unterscheiden, sofern man mehrere davon hat. Sie ist gleich der laufenden Nummer des CLI-Tasks, weswegen man vom allerersten CLI immer mit **1>** begrüßt wird. Der Befehl *prompt* ändert diese Meldung.

So empfängt Sie das CLI nach dem Befehl *prompt* "**Deine Befehle Herr?**" mit der Frage **Deine Befehle Herr?** statt mit **1>**. Möchte man auch in seinen eigenen Prompts die Task-Nummer sehen, so kann man dazu den Platzhalter *%N* verwenden. Dieser Platzhalter wird vor der Anzeige des Textes durch das CLI gegen die laufende Nummer des CLI-Tasks ersetzt. Nach dem Befehl *prompt* "**CLI-Nummer %N:**" meldet das CLI sich mit **CLI-Nummer 1:** wieder.

protect

Format: *protect* <dateiname> *FLAGS* <schalter>

Zweck: dient zum Ändern der *Sicherungs-Schalter* einer Datei.

Erläuterung:

Das Betriebssystem des Amiga verwaltet zu jeder Datei 4 Sicherungs-Schalter, die einen Mißbrauch der Datei möglichst schwierig machen. Jeder dieser Schalter kann AN oder AUS sein und erlaubt oder verhindert bestimmte Operationen mit der Datei. Die vier Schalter (engl *flags*) haben einbuchstabile Namen. Sie lauten:

- r:** die Datei kann gelesen werden, (r für *read*; zu deutsch *lesen*)
- w:** die Datei kann geändert werden, (w für *write*; zu deutsch *schreiben*)
- d:** die Datei kann gelöscht werden, (d für *delete*; zu deutsch *löschen*)
- e:** die Datei kann als Programm (Kommando) aufgerufen werden.

Alle vier Schalter sind bei einer neuen Datei zunächst einmal auf AN. Jede Datei kann also zu Beginn gelesen, geändert, gelöscht und ausgeführt werden. Mit *protect* werden nun alle Schalter auf AN gesetzt, deren Namen hinter dem Schlüsselwort *FLAGS* auftauchen. Der Befehl *protect* kann nur auf einzelne Dateien und leider nicht auf Gruppen von Dateien angewendet werden. Es werden immer nur die Schalter der Datei geändert, deren Name direkt hinter *protect* auftaucht.

So erklärt der Befehl *protect text1 FLAGS rw* die Datei *text1* als lesbar und änderbar. Sie kann aber nicht gelöscht oder als Programm behandelt werden. Hingegen schützt *protect prog de* die Datei *prog* vor dem Lesen und Ändern (und damit z.B. vor dem Kopieren). *Prog* kann aber gelöscht und als Programm aufgerufen werden.

Im Moment ignoriert das CLI und das Betriebssystem des Amiga noch alle Schalter außer *d* (dem Lösch-Schutz)! Spätere Versionen unterstützen hoffentlich aber auch die anderen Schalter.

quit

Format: *quit* <Fehler-Level>

Zweck: dient zum Beenden einer Kommandofolge und der eventuellen Rückmeldung eines aufgetretenen Fehlers.

Erläuterung:

Den Befehl *quit* werden Sie nie direkt an das CLI erteilen. Er ist nur in Kommandofolgen sinnvoll. Trifft das CLI in einer Kommandofolge auf *quit*, dann wird diese Kommandofolge sofort beendet. Kommandofolgen werden "ordnungsgemäß" mit ihrer letzten Zeile beendet, deshalb ist *quit* als eine Art "Notausgang" zu bezeichnen. Aus demselben Grund kann *quit* auch einen Fehlercode zurückgeben, der z.B. dazu führen kann, daß eine andere Kommandofolge abgebrochen wird, die die aktuelle aufgerufen hat. Der Fehlercode kann aber auch mit einem *if*-Befehl abgefragt werden.

Mehr dazu aber im Kapitel *Die Automatisierung des Amiga*.

relabel

Format: *relabel* <laufwerksname> *NAME* <diskname>

Zweck: dient zum Umbenennen von Disketten.

Erläuterung:

Eine neue Diskette erhält ihren Namen üblicherweise bei der Formatierung oder als Kopie einer anderen Diskette. Die dafür "zuständigen" Befehle sind *format* und *diskcopy* bzw. ihre Äquivalente in der Workbench. Möchte man nachträglich den Namen einer Diskette ändern, so ist dies mit *relabel* möglich.

Der Befehl *relabel* benötigt zwei Parameter. Der erste ist der Name eines angeschlossenen Diskettenlaufwerks (meist wohl *df0:* oder *df1:*), in dem sich auch eine Diskette befinden sollte. Der zweite Parameter wird durch das Schlüsselwort *NAME* eingeleitet und ist der Name, den die Diskette im angesprochenen Laufwerk tragen soll. Falls dieser Name Leerzeichen enthält, muß er in Anführungszeichen gesetzt werden. Der Befehl *relabel df0: "Diskette X"* z.B. bewirkt, daß die Diskette im eingebauten Laufwerk des Amiga von nun an den Namen *Diskette X* trägt.

Dateien und Dateiverzeichnisse werden übrigens mit dem Befehl *rename* (s.u.) umbenannt.

rename

Format: *rename*<alter name> *TO*<neuer name>

Zweck: dient zum Umbenennen von Dateien und Dateiverzeichnissen.

Erläuterung:

Der Befehl *rename* gibt einer Datei oder einem Dateiverzeichnis einen neuen Namen. Er hat zwei Parameter: den alten und den neuen Namen der Datei/des Dateiverzeichnisses. Der zweite Parameter kann der besseren Lesbarkeit wegen vom Schlüsselwort *TO* eingeleitet werden (muß es aber nicht).

Mit *rename* kann man Dateien nicht nur im strengen Sinne umbenennen, sondern sie auch von einem Dateiverzeichnis in ein anderes legen, sofern sich beide Dateiverzeichnisse auf derselben Diskette/Festplatte befinden. Hierzu muß man für die beiden Dateinamen nicht den einfachen Namen angeben, sondern den kompletten Namen, der auch die Dateiverzeichnisse enthält, die man durchlaufen muß, um diese Datei zu finden. Da dieser Pfad beim neuen Namen nicht der selbe ist, wird die Datei von einem in ein anderes Dateiverzeichnis bewegt. Das zweite Dateiverzeichnis muß allerdings vor Ausführung des Befehls schon existieren (siehe *makedir*).

Gibt man z.B. den Befehl *rename Text1 TO "alter Text"*, findet nur eine "einfache" Umbenennung der Datei *Text1* im aktuellen Dateiverzeichnis statt. Mit *rename df0:texte/text1 TO df0:alt/text1* wird die Datei *text1* hingegen vom Dateiverzeichnis *df0:texte* nach *df0:alt* bewegt.

Komplette Disketten oder Festplatten werden übrigens mit dem Befehl *relabel* (s.o.) umbenannt.

run

Format: *run* <bel. Befehlsfolge>

Zweck: dient zum Ausführen einer Gruppe von CLI-Befehlen als separater Task (*im Hintergrund*).

Erläuterung:

Der Befehl *run* erlaubt es, einen oder mehrere CLI-Befehle als separaten Task ablaufen zu lassen, ohne zuvor mit *newcli* einen neuen CLI-Task dafür erzeugen zu müssen. Sofort nach dem Befehl *run* kann man dem aktuellen CLI einen neuen Befehl erteilen, während nebenher ein anderer Befehl abläuft. Man nennt dies "Ausführen eines Befehls *im Hintergrund*". Die Bezeichnung *im Hintergrund* kommt daher, daß dieser neue Task nicht mit dem Benutzer kommunizieren kann, da er kein eigenes Fenster besitzt. Er bleibt also im Hintergrund Ihrer Aufmerksamkeit, während Sie sich vordringlich mit anderen Dingen beschäftigen. Nur Meldungen von diesem und anderen Tasks erscheinen in dem CLI-Fenster, von dem aus er gestartet wurde.¹⁶ Üblicherweise wird man deshalb wohl dafür einen CLI-Befehl verwenden, der längere Zeit zu seiner Ausführung benötigt und kein weiteres Eingreifen von Seiten des Benutzers nötig hat.

Der Befehl *run* kennt keine Parameter im eigentlichen Sinn des Wortes. Der Rest der Zeile hinter dem Wort *run* muß ein gültiger CLI-Befehl sein, der dann von *run* im Hintergrund ausgeführt wird. Dieser Befehl kann sich dabei durchaus über mehrere Zeilen ausdehnen, indem man die Möglichkeit der Zeilenfortführung mit Hilfe des Symbols *+* nutzt.

Der folgende Befehl schickt z.B. den Inhalt einer Datei auf den Drucker, nachdem er sie vorher kopiert hat, löscht die Kopie nach Beendigung des Druckvorgangs und meldet dann, daß er fertig ist:

```
1>run copy BriefRobert druckvorlage +
copy druckvorlage PRT: +
delete druckvorlage +
echo "Druckvorgang beendet !"
```

¹⁶ Dies kann dann verwirrend sein, wenn zwei oder mehr Tasks gleichzeitig ins Fenster schreiben. So können z.B. während der Ausführung eines *dir*-Befehls zwischen den aufgelisteten Dateinamen Ausgaben eines Hintergrund-Tasks auftauchen. Hat man gar mehrere Tasks im Hintergrund laufen, die Ausgaben ins CLI-Fenster schicken, ist das Chaos geradezu vorprogrammiert.

Achten Sie bei der Verkettung mehrerer Zeilen darauf, daß hinter dem Fortführungssymbol `+` sofort `<RETURN>` getippt werden muß, ansonsten gilt die folgende Zeile als neue Zeile und nicht als Fortführung der vorangehenden.

say

Format: *say* <text>

Zweck: dient zur Ausgabe eines (meist kurzen) Textes aus dem Lautsprecher.

Erläuterung:

Der Befehl *say* nutzt die Fähigkeit des Amiga zur Sprachsynthese. Er gibt den Text, der als Parameter übergeben wurde, über den oder die angeschlossenen Lautsprecher aus.¹⁷ Die Aussprache ist nicht gerade natürlich zu nennen, aber doch verständlich.

Eine Verwendung von *say* bietet sich vor allem innerhalb von Kommandofolgen (s.u.) an. So könnte die Beendigung einer Kommandofolge z.B. statt mit einer Bildschirmmeldung, die man vielleicht übersehen würde, durch ein lautes "Fertig!" kund getan werden.

¹⁷ Der Amiga besitzt keinen eingebauten Lautsprecher. Wenn kein Lautsprecher an einen oder beide der Stereo-Ausgänge hinten am Gehäuse angeschlossen ist, sind solche Meldungen deshalb nicht zu hören.

search

Format: *search* <name> *SEARCH* <text> *ALL*

Zweck: dient zur Suche nach einem bestimmten Text in einer beliebigen Gruppe von Dateien.

Erläuterung:

Falls Sie einmal in die Verlegenheit kommen, nicht mehr genau zu wissen, in welchen Dateien Sie sich auf bestimmte Begriffe oder Namen bezogen haben, so hilft Ihnen der *search*-Befehl.¹⁸ Er erlaubt es, eine Gruppe von Dateien nach dem Text zu durchsuchen, der hinter dem Schlüsselwort *SEARCH* steht. (Falls dieser Text Leerzeichen enthält muß er in Anführungszeichen gesetzt werden.) Jede Zeile in jeder durchsuchten Datei, die diesen Text enthält, wird ausgegeben. (Diese Liste kann natürlich mittels > auch in eine Datei statt auf den Bildschirm geschickt werden.) Beim Vergleich des Suchtextes mit dem Inhalt der Datei wird Groß- oder Kleinschreibung nicht berücksichtigt.¹⁹

Geben Sie für <name> den Namen eines Dateiverzeichnisses ein, so werden alle Dateien in diesem Dateiverzeichnis durchsucht. Wenn Sie am Ende der Zeile noch das Schlüsselwort *ALL* folgen lassen, erstreckt sich die Suche auch auf alle Dateien in Dateiverzeichnissen, die dem durchsuchten hierarchisch untergeordnet sind (in diesem liegen). Statt ein Dateiverzeichnis durchsuchen zu lassen, können Sie die zu berücksichtigenden Dateien auch in Form eines *Musters* angeben. Statt des Namens eines Dateiverzeichnisses geben Sie dazu an der selben Stelle einfach das Muster an. Nur die Dateien, deren Namen in das Muster passen, werden dann durchsucht. Wird überhaupt kein Muster oder kein Dateiverzeichnisname angegeben, so wird das aktuelle Dateiverzeichnis durchsucht.

So werden z.B. durch den Befehl *search df0: SEARCH "Sehr geehrt"* alle Dateien auf der Diskette im eingebauten Laufwerk nach dem Text "Sehr geehrt" durchsucht. Der Befehl *search Brief#? SEARCH "Hallo!"* hingegen

¹⁸ Der *search*-Befehl ist vor allem für Programmierer gedacht, die in ihren Programmen nach bestimmten Dingen (Prozeduren, Variablen, etc.) suchen. Er kann unter Umständen auch für den "normalen" Anwender von Nutzen sein.

¹⁹ Mit Umlauten hat die Suchmethode allerdings Schwierigkeiten: ein großes Ö ist so z.B. verschieden von einem kleinen ö.

durchsucht alle Dateien im aktuellen Dateiverzeichnis, deren Name mit "Brief" beginnt, nach "Hallo!".

Beim Durchsuchen der verschiedenen Dateien haben Sie die Möglichkeit, die Suche in der aktuellen Datei oder auch komplett abbrechen zu lassen. Halten Sie die <CTRL>-Taste fest und tippen sie gleichzeitig **D**, so wird die Durchsuchung der gerade bearbeiteten Datei abgebrochen und gleich mit der nächsten fortgefahren. Halten Sie hingegen die <CTRL>-Taste fest und tippen gleichzeitig **C**, so wird die ganze Suche abgebrochen.

skip

Format: *skip* <marke>

Zweck: dient zum Überspringen einer Gruppe von CLI-Befehlen innerhalb einer Kommandofolge.

Erläuterung:

Den Befehl *skip* (engl. für *überspringen*, *übergehen*) werden Sie nie direkt an das CLI erteilen. Er ist nur in Kommandofolgen sinnvoll. Trifft das CLI in einer Kommandofolge auf *skip* und dahinter einen Namen, dann werden alle Befehle in dieser Kommandofolge übersprungen, bis der Befehl *lab* (s.o.) und dahinter der selbe Name erscheint. Der Befehl *skip* ist also eine sogenannte *Sprunganweisung* zu einer Marke, wie man bei Programmiersprachen üblicherweise sagt.

Mehr dazu aber im Kapitel *Die Automatisierung des Amiga*.

sort

Format: *sort* <dateiname1> *TO* <dateiname2> *COLSTART* <n>

Zweck: dient zum Sortieren einer Textdatei.

Erläuterung:

Mit dem Befehl *sort* (engl. für *sortieren*) können Sie eine Textdatei sortieren. Eine Textdatei zeichnet sich dadurch aus, daß Sie nur druckbare Zeichen enthält und in Zeilen aufgeteilt ist, die durch ein besonderes Zeichen (den unsichtbaren Zeilenvorschub) getrennt sind. Hierbei enthält die Datei mit dem Namen <dateiname1> den unsortierten Text und die Datei mit dem Namen <dateiname2> nach Ausführung des Befehls den sortierten Text. Die Sortierung geschieht alphabetisch, beginnend mit der ersten Spalte jeder Zeile, und ignoriert Unterschiede zwischen Groß- und Kleinschreibung. Wollen Sie bei der Sortierung erst den Text ab einer bestimmten Spalte in jeder Zeile berücksichtigen, so können Sie die Nummer dieser Zeile hinter dem Schlüsselwort *COLSTART* angeben.

Der Befehl *sort* ist ein sehr primitives Sortierprogramm, das nur für die wenigsten Anwendungsfälle geeignet sein dürfte. Normalerweise kann es nur Dateien sortieren, die maximal 200 Zeilen enthalten. Bei größeren Dateien müssen Sie den Speicher, der dem Programm für diese Zwecke zur Verfügung steht, mit dem Befehl *stack* erhöhen. Wie groß der Speicher für Dateien bestimmter Größe sein muß, ist dabei sehr stark Erfahrungssache. *Sort* ist deshalb eigentlich nur für erfahrene Benutzer und kleine Dateien geeignet!

stack

Format: *stack* <n>

Zweck: dient zum Anzeigen und Ändern der Stack-Größe.

Erläuterung:

Mit dem Befehl *stack* kann die Größe eines bestimmten Speicherbereichs (des sog. *Stacks*) geändert werden, die das CLI einem Programm (Befehl) zuteilt, bevor es gestartet wird. Dieser Speicherbereich dient im wesentlichen für den sich laufend ändernden (dynamischen) Speicherbedarf eines Programms.

Ohne tiefergehende Kenntnisse der Programmierung dürfte der *stack*-Befehl für Sie wenig brauchbar sein. Falls aber bei Anwendung des *sort*-Befehls einmal ein Fehler auftauchen sollte, so liegt dies wahrscheinlich an mangelndem Speicherplatz. Verdoppeln Sie dann versuchsweise diesen Speicherplatz mittels *stack*.

Tippen Sie *stack* ohne Parameter ein, so wird Ihnen die aktuelle Stackgröße angezeigt. Sie beträgt normalerweise 4000 (Byte). Tippen Sie hinter *stack* noch eine Zahl ein, so wird der für den Stack reservierte Platz auf diesem Wert gesetzt. **Vorsicht beim Verkleinern des Wertes unter 4000!** Viele Programme benötigen mindestens diese 4000 Byte.

type

Format: *type* <dateiname> *TO* <ausg.datei> *OPT* *N* *H*

Zweck: dient zum Ausgeben einer Textdatei mit zusätzlichen Möglichkeiten gegenüber **copy**.

Erläuterung:

Der Befehl **type** entspricht weitgehend dem Befehl **copy**, wenn man diesen zur Ausgabe von Dateien auf dem Bildschirm oder dem Drucker "mißbraucht". Er ist allerdings etwas einfacher zu handhaben und bietet zwei neue Möglichkeiten. Wird z.B. das Schlüsselwort **TO** und eine Ausgabedatei nicht angegeben, so erfolgt die Ausgabe im CLI-Fenster. **Type** besitzt allerdings zwei zusätzliche Optionen, die hinter dem Schlüsselwort **OPT** angegeben werden können. Die Option **N** sorgt dafür, daß vor der Ausgabe jeder neuen Zeile die aktuelle Zeilen-Nummer ausgegeben wird. Die Option **H** bewirkt, daß die Datei nicht in Form von Text, sondern als Folge von sogenannten **Hexadezimal-Zahlen** ausgegeben wird, was wohl nur für Programmierer oder sehr erfahrene Benutzer von Interesse ist.

wait

Format: *wait* <n> SECS MINS UNTIL<uhrzeit>

Zweck: dient zum Verzögern eines Befehls entweder um eine gewisse Zeit oder bis zu einem bestimmten Zeitpunkt.

Erläuterung:

Der Befehl *wait* dürfte vor allem innerhalb von Kommandofolgen oder *run*-Befehlen sinnvoll sein. Er wird erst dann beendet, wenn ein gewisser Zeitraum vergangen ist, oder wenn eine bestimmte Uhrzeit erreicht wurde. Die letzte Möglichkeit ist natürlich nur dann sinnvoll einsetzbar, wenn die interne Uhrzeit korrekt ist (z.B. mittels *date* gestellt wurde). *Wait* verzögert also die Ausführung eines nachfolgenden Befehls und ist deshalb für sich alleingegenommen reichlich sinnlos.

Wait ohne jeden Parameter wartet genau eine Sekunde. Der Befehl *wait 7 MINS* wartet 20 Minuten, bis er beendet wird und *wait UNTIL 07:32* wartet bis 7 Uhr 32. Wird hinter *wait* nur eine Zahl und sonst nichts angegeben, wartet *wait* diese Zahl von Sekunden.

why

Format: *why*

Zweck: dient zum Erläutern der Gründe für den Fehlschlag des vorangehenden CLI-Befehls.

Erläuterung:

Wenn ein CLI-Befehl — aus welchem Grund auch immer — mißlang, so wird üblicherweise eine Fehlermeldung ausgegeben. Diese Fehlermeldung kann ausreichen, manchmal ist sie aber recht knapp oder besteht nur aus einer Zahl (eine Unsitte, deren Gründe weit zurück in der Urgeschichte der Computer, also vor 1980 liegen).

Tippt man sofort nach dem fehlgeschlagenen Befehl *why*, so bekommt man eine knappe Erläuterung der Fehlerursache in "Klartext". Dieser Klartext ist aber oft mit so vielen Fachausdrücken geladen, daß er Ihnen auch nicht immer helfen kann. In diesem Fall hilft nur Grübeln und wenn auch das nichts fruchtet, rufen Sie Ihren Händler an, oder einen Bekannten, von dem Sie wissen, daß er Ihnen auch immer solche Fachausdrücke um die Ohren schlägt.

9.4 Verwendung von Mustern

Einige der CLI-Kommandos (z.B. *copy* und *delete*) bieten Ihnen die Möglichkeit, gleichzeitig mehrere Dateien mit einem Befehl zu bearbeiten. Idealerweise sollten alle diese Dateien dazu in einem Dateiverzeichnis liegen. Falls sie Ihre Dateiverzeichnisse ordentlich angelegt haben, trifft dies vielleicht auch oft zu — wahrscheinlich aber leider nicht immer. Hier kann Ihnen manchmal die Möglichkeit helfen, *Muster* (engl. *pattern*) anzugeben, in die der Dateiname *passen* muß, um in die Gruppe der Dateien aufgenommen werden, die mit diesem Befehl bearbeitet werden sollen. Manchmal ist es möglich, den Namen eines Dateiverzeichnisses gegen ein Muster zu ersetzen. Manchmal muß das Muster aber auch durch ein spezielles Schlüsselwort eingeleitet werde (z.B. beim Befehl *list*).²⁰

Ein kurzer Tip vorweg: Für einen Menschen, der Sie ja sind, ist es nicht immer ganz einfach, zu durchschauen, welche Dateien in ein solches Muster passen. Dies gilt besonders in der Anfangszeit, solange Sie noch nicht so gut mit Mustern vertraut sind. Gerade bei Befehlen, die Dateien modifizieren und/oder löschen können, kann es deshalb zu "peinlichen" Effekten kommen, wenn plötzlich mehr oder andere Dateien durch einen Befehl erfaßt werden, als Sie dachten. **Solange Sie sich nicht ganz sicher sind, welche Dateinamen in ein bestimmtes Muster passen, verwenden Sie es niemals in Zusammenhang mit Befehlen, die Dateien modifizieren oder zerstören (löschen) könnten!**

Muster unterscheiden sich von den normalen Namen von Dateien und Dateiverzeichnissen durch besondere Zeichen (Buchstaben), die in ihnen auftreten. Diese Sonderzeichen sind () ? % # und | . Es ist deshalb wenig ratsam, wenn auch nicht unmöglich, diese Zeichen in Dateinamen zu verwenden.

Jedes dieser Sonderzeichen hat bei der Bildung von Mustern eine bestimmte Bedeutung. Einige sind Platzhalter für bestimmte Buchstaben, andere fügen Teile von Mustern zu einem großem Muster zusammen. Muster werden gebildet, indem man "normale" Buchstaben und Sonderzeichen zu einem Muster zusammenfügt.

²⁰ Der Informatiker nennt solche und ähnliche Muster reguläre Ausdrücke. Diese Muster sind für die verschiedensten Zwecke in der Informatik recht weit verbreitet. Es gibt andere Betriebssysteme z.B. UNIX™, die es dem Benutzer erlauben, bei fast jedem Befehl solche Muster für Dateinamen zu verwenden, beim Amiga gibt es jedoch nur einige Befehle, die sie verstehen.

Der einfachste Platzhalter ist `?`. Er paßt auf jeden einzelnen Buchstaben des Alphabets. Der Platzhalter `%` hingegen paßt nur auf den leeren Text, der aus keinerlei Buchstaben zwischen zwei anderen Buchstaben oder Mustern besteht. Steht in einem Muster hingegen ein normaler Buchstabe, z.B. `A`, so muß auch an der entsprechenden Stelle im Dateinamen, der in dieses Muster passen soll, ein `A` stehen. In das Muster `A?B` passen also alle dreibuchstabigen Dateinamen, deren erster Buchstabe ein `A` und deren letzter Buchstabe ein `B` ist.

Die anderen Sonderzeichen dienen zum Kombinieren von Mustern zu größeren und komplexeren Mustern. Die folgenden Beschreibungen sind recht formal, und wenn sie nicht gleich verständlich für Sie sind, machen Sie sich nichts draus! Am Ende dieses Abschnittes folgen eine ganze Reihe von Beispielen auch für komplexere Muster. Diese sollten völlig ausreichen, Ihnen einen Eindruck vom Aufbau von Mustern zu geben.

Ist `<m>` ein Muster, dann ist `#<m>` ein Muster, in das alle Texte passen, die aus Null oder mehr Wiederholungen von `<m>` bestehen. In das Muster `A#B` passen also alle Texte der Form `A`, `AB`, `ABB`, `ABB`, usw.

Sind `<m1>` und `<m2>` zwei Muster, dann ist `<m1> | <m2>` ein Muster, in das alle Texte passen, die entweder in das Muster `<m1>` oder in das Muster `<m2>` passen. In das Muster `Ja|Nein` passen also die Texte `Ja` und `Nein`.

Sind `<m1>` und `<m2>` zwei Muster, dann ist `<m><m2>` ein Muster, in das alle Texte passen, die zunächst in das Muster `<m1>` passen und deren "restlicher" Text dann in das Muster `<m2>` paßt.

Die runden Klammern (und) dienen dazu, auch größere Muster zu bilden, die die einzelnen oben vorgestellten Möglichkeiten nutzen. So werden größere Muster einerseits lesbarer und andererseits auch eindeutig genug, damit der Amiga sie verstehen kann. Klammern können Sie eigentlich nie zuviel verwenden und deshalb sollten sie auch reichlich Anwendung finden.

Ein weiteres Zeichen, das Hochkomma `'`, dient schließlich dazu, die besondere Bedeutung der Sonderzeichen aufzuheben. Schreiben Sie direkt vor einem Sonderzeichen ein Hochkomma, dann können Sie so auch nach Texten suchen, die dieses Sonderzeichen enthalten. In das Muster `A#('?)` z.B. passen die Texte `A`, `A?`, `A??????`, usw.

Im folgenden werden nun noch ein paar Beispiele für komplexere Muster vorgestellt, die Ihnen hoffentlich eine Vorstellung davon geben, was mit Mustern möglich ist. Achten Sie bitte besonders auf die Bedeutung von Klammern für den Aufbau von Mustern.

Muster	passende Texte:
A#BC	AC, ABC, ABBC, ABBBC, usw.
A#(BC)	A, ABC, ABCBC, ABCBC, ABCBCBC, usw.
A#(B C)D	AD, ABD, ACD, ABBD, ACCD, ABCD, ACBD, usw.
A?B	AAB, ABB, ACD, usw., A1B, A2B, usw.
A#?B	AB, A9B, ASchubiduB, AXYZ17QRXB, usw.
A(B %)#C	A, AB, ABC, AC, ACC, ABCCCCC, usw.
#?.TEXT	Brief.TEXT, alpha.TEXT, X1.TEXT, usw.
brief#?	brief, brief2, briefAnMama, "Brief an Mama", usw.
'?#?'#	?#, ?AB#, ?XYZ#, usw.

9.5 Überblick

Zum Abschluß dieses Kapitels folgt nun eine Liste mit allen CLI-Befehlen. Diese Liste dürfte meist für den täglichen Einsatz des CLI ausreichen. Falls Sie die genauen Parameter, die ein bestimmter Befehl hat, vergessen haben, so können Sie diese meist mit der Help-Funktion erfahren. Tippen Sie einfach hinter dem Namen des Befehls ein Fragezeichen und dann <RETURN>. Die Namen der möglichen Parameter werden Ihnen dann gleich mitgeteilt, egal, ob sie nötig oder optional sind und ob der Parameter-Name dem Parameter voran gehen muß.

Hinter den Parameter-Namen befinden sich manchmal zusätzlich noch die Codes /A, /K und /S. Der Code /A bedeutet, daß der Befehl den Parameter unbedingt benötigt (Parameter ohne /A sind optional). Der Code /K bedeutet, daß man beim Aufruf des Befehls vor dem eigentlichen Parameter noch den Namen des Parameters tippen muß. Dies hilft dem CLI, den Parameter von anderen zu unterscheiden. Der Code /S bedeutet, daß der Parameter ein "Schalter" ist. Der Parametername allein hat in diesem Fall eine Wirkung. Je nachdem, ob man den Namen beim Aufruf tippt oder nicht, verhält sich der Befehl verschieden.

Dies dürfte als Gedächtnisstütze üblicherweise genügen.

System-Befehle und Befehle für die Dateiverwaltung:

Befehl	Wirkung
ASSIGN	Ändert logische Gerätezuordnungen im System (kann für Abkürzungen mißbraucht werden).
CD	Ändert das "aktuelle Dateiverzeichnis".
COPY	Kopiert eine Datei oder eine Gruppe von Dateien.
DATE	Zeigt oder ändert das intern gespeicherte Systemdatum und die Systemzeit.
DELETE	Löscht eine Datei oder eine Gruppe von Dateien.
DISKCOPY	Kopiert komplette (nicht kopiergeschützte) Disketten recht schnell.
DIR	Zeigt den Inhalt eines Dateiverzeichnisses an.
ENDCLI	Beendet das CLI und schließt dessen Fenster.
FAULT	Zeigt den Fehlertext zu einer Fehlernummer an.
FORMAT	Formatiert und initialisiert eine 3,5-Zoll-Diskette.
INFO	Liefert Überblicksinformationen über das System.
INSTALL	Macht aus einer formatierten Diskette eine bootfähige (startfähige) Diskette.
MAKEDIR	Legt ein neues Dateiverzeichnis an.
RELABEL	Ändert den Namen einer Diskette.
RENAME	Benennt eine Datei um und/oder legt sie in ein anderes Dateiverzeichnis.

Utilities (etc.):

Befehl	Wirkung
ED	Ruft den Fullscreen-Texteditor auf.
EDIT	Ruft den Zeilen-Texteditor auf.
FILENOTE	Versieht eine Datei mit einem bis zu 80 Buchstaben langen Kommentar.
JOIN	Verbindet bis zu 15 einzelne Dateien zu einer neuen Datei.
LIST	Listet detaillierte Informationen über eine Datei oder eine Gruppe von Dateien auf.
NEWCLI	Erzeugt ein neues CLI (als eigenen unabhängigen Prozeß).
PROMPT	Ändert die Meldung, mit der das CLI meldet, daß es zum Empfang des nächsten Befehls bereit ist.
PROTECT	Setzt die Schutzflags einer Datei (unter anderem Lese-, Schreib-, Lösch-Schutz).
RUN	Läßt einen oder mehrere CLI-Befehle im Hintergrund ablaufen.
SEARCH	Durchsucht alle Dateien eines Dateiverzeichnisses nach einem bestimmten Text.
SORT	Sortiert Textdateien alphabetisch nach der ersten oder einer beliebigen Spalte in jeder Zeile.
STACK	Ändert die Größe des Stacks, der beim Start eines anderen Programms vom CLI reserviert wird (Default = 4000 Byte).
STATUS	Zeigt verschiedene Informationen über andere Prozesse (Tasks), die im Moment geladen sind (vorzugsweise andere CLI-Prozesse).
TYPE	Gibt eine Datei auf dem Bildschirm aus (auf Wunsch hexadezimal oder mit Zeilennummern).
WHY	Zeigt (als Text) die Ursache des letzten Fehlers, der sich bei der Ausführung eines Programms ereignet hat.

Befehle für Kommandofolgen (Batch-Files):

Befehl	Wirkung
ECHO	Gibt (während der Ausführung einer Kommandofolge) einen Text aus.
EXECUTE	Startet eine Kommandofolge (in einer Text-Datei).
FAILAT	Verhindert den Abbruch einer Kommandofolge bei geringfügigen Fehlern.
IF	Leitet die bedingte Ausführung einer Teilfolge von Befehlen in einer Kommandofolge ein.
LAB	Versieht eine Stelle in einer Kommandofolge mit einer Marke (s. SKIP).
QUIT	Verläßt eine Kommandofolge "mittendrin" und gibt eventuell einen Fehlercode zurück.
SKIP	Springt (nur nach vorne) zu einer Marke (s. LAB).
WAIT	Wartet eine bestimmte Zeitspanne oder bis zu einem bestimmten Zeitpunkt mit der Ausführung des nächsten Befehls.

10

Der Text-Editor *ed*

Eine Aufgabe, die — nicht nur im CLI — immer wieder anfällt, ist das Erzeugen und Ändern von Textdateien. Ob dies einfache Notizen, komplexe Programme oder (vor allem) Kommandofolgen sind, immer wird ein Werkzeug benötigt, welches diese Aufgabe so gut wie möglich unterstützt. Ein solches Werkzeug wird üblicherweise *Text-Editor* genannt. ***Ed*** ist ein solcher Text-Editor mit dem *reine Textdateien* behandelt werden können. Es handelt sich nicht um eine komplette Textverarbeitung, die man z.B. für Büro Zwecke einsetzen könnte, sondern nur um ein Hilfsmittel zum Bearbeiten von unstrukturierten Texten.¹ Ein Text-Editor erkennt keine Seiten, Kapitel, Absätze, und vielleicht nicht einmal Worte. Er erkennt nur einen Strom von Buchstaben, der in Zeilen aufgeteilt ist. In einem Text-Editor können Texte nicht fett gedruckt oder unterstrichen werden. Es ist ein sehr simples Werkzeug, das selbst für Briefe etwas spartanisch ist, aber doch wichtig für die Aufbereitung von Texten, die als Eingaben für andere Programme dienen sollen oder für die Betrachtung von Texten, die andere Programme erzeugen.

So bestehen z.B. Kommandofolgen nur aus Buchstaben und Zeilen, und auch die Dateien, die entstehen, wenn man die Ausgabe eines CLI-Befehls mittels `>` vom CLI-Fenster auf eine Datei umleitet, sind meist einfache Textdateien.

Der Text-Editor ***ed*** ist allerdings schon ein recht komfortabler Vertreter seiner Gattung. Es ist ein sogenannter *Bildschirm-Editor* (engl. *screen editor*), der uns einen Ausschnitt — ein Fenster — aus dem bearbeiteten Text zeigt. In

¹ Im angloamerikanischen Bereich wird diese Unterscheidung auch im verbalen Bereich etwas deutlicher. Bei einem komfortablen Textverarbeitungssystem spricht man meist von einem *word processor* und bei einem einfachen Text-Editor eben auch von einem *editor*.

diesem Ausschnitt kann man mit den Pfeil- oder Cursor-Tasten rechts unten auf der Tastatur hin- und herfahren und an beliebigen Stellen Texte überschreiben, löschen oder ergänzen. Andere Text-Editoren — die Vertreter der Gattung *Zeilen-Editor* (engl. *line editor*) z.B. — machen die Bearbeitung eines Textes wesentlich umständlicher. Sie erlauben unter anderem nur die Bearbeitung einer aktuellen Zeile, und jeder Wechsel zu einer anderen Zeile ist mit speziellen Befehlen verbunden.² Obwohl **ed** in mancher Hinsicht einiges zu wünschen übrig läßt, kann man mit ihm doch schon recht gut auskommen.

10.1 Aufrufen und Verlassen von ed

Der Editor **ed** (wenn ich im folgenden Editor schreibe, meine ich natürlich immer einen Text-Editor) wird üblicherweise vom CLI aus aufgerufen. Das genaue Aufruf-Format lautet:

```
ed <dateiname> SIZE <n>
```

Dabei kann **<dateiname>** sowohl eine neue Datei sein, die Sie mit **ed** erzeugen wollen, als auch eine alte, die Sie verändern oder nur betrachten wollen. SIZE ist ein optionaler Parameter. Wenn Sie dem Schlüsselwort SIZE eine Zahl folgen lassen, so bestimmt diese die *Puffergröße* von **ed**, d.h. die maximale Größe des Textes (in Buchstaben bzw. Anschlägen), die sie mit **ed** gerade noch bearbeiten können. Standardmäßig ist dieser Puffer 40 000 Anschläge groß — mehr als ausreichend für unsere ersten Versuche.

Starten Sie nun das CLI, falls Sie sich noch nicht darin befinden, und legen Sie mit **makedir df0:texte** ein neues Dateiverzeichnis für die folgenden Versuche an. Sie können es nach Beendigung dieses Kapitels löschen — es wird aber nur ein paar tausend Byte groß werden und Ihren Diskettenplatz kaum einschränken. Das neue Dateiverzeichnis machen Sie nun bitte mit **cd df0:texte** zum aktuellen Dateiverzeichnis. Starten Sie nun **ed**, indem Sie **ed text1** eintippen und dann natürlich <RETURN>.

² Zeilen-Editoren haben unter anderem andere Vorteile. Der Zeilen-Editor **edit** z.B., der sich ebenfalls auf Ihrer Workbench-Diskette befindet, kann hervorragend für automatisch durchgeführte, größere Änderungen an großen Texten eingesetzt werden. Benötigen Sie ein solches Werkzeug, so können Sie mehr über **edit** im *AmigaDOS User's Manual* erfahren, das Ihnen Ihr Amiga-Händler besorgen kann.

Sie sollten sich nun in *ed* befinden. Lassen Sie die Tastatur einen Moment in Ruhe und betrachten Sie den Bildschirm. Ein neues Fenster hat sich geöffnet, das den gesamten Bildschirm füllt und in dessen linker oberer Ecke ein kleines helles Rechteck erscheint, der sogenannte *Cursor* (zu deutsch etwa *Schreibmarke*; ich werde aber auch im folgenden das inzwischen auch bei uns weit verbreitete Wort *Cursor* verwenden). Das *ed*-Fenster ist ein ganz normales Fenster und kann mit den üblichen Methoden verkleinert und dann verschoben werden. Probieren Sie dies nun und Sie werden sehen, daß dahinter das CLI-Fenster auftaucht und alle anderen Objekte der Workbench. Sie können auch dieses Fenster ergreifen und verschieben, verkleinern und vergrößern und andere Programme starten (sofern noch genügend Speicher dafür vorhanden ist). Sie können aber keine Befehle in das CLI-Fenster eintippen, von dem aus Sie *ed* aufgerufen haben.³ Da Sie direkt *ed* aufgerufen haben, wartet das CLI jetzt erst auf die Beendigung von *ed*, bevor es neue Befehle akzeptiert.⁴

Vergrößern Sie das *ed*-Fenster nun wieder auf angenehme Größe (es muß nicht die volle Bildschirmgröße sein), damit Sie mit Ihren ersten Versuchen beginnen können.

Klicken Sie sicherheitshalber einmal mit der Maus ins Innere des *ed*-Fensters und tippen danach ein paar Buchstaben. Es kann ruhig ein sinnvoller Satz sein, Sie können aber natürlich auch einfach mit den Fingern etwas auf die Tastatur trommeln. Die Buchstaben erscheinen nun nacheinander — in der Reihenfolge in der Sie sie eingetippt haben — auf dem Bildschirm. Der Cursor, das helle Rechteck wandert dabei immer weiter und steht immer rechts neben dem letzten eingegebenen Buchstaben. Drücken Sie nun ein paarmal auf die <Backspace>-Taste in der oberen rechten Ecke der Tastatur (falls Sie es vergessen haben, es ist die Taste mit dem Pfeil nach links <-darauf). Sie sehen dann, wie die letzten Buchstaben, die Sie getippt haben, gelöscht werden und der Cursor wieder nach links wandert.

Bis hierhin ist die Eingabe von Text genauso, wie Sie es aus dem CLI kennen. Eingetippter Text erscheint unter dem Cursor, der dadurch nach rechts wandert. Mit der <Backspace>-Taste kann jeweils der letzte Buchstabe rechts gelöscht werden.

³ Sie können zwar Befehle tippen, und sie erscheinen auch am Bildschirm. Mit der Ausführung wird aber erst dann begonnen, wenn Sie *ed* verlassen..

⁴ Wollen Sie den Editor aufrufen und trotzdem noch die Möglichkeit haben, das CLI zu nutzen, so müssen Sie den "Umweg" über den *run*-Befehl wählen. Mehr dazu bei der Beschreibung des *run*-Befehls.

Nun brechen Sie bitte Ihre erste **ed**-Sitzung einmal ab, und kehren Sie zurück ins CLI. Sie wissen ja, es gibt zwei wichtige Dinge, die man sofort über jedes Programme lernen muß: wie man hineinkommt und wie man unbeschadet wieder herauskommt. Wie man hineinkommt, haben Sie schon gesehen, und nun werden Sie sehen, wie man wieder herauskommt.

Das Programm **ed** wird über einen sogenannten *Escape-Befehl* verlassen, von denen Sie weiter unten noch einige mehr kennenlernen werden. Tippen Sie zunächst kurz auf die <ESC>-Taste (<ESC> steht für *escape*) am oberen linken Rand der Tastatur. Der Cursor springt daraufhin an den unteren Rand des Fensters und erwartet eine weitere Eingabe. (Dies geschieht immer bei Escape-Befehlen.) Tippen Sie nun den Buchstaben **x** und dann <RETURN>. Es erscheint nun kurz eine Meldung am unteren Fensterrand, die wahrscheinlich aber zu schnell verschwindet, als daß Sie sie lesen könnten. Danach verschwindet das **ed**-Fenster, und Sie sind wieder im CLI.

Tippen Sie nun den Befehl **dir** — und wie Sie sehen, haben Sie soeben eine neue Datei erzeugt, die unter dem Namen **text1** aufgelistet wird. Diese Datei werden wir nun im nächsten Kapitel weiter bearbeiten.

10.2 Die direkten ed-Befehle

Die Befehle, die **ed** für die Bearbeitung, das Abspeichern und Lesen von Texten kennt, zerfallen in zwei große Gruppen. Die *direkten* Befehle werden sofort ausgeführt, nachdem Sie auf eine bestimmte Taste oder eine Kombination von zwei Tasten gedrückt haben. Die *erweiterten* oder *Escape-Befehle* werden eingeleitet durch ein Drücken der <ESC>-Taste und erwarten danach immer eine weitere Eingabe, bevor der Befehl nach dem Drücken der <RETURN>-Taste endlich gestartet wird. In diesem Abschnitt werden die direkten Befehle vorgestellt, die Sie wahrscheinlich auch am häufigsten benötigen werden und im folgenden Abschnitt dann die *Escape-Befehle*.

Starten Sie nun bitte erneut **ed** mit dem Befehl **ed text1**. Diesmal ist das erscheinende Fenster nicht leer, sondern zeigt Ihnen bereits den Text, den Sie bei der ersten Sitzung eingegeben haben. Dies illustriert sehr schön das unterschiedliche Verhalten von **ed**, je nachdem ob Sie eine bereits vorhandene Datei editieren oder eine neue Datei erzeugen. Verlassen Sie **ed** nämlich in der oben vorgestellten Weise, wird der Text, den Sie bearbeitet haben, immer in eine Datei geschrieben, deren Namen Sie **ed** beim Aufruf als ersten Parameter mitgeteilt haben. Existierte bereits eine Datei dieses Namens, wird ihr alter

Inhalt dabei zerstört. Gab es noch keine Datei dieses Namens, so wird eine neue unter diesem Namen erzeugt und in ihr der Text abgelegt.

Zwischendurch, während Sie einen Text bearbeiten, wird nur ein Pufferinhalt geändert, der im Hauptspeicher (RAM) der Zentraleinheit verwaltet wird. Die Datei auf der Diskette bleibt unbeeinträchtigt. Sie brauchen sich also erst am Ende einer Sitzung zu überlegen, ob Sie Ihre Änderungen wirklich abspeichern, oder lieber die alte Version eines Textes behalten wollen. Auch können sie jederzeit während der Arbeit an einem Text zur alten Version zurückkehren. Nach diesen hoffentlich beruhigenden Worten nun aber zu den direkten Befehlen.

Fast jede der Tasten auf der Tastatur des Amiga ist strenggenommen ein solcher direkter Befehl. Die Buchstaben-Tasten sind z.B. ein Befehl, an der Stelle, an der sich gerade der Cursor befindet, einen Buchstaben einzufügen. Die <RETURN>-Taste ist ein Befehl, eine neue Zeile zu beginnen usw. Direkte Befehle zerfallen in die folgenden Gruppen:

- Bewegung des Cursors
- Einfügen von Text
- Löschen von Text
- Verschieben des Text-Fensters im Gesamt-Text
- Besonder Befehle

10.2.1 Bewegung des Cursors

Die wichtigsten Befehle, die Sie innerhalb von *ed* nahezu andauernd benötigen werden, sind die Befehle für die Bewegung des Cursors. Hierzu dienen vor allem die vier mit einem Pfeil versehenen Tasten rechts unten auf der Tastatur. Dies sind die sogenannten *Cursor-Tasten*. Sie bewegen den Cursor jeweils um eine Position in der angegebenen Richtung. Versuchen Sie es einmal, um damit vertraut zu werden.

Falls ihr Text im Moment nur aus einer Zeile besteht, tippen Sie einfach mehr ein. Tippen sie dabei vor allem häufig auf die <RETURN>-Taste, um neue Zeilen zu beginnen. Tippen Sie solange Text ein, bis das Fenster wenigstens gefüllt ist, damit Sie genug für die folgenden Experimente haben. Falls Ihnen kein passender Text einfällt, tippen Sie einfach eine Buchseite oder einen Zeitungsartikel ab. Sobald Sie am unteren Ende des *ed*-Fensters angekommen sind (machen Sie es ruhig etwas kleiner, damit Sie den Effekt schnellen sehen), rutscht der bisher getippte Text nach oben, wie auch im CLI-Fenster. (Im Gegensatz zum CLI-Fenster kann man ihn aber wieder zurückholen, wie Sie weiter unten noch sehen werden.)

Sie sollten nun genügend Text für die meisten der folgenden Versuche haben. Experimentieren Sie darin ein wenig mit den Cursor-Tasten. Halten Sie auch einmal eine Taste etwas länger fest und beobachten, was geschieht. Nach einer kleiner Pause "wiederholt sich die Taste selbst" und der Cursor schießt in die angegebene Richtung, als hätten Sie sehr oft auf die Taste gedrückt. Dies funktioniert bei allen Cursor-Tasten und ist sehr praktisch, sobald man damit umgehen kann. Die Länge der Pause vor Beginn der Tasten-Wiederholung und die Geschwindigkeit der Wiederholung können übrigens mit dem Programm *Preferences* eingestellt werden.

Wenn der Cursor an das rechte Ende des Fensters kommt (machen Sie es ruhig etwas kleiner, damit Sie den Effekt schnellen sehen), rutscht der ganze Text nach links. Auf diese Weise können Sie auch Texte bearbeiten, die breiter sind, als das Fenster werden kann. *Ed* kann Texte bearbeiten, in denen die Zeilen eine maximale Breite von 255 Buchstaben haben. Das dürfte für die meisten Zwecke ausreichen. Um wieder zurückzukommen, um also den Text am linken Rand wiedersehen zu können, halten Sie einfach die Cursor-Taste fest, die nach links zeigt. Sobald der Cursor am linken Fensterrand ankommt, verschiebt sich der Fensterinhalt nach rechts. Allerdings nur solange, bis Sie in der ersten Spalte angekommen sind.

Genauso können Sie übrigens auch den Text nach oben und unten verschieben, indem Sie die nach oben und nach unten weisenden Cursor-Tasten verwenden. Probieren Sie es ruhig aus. Es funktioniert allerdings nur, wenn Ihr Text mehr Zeilen hat, als im Fenster Platz haben. Das Fenster ist also wirklich ein Fenster in einen Text, das einen beliebigen Ausschnitt der Gesamtheit zeigen kann. Die folgende Abbildung versucht, dies bildlich zu verdeutlichen.

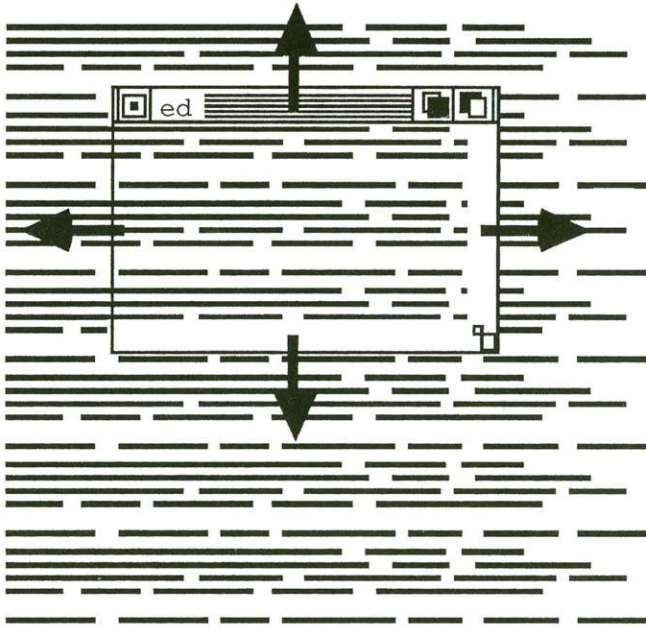


Bild 10 - 1: Ein langer Text und der Ausschnitt, den ed davon zeigt.

Nun aber genug des Hin- und Herhüpfens im Text. Es gibt noch einige spezielle Befehle, die Sie zu "herausragenden" Stellen in einem Text bringen. Die Tastenkombination `<CTRL>-]` bringt Sie stets an das Ende der Zeile, in der sich der Cursor gerade befindet. Wenn der Text breiter ist als das Fenster, verschiebt er sich dabei natürlich nach links. Falls sich der Cursor schon auf dem letzten Buchstaben der Zeile befand, springt er bei `<CTRL>-]` zum Anfang der Zeile.

Ein Wort zur Schreibweise für solche Befehle: `<CTRL>`, dann ein Bindestrich und dahinter ein Buchstabe bedeutet, daß Sie die Taste am linken Rand der Tastatur festhalten sollen, auf der **CTRL** steht und dann gleichzeitig die Taste mit dem anderen Buchstaben drücken. Der Bindestrich bedeutet, daß Sie beide Tasten **gleichzeitig** drücken müssen. (Fehlt ein Bindestrich bei ähnlichen Beschreibungen, müssen die beiden Tasten nacheinander gedrückt werden.) `<CTRL>-]` bedeutet also Festhalten der `<CTRL>`-Taste und Drücken der Taste "]" (eckige Klammer zu).

Die Tastenkombination `<CTRL>-E` bringt Sie zum Anfang der ersten Zeile auf dem Bildschirm. Falls der Cursor aber schon dort war, springt er bei `<CTRL>-E` zum Ende der letzten Zeile auf dem Bildschirm.

Die Tastenkombination <CTRL>-T bringt Sie zum Anfang des nächsten Wortes am Bildschirm. <CTRL>-T bringt Sie auf das Leerzeichen, das dem vorangehenden Wort folgt. Falls diese beiden Positionen außerhalb des Fensters liegen, wird der Text entsprechend verschoben.

Die <TAB>-Taste schließlich (mitten links auf der Tastatur) springt immer um N Schritte nach rechts. Die Zahl N, also die Schrittweite, ist durch einen anderen Befehl einstellbar.⁵

Damit wären die Möglichkeiten der Cursor-Bewegung zunächst einmal aufgezählt. Spielen Sie ruhig ein wenig mit den Möglichkeiten herum. Weiter unten werden Sie noch andere Möglichkeiten für "größere Sprünge" im Text kennenlernen, die aber in eine andere Kategorie fallen. Falls Ihnen das Gelernte erst einmal reicht, beenden Sie die Sitzung mit <ESC>x<RETURN>.

10.2.2 Einfügen von Text

Beim Eintippen des Textes haben Sie bestimmt das eine oder andere Mal einen Fehler gemacht. Fällt er sofort auf, kann man einen solchen Fehler ganz gut mit der <Backspace>-Taste beseitigen. Dies lohnt aber kaum, wenn er schon zwei Zeilen zurückliegt. Für die Korrektur solcher Fehler und größere Änderungen an Texten müssen Sie Texte löschen und nachträglich alte Texte einfügen können. Zunächst kommt das Einfügen dran; das Löschen von Text wird im nächsten Abschnitt behandelt.

Falls Sie sich nicht noch im *ed* befinden, starten Sie Ihn nun wieder mit *ed text1*. Sie bekommen dann wieder den Beginn des im letzten Kapitel fertiggestellten Textes gezeigt und können dort weitermachen, wo Sie bei der letzten Sitzung aufgehört haben. Gehen Sie nun mit den Cursor-Tasten oder andern Bewegungsbefehlen an eine beliebige Stelle im Text und tippen einige neue Worte ein. Wie Sie sehen, bewegt sich der alte Text rechts der Cursorposition beim Tippen nach rechts, um den neu getippten Buchstaben Platz zu machen. Die meisten Textverarbeitungssysteme und Editoren arbeiten so. Neuer Text überschreibt nicht den alten, sondern wird zwischen zwei Buchstaben eingesetzt. Wollen Sie also ein Wort gegen ein anderes ersetzen, so müssen Sie das alte zuerst löschen und dann das neue einfügen.

⁵ Wer schon andere Textverarbeitungssysteme kennt, wird die <TAB>-Taste zunächst vielleicht mißverstehen. Die <TAB>-Taste auf dem Amiga springt N Buchstaben weiter, fügt aber keine Buchstaben (Leerzeichen oder ein besonderes Tab-Zeichen) in den Text ein.

Die Zeilen können maximal 255 Buchstaben lang werden. Dabei ist es egal, ob Sie dieses Limit beim Eintippen einer neuen Zeile überschreiten oder beim Einfügen in eine bestehende. Bei längeren Zeilen gibt es eine Fehlermeldung (*Line too long*), die in der untersten Zeile des **ed**-Fensters erscheint.

Tippen Sie statt eines normalen Buchstabens ein `<RETURN>`, so beginnt eine neue Zeile — wie sie es ja auch schon kennen. Der Cursor springt aber nicht einfach in die nächste Zeile, sondern er fügt eine neue leere Zeile zwischen der aktuellen und der nächsten ein. Sie haben dies am Anfang vielleicht nur nicht bemerkt, weil hinter der Zeile, die Sie gerade eingetippt haben, keine andere mehr kam. Sie können auf diese Weise auch eine alte Zeile in zwei neue aufspalten, indem Sie den Cursor in die alte Zeile an die Stelle bewegen, an der getrennt werden soll, und dann `<RETURN>` tippen.

Das waren aber schon alle "Befehle" für das Einfügen von Text. Nachdem wir bisher kreativ waren und nur neuen Text erzeugt haben, werden wir uns nun mit der Zerstörung beschäftigen.

10.2.3 Löschen von Text

Einen der wichtigsten Befehle zu Löschen von Text haben Sie ja bereits kennengelernt: die `<Backspace>`-Taste. Diese Taste löscht den Buchstaben links vom Cursor und bewegt dann den Cursor um eine Position nach links. Das funktioniert nicht nur am Ende einer Zeile, sondern auch mitten drin. Der Text rechts vom gelöschten Buchstaben (falls vorhanden) rutscht nämlich ebenfalls um eine Position nach links. Probieren Sie es aus!

Mit der `<DEL>`-Taste (unter der `<Backspace>`-Taste) können Sie den Buchstaben genau an der Cursorposition löschen. Der Cursor bleibt dabei stehen, nur der Text rechts davon rutscht nach links.

Die Tastenkombination `<CTRL>-O` (der Buchstabe 'o'; nicht die Zahl Null) löscht schon etwas größere "Brocken". Ihre Wirkung ist abhängig von dem Buchstaben an der Cursorposition. Ist dieser Buchstabe ein Leerzeichen, so werden dieses und alle folgenden Leerzeichen bis hin zum nächsten "Nicht-Leerzeichen" gelöscht. Befindet sich kein Leerzeichen an der Cursorposition, löscht `<CTRL>-O` alle Buchstaben ab der Cursorposition bis zum nächsten Leerzeichen.

Die Tastenkombination `<CTRL>-Y` löscht den "Rest" der Zeile ab der Cursorposition und rechts davon.

Die Tastenkombination <CTRL>-B löscht die ganze aktuelle Zeile; egal wo der Cursor darin steht. Der folgende Text rutscht um eine Zeile hoch.

10.2.4 Verschieben des Text-Fensters im Text

Wie bereits erwähnt, zeigt *ed* immer nur einen Ausschnitt des gesamten bearbeiteten Textes. Er kann sich sowohl horizontal als auch vertikal weit über die Größe des Fensters hinaus erstrecken. Bei verschiedenen Befehlen wird automatisch dieser Ausschnitt verändert, so z.B. wenn man mit dem Cursor an den Rand des Fensters stößt. Es gibt aber auch Befehle, mit denen man den Text ganz explizit im Fenster verschieben kann. <CTRL>-D und <CTRL>-U sind zwei Tastenkombinationen, mit denen der Text im Fenster gleich ein ganzes Stück nach unten (engl. *down*) bzw. nach oben (engl. *up*) bewegt werden kann.

Die Tastenkombination <CTRL>-D bewegt den Text im Fenster, falls möglich, um 12 Zeilen nach unten (oben erscheinen also 12 neue Zeilen), und bewegt gleichzeitig den Cursor 12 Zeilen nach oben.

Die Tastenkombination <CTRL>-U bewegt den Text im Fenster, falls möglich, um 12 Zeilen nach oben (unten erscheinen also 12 neue Zeilen), und bewegt gleichzeitig den Cursor 12 Zeilen nach unten.

10.2.5 Besondere Befehle

Zwei weitere direkte Befehle müssen noch erwähnt werden.

Die Tastenkombination <CTRL>-G wiederholt den letzten Escape-Befehl, der erteilt wurde. Escape-Befehle sind ja typischerweise viel umständlicher einzugeben, weshalb dieser Befehl einige Tipparbeit sparen kann. Wie sinnvoll das sein kann, davon mehr in einem der nächsten Abschnitte.

Die Tastenkombination <CTRL>-F ändert die Groß-/Kleinschreibung des Buchstabens an der Cursorposition und verschiebt den Cursor um eine Position nach rechts. Leerzeichen und Sonderzeichen werden nicht beeinflusst (Umlaute leider auch nicht). Dies kann besonders praktisch für Programmierer sein, ist manchmal aber auch für die Fehlerkorrektur von Nutzen.

10.3 Escape-Befehle

Die zweite Gruppe von Befehlen, die **ed** erkennt, ist etwas umständlicher zu erteilen. Einen davon, den Befehl zum Verlassen von **ed**, haben sie ja schon kennengelernt. Alle diese Befehle verlangen mehrere Tastendrucke, können dafür aber auch einiges mehr als die direkten Befehle. Zusätzlich haben fast alle direkten Befehle noch einmal ein Gegenstück als Escape-Befehl, der im wesentlichen das selbe tut. Dies ergibt erst im Zusammenhang mit der Möglichkeit, Befehle sehr einfach zu wiederholen, einen Sinn. Auf diese Möglichkeit wird im vorletzten Abschnitt dieses Kapitels noch etwas näher eingegangen.

Die Beschreibung der meisten Escape-Befehle wird knapp ausfallen. Eine ausführliche Erörterung verbietet sich aus Platzgründen. Nicht alle Leser dieses Buches würden sie schließlich auch zu schätzen wissen, da Sie all diese Befehle vielleicht nie nutzen werden. Ein wenig mit den Befehlen zu experimentieren, die Sie wirklich interessieren, ist die beste Möglichkeit, damit vertraut zu werden (natürlich immer auf der Kopie der Original-Workbench-Diskette, die Sie zu Beginn dieses Buchteils ja anfertigen sollten).

10.3.1 Grundsätzliches zu Escape-Befehlen

Escape-Befehle werden immer durch einen Druck auf die <ESC> (Escape-Taste) eingeleitet. Der Cursor springt dann in die unterste Zeile des Fensters, wo Sie einen Befehl eingeben können und nach der Ausführung des Befehls wieder zurück in den Text, eventuell an die Stelle, an der Sie <ESC> gedrückt haben, unter Umständen aber auch an eine andere. Diese Zeile am unteren Rand des Fensters heißt *Kommando-Zeile* oder *Befehls-Zeile* und dient unter anderem auch für die Mitteilung von Fehlermeldungen.

Escape-Befehle bestehen aus einem oder zwei Buchstaben, worauf eventuell noch ein oder zwei Parameter in Form eines Textes folgen. (Ob man den Befehl groß oder klein schreibt ist übrigens egal, ich werde ihn hier immer groß schreiben. Die kleine Schreibweise ist aber meist praktischer.) Zur Korrektur der Befehle bei Tippfehlern stehen nicht allzu komfortable Möglichkeiten zur Verfügung. Wie im CLI kann nur der jeweils letzte Buchstabe der Zeile mittels der <Backspace>-Taste gelöscht werden. Sind Sie mit der Eingabe eines Befehls fertig, so beenden Sie ihn durch Druck auf die <RETURN>-Taste oder die <ESC>-Taste. Bei Verwendung der

<RETURN>-Taste kehrt der Cursor nach der Ausführung des Befehls zurück in das Text-Fenster. Tippen Sie aber auf die <ESC>-Taste, so bleibt der Cursor auch nach der Ausführung des Befehls in der Kommando-Zeile und Sie können den nächsten Befehl erteilen.

Die Parameter von Escape-Befehlen sind Texte, die, im Gegensatz zu den Texten für CLI-Befehle, immer durch besondere Zeichen begrenzt werden müssen. Im CLI gab es zwar als Begrenzer auch die Anführungszeichen (""); sie waren aber nur bei Texten nötig, die Leerzeichen enthalten. Die Parameter von Escape-Befehlen können durch beliebige Sonderzeichen begrenzt werden, solange sie mit dem selben Sonderzeichen aufhören, mit dem Sie auch begonnen wurden.

```
"Ein Text" 'Text' /noch einer/ ;komisch, nicht?;
```

Diese vier Texte sind also vier völlig legitime Parameter für Escape-Befehle mit sehr verschiedenen Begrenzern. Auf diese Weise können auch Texte eingegeben werden, die selbst ein beliebiges Sonderzeichen enthalten.

Bei allen Beschreibungen der verschiedenen Escape-Befehle werden das einleitende Tippen auf die <ESC>-Taste und das beendende <ESC> oder <RETURN> im allgemeinen nicht miterwähnt. Sie sind aber nichtsdestotrotz immer notwendig!

10.3.2 Globale Escape-Befehle

Globale Escape-Befehle sind Befehle, die den Text als ganzes betreffen, ihn also z.B. abspeichern oder Formateinstellungen ändern oder abfragen.

Der erste globale Befehl, den Sie kennengelernt haben, war **X**, mit dem *ed* verlassen wird, mit dem gleichzeitig aber der Text-Puffer, der die gerade bearbeitete Version des Textes enthält, auf Diskette (bzw. Festplatte) gesichert wird. Erst ab diesem Moment sind alle Änderungen unwiderruflich.

Der Befehl **Q** beendet ebenfalls das Programm, jedoch ohne zuvor die Änderungen in der Datei zu sichern. Haben Sie aber Änderungen, die noch nicht gesichert wurden, so werden Sie vor dem Verlassen des Programms noch gefragt, ob Sie diese Änderungen wirklich "vergessen" wollen. Nur wenn Sie mit **y** (für *yes* bzw. *Ja*) antworten, sind die Änderungen wirklich verloren. Sie erhalten so noch eine letzte Chance. Normalerweise werden Sie *ed* aber wohl immer mit <ESC>**X** verlassen.

Zum "Zwischendurch"-Sichern Ihres Textes dient der Befehl **SA** (*save all* oder zu deutsch *sichere alles*). Sie sollten diesen Befehl bei größeren Texten

ruhig öfter einmal anwenden, damit Ihre Arbeit nicht bei jeder mittleren Katastrophe verlorenggeht. Schließlich könnte es ja einen Stromausfall geben oder Ihr Sprößling kommt vielleicht auf den Gedanken, Ihrem Computer einmal den Stecker herauszuziehen.

Sollten Sie einmal bemerken, daß die Änderungen, die Sie in einer Zeile gemacht haben, ihnen nicht mehr gefallen, so können Sie den Befehl <ESC>-U verwerde. *U* steht für *Undo* (engl. für *Rückgängig machen, widerrufen*). Immer wenn Sie in eine neue Zeile kommen und dort Veränderungen beginnen, legt *ed* zuvor noch eine Kopie der alten Zeile in einen Zwischenspeicher. Diese Kopie wird durch den *U*-Befehl wieder "zurückgeholt".

10.3.3 Text-Blocks

Wenn Sie größere Bereiche im Text löschen oder eine Gruppe von Zeilen duplizieren oder verschieben wollen, können Sie *Text-Blocks* verwenden. Es gibt immer genau einen aktuellen Text-Block, den Sie durch spezielle Escape-Befehle markieren können. Text-Blocks können gelöscht, an einer anderen Stelle in den Text eingefügt und einzeln auf Diskette gesichert werden.

Den Anfang eines Text-Blocks markieren Sie, indem Sie die entsprechende Zeile mit dem Cursor aufsuchen und dann den Befehl <ESC>-BS (*Block Start*) geben. Gehen Sie dann in die letzte Zeile, die gerade noch zum Block gehören soll, und tippen Sie <ESC>BE (*Block Ende*) ein. (Blocks enthalten immer nur ganze Zeilen!)

Um den so markierten Block an anderer Stelle einzufügen, gehen Sie mit dem Cursor an diese Stelle (meist an den Beginn einer Zeile) und geben den Befehl <ESC>-IB (*Insert Block; Block einfügen*) ein. IB können Sie beliebig oft aufrufen und so beliebig viele Kopien des markierten Blocks an vielen Stellen eines langen Textes einfügen. Wollen Sie danach den Original-Block aus dem Texte entfernen, so brauchen Sie nur <ESC>DB (*Delete Block; Block löschen*) einzugeben. DB löscht immer den zuletzt markierten Block. Dieser ist danach unwiederbringlich verloren!

Manchmal ist es sinnvoll, einen Text in mehrere kleine Texte aufzuspalten. Dies können Sie erreichen, indem Sie den markierten Block in einer separaten Datei abspeichern (und ihn dann vielleicht löschen). Der Befehl <ESC>-WB <dateiname> (*Write Block; Block schreiben*), leistet genau das. <dateiname> muß dabei natürlich durch den Namen einer Datei (in Begrenzer-Zeichen; s.o) ersetzt werden. Auf ähnliche Weise können Sie eine Datei an

der aktuellen Cursor-Position in den gerade bearbeiteten Text einfügen indem Sie `<ESC>-IF <dateiname>` (*Insert File; Datei einlesen*) eingeben. Die folgende Beispiel-Befehlsfolge ersetzt z.B. den gerade markierten Block durch den Inhalt einer Datei *Neu* aus dem Dateiverzeichnis *texte*, wobei zuvor aber der alte Blockinhalt in der Datei *Alt* gesichert wird.

```
<ESC>-WB !:texte/Alt!  
<ESC>-DB  
<ESC>-IF !:texte/Neu!
```

10.3.4 Befehle zur Cursor-Bewegung

Alle Befehle zur Cursor-Bewegung gibt es auch noch einmal als Escape-Befehle. Ich gehe nicht mehr zu ausführlich darauf ein, da sie ja im wesentlichen bereits bekannt sind.

Der Befehl **B** setzt den Cursor in die letzte Spalte der letzten Zeile des gesamten Textes (nicht des aktuellen Fensters, obwohl dies gelegentlich übereinstimmen kann). Der Befehl **T** setzt den Cursor entsprechend in die erste Spalte der ersten Zeile des gesamten Textes.

Der Befehl **CE** setzt den Cursor in die letzte Spalte der aktuellen Zeile. **CL** bewegt ihn um eine Position nach links. **CR** bewegt den Cursor um eine Position nach rechts und **CS** bewegt ihn zum Start der aktuellen Zeile.

Der Befehl **N** setzt den Cursor in die nächste Zeile. **P** setzt ihn in die vorangehende Zeile. Mit dem Befehl **M** und einer Zahl dahinter schließlich kann man zu einer bestimmten Zeile in einem langen Text springen. Deren Nummer muß nur bekannt sein. Der folgende Befehl springt so z.B. in die 113te Zeile eines Textes.

```
<ESC>M 113
```

Dies ist besonders für Programmierer interessant, denen Programmfehler oft als Zeilenzahlen angegeben werden. Die anderen Escape-Befehle sind aber wohl nur in Befehlsfolgen interessant (s.u.), da es wesentlich knappere direkte Befehle für die meisten dieser Bewegungen gibt.

10.3.5 Suchen und Ändern von Text

Während die meisten Escape-Befehle, die Sie bis jetzt kennenlernten, auch mit direkten Befehlen zu erreichen waren, machen einem die Befehle zum

Suchen und vor allem zum Ändern von Textstellen das Leben wirklich leichter. Diese Möglichkeiten werden oft – fälschlicherweise – für den Inbegriff der Textverarbeitung überhaupt gehalten.

Der Befehl <ESC>F <text> (*Find; finde*) sucht <text>, ausgehend von der aktuellen Cursor-Position, positioniert den Cursor auf den Beginn des ersten Vorkommens. Falls nötig, wird das Text-Fenster dazu entsprechend verschoben. Genauso verhält sich der Befehl <ESC>BF (*Backwards Find; finde rückwärts*), der jedoch nicht von der aktuellen Position nach "unten" sondern zurück, in Richtung Anfang der Datei, sucht.

<ESC>E /alt/neu/ (*Exchange; Austauschen*) sucht den Text /alt/ und ersetzt ihn durch /neu/, falls er gefunden wird. Wollen Sie sicher gehen, daß nur die Textstelle geändert wird, die Sie gemeint hatten, und nicht eine, die zufällig genauso lautet, so können Sie statt E auch EQ (*Exchange and Query; Austauschen mit Nachfragen*) verwenden. Bevor die Textstelle wirklich ersetzt wird, fragt ed Sie dann um Bestätigung. Nur wenn Sie y(es) antworten, wird die Stelle wirklich geändert. Die folgende Befehlszeile ersetzt z.B. das nächste Vorkommen von **Haus** durch **Hütte** — aber nur, wenn Sie die Ersetzung ausdrücklich bestätigen.

<ESC>-EQ /Haus/Hütte/

(Mit <CTRL>-G) können Sie diesen Befehl dann beliebig oft wiederholen; s.o.) Achten Sie bei diesem Beispiel-Befehl darauf, daß bei E und EQ die beiden Textstellen durch nur **ein** Begrenzerzeichen getrennt werden.

Normalerweise ist es F, E, und EQ nicht egal, wie der Text geschrieben ist, der gesucht wird. Ob im Text wirklich **Haus**, **hAuS**, **haUs** usw. steht, spielt also wirklich eine Rolle. Wollen Sie aber, daß diese Befehle nicht auf die genaue Schreibweise achten (Groß-/Klein-Schreibung also ignorieren) können Sie den Befehl UC (*Upper Case; Großschreibung*) verwenden. Wollen Sie später Kleinschreibung wieder von Großschreibung unterscheiden, so tippen Sie <ESC>LC (*Lower Case; Kleinschreibung*) ein.

10.3.6 Einfügen und Löschen von Text

Genauso wie die Befehle zur Cursor-Steuerung bieten die Befehle zum Einfügen von Text mittels Escape-Befehlen nichts wirklich neues. Text einzufügen ist mit direkten Befehlen wesentlich leichter. Wenn Sie jedoch erst einmal ihre ersten Befehlsfolgen mit Wiederholungsfaktoren erstellen, werden Sie die Befehle auch aus dieser Gruppe zu schätzen wissen. Mehr zu diesem Thema aber im folgenden Abschnitt.

Der Befehl <ESC>I <text> (*Insert; einfügen*) setzt <text> in eine neue Zeile **vor** der aktuellen (in der sich der Cursor befindet). Genauso setzt <ESC>A <text> (*Append; anhängen*) <text> in eine neue Zeile **hinter** der aktuellen. Wie üblich muß der einzufügende Text in Begrenzer-Zeichen eingeschlossen werden (s.o.).

Auch das Zerteilen und Zusammenfügen von Text-Zeilen kann mit Escape-Befehlen genauso wie mit direkten Befehlen erreicht werden. <ESC>S (*Split; teile*) beendet die aktuelle Zeile an der aktuellen Cursor-Position und fügt den Rest der Zeile in eine neue Zeile dahinter ein. <ESC>J (*Join; zusammenfügen*) fügt die Folgezeile an das Ende der aktuellen an. Die Befehlsfolge <ESC>S <ESC>J hebt sich also selbst wieder auf. Mit <ESC>D (*Delete, löschen*) kann die gesamte aktuelle Zeile, mit <ESC>B der aktuelle Buchstabe gelöscht werden. <ESC>B wirkt also z.B. wie das Drücken der -Taste.

10.3.7 Einige zeitsparende Tricks

Statt jedesmal nur einen Escape-Befehl einzutippen, können auch gleich mehrere hintereinander eingegeben werden. Mehrere Befehle in einer Zeile werden durch Semikolons getrennt. Der folgende Befehl setzt den Cursor also auf etwas umständliche Weise in die dritte Zeile (der M-Befehl wäre etwas einfacher):

```
T ; N ; N
```

Soll ein und derselbe Befehl mehrfach ausgeführt werden, so kann man dies auf zwei verschiedene Weisen erreichen. Einmal kann man das direkte Kommando <CTRL>-G verwenden, mit dem jeweils der letzte Escape-Befehl wiederholt wird. Besonders bei Such- und Änderungs-Befehlen ist das ganz praktisch, wenn z.B. die erste Fundstelle nicht das Vorkommen des Textes ist, das man eigentlich gesucht hat.

Wenn man schon vorher weiß, wie oft ein Befehl ausgeführt werden soll, schreibt man einfach eine Zahl vor den Befehl. Der folgende Befehl ändert die nächsten vier Vorkommnisse von *der* in *die* um:

```
4 E /der/die/
```

Will man mehrere Befehle zusammen mehrfach ausführen, so kann man diese Befehle, zusammen in die Befehlszeile schreiben, mit runden Klammern gruppieren und die gewünschte Anzahl von Wiederholungen vor die Klammer schreiben.

Statt einer Zahl kann man immer auch **RP** schreiben. Der Befehl (oder mehrere) wird dann so lange wiederholt, bis ein Fehler passiert. Ein solcher Fehler wäre z.B. der Befehl, eine Zeile nach oben zu gehen, wenn der Cursor bereits in der ersten Zeile ist. Der folgende Befehl ist also eine etwas umständliche Methode, um in die letzte Zeile einer Datei zu gelangen.

RP N

10.4 Liste aller ed-Befehle

Direkte Befehle:

Befehl	Wirkung
<Backspace>	Buchstaben links vom Cursor löschen.
	Buchstaben an der Cursorposition löschen
<RETURN>	Zerteilen der aktuellen Zeile in zwei
<TAB>	Cursor N Schritte nach rechts
<CTRL>-B	Löschen einer Zeile
<CTRL>-D	Text um 12 Zeilen nach unten schieben
<CTRL>-E	Cursor zum Anfang (oder Ende) des Fensters
<CTRL>-F	Groß-/Kleinschreibung ändern
<CTRL>-G	Wiederholen des letzten Escape-Befehls
<CTRL>-O	Rest des Wortes oder Leerraums löschen
<CTRL>-R	Cursor zurück hinter das vorangehende Wort
<CTRL>-T	Cursor vor zum Anfang des nächsten Wortes
<CTRL>-U	Text um 12 Zeilen nach oben schieben
<CTRL>-Y	Rest der Zeile löschen
<CTRL>-]	Cursor zum Anfang (oder Ende) der Zeile

Escape-Befehle:

Befehl	Wirkung
<ESC> A/Text/	Text hinter aktueller Zeile einfügen
<ESC> B	Cursor zum Ende des Textes
<ESC> BE	Block-Ende beim Cursor markieren
<ESC> BF/Text/	Text rückwärts suchen
<ESC> BS	Block-Start beim Cursor markieren
<ESC> CE	Cursor zum Ende der Zeile
<ESC> CL	Cursor eine Position nach links
<ESC> CR	Cursor eine Position nach rechts
<ESC> CS	Cursor zum Start der Zeile
<ESC> D	Aktuelle Zeile löschen
<ESC> DB	Block löschen
<ESC> DC	wie
<ESC> E alt/neu/	Ändere alt in neu
<ESC> EQ	Ändere alt in neu ; frage vorher um Bestätigung
<ESC> EX	Rechten Rand ausdehnen
<ESC> F/Text/	Suche Text
<ESC> I/Text/	Text vor aktueller Zeile einfügen
<ESC> IB	Block beim Cursor einfügen
<ESC> IF/Name/	Datei Name beim Cursor einfügen
<ESC> J	Aktuelle und folgende Zeile zusammenfügen
<ESC> LC	Groß-/Kleinschreibung (bei F, E und EQ) beachten
<ESC> M n	Cursor nach Zeile n
<ESC> N	Cursor in die nächste Zeile
<ESC> P	Cursor in die vorangehende Zeile
<ESC> Q	Programm ohne Sicherung des Textes beenden
<ESC> RP	Folge-Befehl wiederholen, bis Fehler passiert
<ESC> S	Wie <RETURN>
<ESC> SA	Text-Puffer auf Diskette sichern
<ESC> SB	Block zeigen
<ESC> SH	Wichtige Informationen zeigen
<ESC> SL n	Linken Rand setzen
<ESC> SR n	Rechten Rand setzen
<ESC> ST n	Tabstops setzen
<ESC> T	Cursor zum Anfang des Textes
<ESC> U	Alle Änderungen am Text rückgängig machen
<ESC> UC	Groß-/Kleinschreibung (bei F, E und EQ) <u>nicht</u> beachten
<ESC> WB/Name/	Block in Datei Name schreiben
<ESC> X	Beenden von ed; vorher Text aber sichern
<ESC> A/Text/	Text hinter aktueller Zeile einfügen

Befehl	Wirkung
<ESC> B	Cursor zum Ende des Textes
<ESC> BE	Block-Ende beim Cursor markieren
<ESC> BF/ <i>Text</i> /	<i>Text</i> rückwärts suchen
<ESC> BS	Block-Start beim Cursor markieren
<ESC> CE	Cursor zum Ende der Zeile
<ESC> CL	Cursor eine Position nach links
<ESC> CR	Cursor eine Position nach rechts
<ESC> CS	Cursor zum Start der Zeile
<ESC> D	Aktuelle Zeile löschen
<ESC> DB	Block löschen

11

Die Automatisierung des Amiga

Ein Vorteil, den das CLI gegenüber der Workbench bietet, ist die Möglichkeit, langwierige, häufig anfallende Aufgaben zu automatisieren. Hierzu wird eine Folge von CLI-Befehlen zu einer sogenannten *Kommandofolge* zusammengefaßt. Eine solche Kommandofolge erhält einen Namen und kann mit diesem Namen durch den Befehl *execute* gestartet werden. Sie läuft dann selbständig und normalerweise ohne weiteres Zutun des Benutzers ab. Mit Hilfe des Befehls *run* kann man die meisten Kommandofolgen sogar als separaten Task im Hintergrund ablaufen lassen und sich unterdessen um andere Dinge kümmern.

Kommandofolgen sind einfachen Programmen sehr ähnlich. Keine Angst vor der Programmierung; die Programmiersprache für Kommandofolgen ist sehr einfach! Sie kennt aber schon bedingte Anweisungen, Sprunganweisungen, Parameter und Variablen. Durch die Möglichkeit, in Kommandofolgen andere Kommandofolgen — und natürlich normale Kommandos — aufzurufen, können auch größere Aufgaben angegangen werden, indem man ein großes Problem in kleine Teilaufgaben zerlegt.

11.1 Aufgabe von Kommandofolgen

Der Sinn von Kommandofolgen wird Ihnen vielleicht nicht gleich klar sein. Bedarf für sie taucht üblicherweise erst dann auf, wenn man intensiv mit dem Computer arbeitet. Sie werden rasch merken, daß viele Dinge, die Sie täglich tun, sich immer wiederholen oder sich zumindest ähneln.

Nehmen wir einmal an, Sie würden recht häufig die drei folgenden CLI-Befehle hintereinander ausführen:

```
sort    kundenliste TO druckdatei
type    druckdatei  TO PRT:
delete  druckdatei
```

Dann ist es doch eigentlich recht lästig, daß Sie jedesmal alle drei Befehle fehlerfrei eintippen müssen, nach jedem Befehl darauf warten müssen, daß er fertig wird und alles in allem eine stupide mechanische Tätigkeit ausführen müssen, für die ein Computer viel besser geeignet ist.

Genau dies leisten Kommandofolgen! Wie wir im folgenden sehen werden, können wir eine Kommandofolge schreiben, die das selbe leistet wie die drei oben gezeigten Zeilen und nur mit einem einzigen Befehl aufgerufen werden kann. Falls die Datei, die sortiert und gedruckt werden soll, nicht immer denselben Namen hat, kann die Kommandofolge auch daran flexibel angepaßt werden. Kurz: Kommandofolgen sparen Tipparbeit, machen Fehler unwahrscheinlicher und nehmen uns stupide Arbeit ab.

11.2 Eine einfache Kommandofolge

Das Format von Kommandofolgen ist denkbar einfach. Man tippt einfach die CLI-Befehle in genau derselben Form, wie man sie auch ins CLI-Fenster getippt hätte, in eine Textdatei ein. Idealerweise macht man dies mit einem Texteditor, z.B. *ed*, der ja im vorangehenden Kapitel bereits vorgestellt wurde.

Für die folgenden Versuche starten Sie nun bitte das CLI – falls Sie sich noch nicht darin befinden. Richten Sie mit *makedir df0:test* ein neues Dateiverzeichnis namens *test* ein. Danach machen Sie bitte *test* mit *cd df0:test* zum aktuellen Dateiverzeichnis. Dieses Dateiverzeichnis wird die Dateien aufnehmen, die für die folgenden Versuche benötigt werden.

Sie werden nun ihre erste Kommandofolge erstellen. Dazu werden Sie den Texteditor *ed* anwenden. Falls Sie beim Eintippen der folgenden Zeilen Fehler machen sollten, benutzen Sie bitte die Korrekturmöglichkeiten von *ed*, die im Kapitel *Der Texteditor ed* beschrieben wurden. Starten Sie zunächst *ed* mit *ed ZeigeText* und tippen dann die folgenden Zeilen ein:

```
echo "Start der Kommandofolge"  
copy text TO kopie  
type kopie OPT N  
delete kopie  
echo "Kommandofolge beendet!"
```

Verlassen Sie nun **ed** und sichern Ihr Werk mit **<ESC>x**. Wenn Sie nun **cd** tippen, werden Sie Ihre erste Kommandofolge unter dem Namen **ZeigeText** angezeigt bekommen. Geben Sie nun den Befehl **execute ZeigeText** ein und die Kommandofolge **ZeigeText** wird gestartet. Sie wird aber schon in der zweiten Zeile abgebrochen, da der Befehl **copy text TO kopie** nicht funktionieren kann. Es gibt ja keine Datei namens **text** im aktuellen Dateiverzeichnis die kopiert werden könnte. Diese müssen wir erst einmal erzeugen.

Dazu gehen wir mit **ed text** zurück in den Texteditor **ed** und tippen einnige sinnvolle oder sinnlose Zeilen ein. Was diese enthalten bleibt völlig Ihnen überlassen. Verlassen Sie danach wieder **ed** mit **<ESC>x** und starten Sie erneut Ihre Kommandofolge mit **execute ZeigeText**. Es sollten nun die folgenden Zeilen auf Ihrem Bildschirm erscheinen:

```
1>execute ZeigeText  
Start der Kommandofolge  
"Hier erscheinen die von Ihnen einge-"  
"gebenen Zeilen der Datei text mit"  
"vorangestellten Zeilennummern ....."  
Kommandofolge beendet!  
1>
```

Die Kommandofolge **ZeigeText** hat also soeben selbstständig und ohne Ihr Zutun fünf CLI-Befehle ausgeführt. Nicht schlecht für den Anfang! Nur ist die gezeigte Kommandofolge noch sehr unflexibel. Sie kann nur eine einzige Datei, die den Namen **text** hat, ausgeben. Wollen Sie einmal eine andere Datei ausgeben, so müssten Sie dafür eine neue Kommandodatei schreiben, die sich kaum von der ersten unterscheiden würde. Nur der Name **text** müsste bei jedem Vorkommen durch einen anderen ersetzt werden. Eine sehr lästige Sache.

Gottseidank kann aber der **execute**-Befehl diese Ersetzung für uns übernehmen. Wir schreiben dazu eine Kommandofolge, die eine beliebige Datei ausgeben kann, der wir dazu aber beim Aufruf den Namen der Kommandofolge als *Parameter* übergeben müssen.

11.3 Kommandofolgen mit Parametern

Um Parameter in unsere Kommandofolge "einzubauen", müssen wir zwei Dinge tun. Wir müssen am Anfang der Kommandofolge dem *execute*-Befehl mitteilen, welche Parameter mit welchen Namen die Kommandofolge benötigt. In der Kommandofolge selbst muß dann noch der Name *text* gegen einen Platzhalter ersetzt werden. Diesen Platzhalter ersetzt *execute* dann, bevor es die Kommandofolge wirklich ausführt, gegen den Namen, den Sie für diesen Zweck angeben. Die Kommandofolge *ZeigeText* sieht nun aus wie folgt:

```
.KEY datei/a
echo "Start der Kommandofolge"
copy <datei> TO kopie
type <datei> OPT N
delete kopie
echo "Kommandofolge beendet!"
```

Wie Sie sehen, ist überall, wo vorher *text* stand, nun der Platzhalter *<datei>* aufgetaucht. Ferner ist eine zusätzliche erste Zeile eingefügt worden, die angibt, welche Parameter die Kommandofolge benötigt. Führen Sie nun bitte die nötigen Änderungen an der Datei *ZeigeText* mit Hilfe von *ed* durch.

Die neue erste Zeile von *ZeigeText* sieht deutlich anders aus als die restlichen Zeilen. Sie beginnt z.B. mit einem Punkt, was sie als eine besondere Zeile kennzeichnet, die nicht als Befehl ausgeführt werden darf, sondern spezielle Informationen für *execute* enthält. *Execute* sucht vor der Ausführung jeder Kommandofolge nach solchen Zeilen und wertet sie aus. Das Wort *KEY* nach dem Punkt bedeutet, daß der Rest der Zeile die Spezifikation eines Parameters enthält. In diesem Fall lautet sie *.KEY datei/a*, was bedeutet, daß *ZeigeText* einen Parameter namens *datei* hat und dieser Parameter notwendig ist, also beim Aufruf nicht fortfallen darf. Letzteres regelt der Zusatz */a*, der nicht zum eigentlichen Namen des Parameters gehört. Stünde hinter *datei* kein */a*, so wäre *datei* ein optionaler Parameter, was unter Umständen auch recht sinnvoll sein kann. Mehr darüber aber weiter unten.

Probieren Sie nun einmal das neue *ZeigeText* aus. Wenn Sie es versuchen wie bisher, erhalten Sie sofort eine Fehlermeldung, die besagt, daß Sie einen Parameter vergessen haben. Tippen Sie aber *execute ZeigeText text*, so erhalten Sie am Bildschirm dasselbe Ergebnis wie früher mit *execute ZeigeText*. Probieren Sie auch einmal *execute ZeigeText ZeigeText*

aus. Ihnen wird dann am Bildschirm der Inhalt der Kommandofolge selbst gezeigt.

Kommen wir nun aber zum trockeneren Teil dieses Kapitels. Bisher haben Sie ja nur am Beispiel gesehen, wie man Parameter verwendet. Nun sollen Ihnen auch alle Details offenbart werden.

Wie schon oben zu sehen, gehört zu jeder Kommandofolge mit Parametern eine KEY-Zeile. Eine solche Zeile teilt *execute* mit, welche Parameter benötigt werden, wie sie heißen, behandelt werden müssen und ob sie optional oder notwendig sind. In einer KEY-Zeile können gleichzeitig mehrere Parameter spezifiziert werden. Die einzelnen Parameter werden dann durch Kommata getrennt. Die folgende Zeile spezifiziert z.B. drei (optionale) Parameter:

```
.KEY datei,disk,name
```

Hinter dem Namen jedes Parameters können, von einem Schrägstrich getrennt, zwei verschiedene Buchstaben stehen, die angeben, wie der Parameter zu behandeln ist. Der Zusatz */A* bedeutet, daß der Parameter nötig ist, also beim Aufruf unbedingt angegeben werden muß (mehr zu optionalen Parametern weiter unten). Der Zusatz */K* bedeutet, daß der Parameter immer von seinem Namen eingeleitet werden muß. Würde die erste Zeile von *ZeigeText* *.KEY datei/A/K* lauten, so wäre *execute ZeigeText text* kein legaler Aufruf der Kommandofolge mehr. Die korrekte Schreibweise wäre nun *execute ZeigeText DATEI text* (die Großschreibung von *datei* dient nur der besseren Lesbarkeit). Der Sinn dieser auf den ersten Blick etwas umständlichen Einrichtung liegt in der besseren Lesbarkeit eines Aufrufs — besonders, wenn optionale Parameter ins Spiel kommen.

Hat eine Kommandofolge drei optionale Parameter und Sie übergeben nur zwei, hat es *execute* schwer, "sich zu entscheiden". Haben Sie den ersten und zweiten Parameter übergeben und den dritten freigelassen? Oder handelt es sich um den ersten und dritten und der zweite soll freibleiben? Oder sind es gar der zweite und dritte Parameter und der erste bleibt unbelegt? Schreiben Sie aber vor die beiden Parameter auch noch ihre Namen, wird die Sache eindeutig. Sie können dann sogar die Reihenfolge austauschen und erst den zweiten, dann den ersten Parameter angeben, usw.. Die Namen machen die ganze Sache eindeutig.

Nachdem Sie nun einen oder mehrere Parameter spezifiziert haben, können sie in der Kommandofolge verwendet werden. An den Stellen, an denen der Text erscheinen soll, den der Benutzer für einen bestimmten Parameter angegeben hat, tippen Sie den Namen des Parameters in spitzen Klammern, z.B. *<datei>*. Beim Aufruf der Kommandofolge wird dann jedes

Vorkommen des Platzhalters **<datei>** gegen **text** ersetzt, wenn Sie **execute ZeigeText text** eingegeben haben.

Problematisch wird diese Ersetzung bei optionalen Parametern, für die kein Wert angegeben wurde. Gegen welchen Text sollen die entsprechenden Platzhalter ersetzt werden? Wenn Sie nichts "besonderes" unternehmen, ersetzt **execute** solche Platzhalter gegen den leeren Text. Dies dürfte manchmal sinnvoll sein, oft aber auch nicht. Hierfür gibt es eine besondere Einrichtung, die sogenannten *Defaultwerte*. Ein Defaultwert wird immer dann für einen Platzhalter eingesetzt, wenn dem entsprechenden Parameter beim Aufruf der Kommandofolge kein Wert zugeteilt wurde. Es gibt zwei Möglichkeiten, einem Parameter einen Defaultwert zuzuteilen.

Die Zeile **.DEF datei text** ordnet dem Parameter **datei** z.B. den Defaultwert **text** zu. Das allgemeine Format einer solchen DEF-Zeile sieht dabei aus wie folgt:

```
.DEF <Parametername> <Defaultwert>
```

Diese Zeile ordnet den Defaultwert für den Rest der Kommandofolge dem Parameter zu. Jedes Vorkommen des entsprechenden Platzhalters wird dann, natürlich nur, falls kein anderer Wert beim Aufruf übergeben wurde, gegen den Defaultwert ersetzt. Flexibler ist die Möglichkeit, direkt beim Platzhalter einen Defaultwert anzugeben. Jeder Platzhalter eines Parameters kann so einen unterschiedlichen Defaultwert haben. Hierzu schreibt man hinter dem Namen des Parameters im Platzhalter den Defaultwert, getrennt durch ein Dollarzeichen. **<datei\$text>** hat also für diesen einen Platzhalter dieselbe Wirkung, wie **.DEF datei text** für die ganze Kommandofolge. Er wird durch **text** ersetzt, falls kein anderer Wert beim Aufruf übergeben wurde.

Diese Möglichkeit soll jetzt sofort für die erste Beispiel-Kommandofolge eingesetzt werden. Hierzu wird sie nun geändert, bis sie das folgende Aussehen hat:

```
.KEY datei
.DEF datei text
echo "Start der Kommandofolge"
copy <datei> TO kopie
type <datei> OPT N
delete kopie
echo "Kommandofolge beendet!"
```

Es muß also nur in die erste Zeile der alten Datei /a gelöscht werden und die zweite Zeile so eingefügt werden, wie sie hier gezeigt wurde. Falls Sie nicht mehr wissen, wie die Änderungen auszuführen sind, lesen sie einfach noch einmal im Kapitel *Der Texteditor* ed nach.

Nun sollten Sie Ihre neue und verbesserte Kommandofolge aber auch ausprobieren. Verlassen Sie dazu den `ed`, falls nötig, mit `<ESC>x` und tippen Sie dann *execute ZeigeText*. Wie in der allerersten Version der Kommandofolge *ZeigeText* wird Ihnen nun der Inhalt der Datei *text* gezeigt. Tippen Sie aber *execute ZeigeText* und dahinter den Namen einer anderen Textdatei, so wird diese angezeigt. Defaultwerte sind ganz praktisch, nicht wahr?

Damit wissen sie nun schon fast alles, was über Parameter zu sagen wäre. Außer den zwei Punkt-Befehlen `.KEY` und `.DEF` gibt es aber noch eine Reihe anderer, die zwar nicht häufig benötigt werden, die aber der Vollständigkeit halber nun auch noch aufgelistet werden. Wie `.KEY` und `.DEF` müssen auch sie alle am Beginn einer neuen Zeile stehen, sonst erkennt `execute` sie nicht als Punkt-Befehle.

`.K` ist eine Abkürzung für `.KEY`.

`.DOT` und dahinter ein einzelner Buchstabe führt für den Rest der Kommandofolge dazu, daß nicht mehr der Punkt, sondern der hier angegebene Buchstabe zur Einleitung von Punkt-Befehlen dient. Nach `.DOT !` wird also `.DEF` nicht mehr erkannt; man muß statt dessen `!DEF` schreiben.

`.BRA` und dahinter ein einzelner Buchstabe führt für den Rest der Kommandofolge dazu, daß nicht mehr das Kleiner-Zeichen `<` zur Einleitung von Platzhaltern dient. Statt dessen müssen Platzhalter für Variablen jetzt mit dem angegebenen Buchstaben beginnen.

`.KET` und dahinter ein einzelner Buchstabe führt für den Rest der Kommandofolge dazu, daß nicht mehr das Größer-Zeichen `>` zur Beendigung von Platzhaltern dient. Statt dessen müssen Platzhalter für Variablen jetzt mit dem angegebenen Buchstaben enden.

`.DOL` oder `.DOLLAR` und dahinter ein einzelner Buchstabe führt für den Rest der Kommandofolge dazu, daß nicht mehr das Dollar-Zeichen `$` zur Trennung von Parameternamen und Defaultwerten in Platzhaltern dient. Statt dessen müssen Parameternamen und Defaultwerte in Platzhaltern jetzt durch den angegebenen Buchstaben getrennt werden.

`<Leerzeichen>`, also ein Punkt am Anfang der Zeile, gefolgt von einem Leerzeichen, erklären den Rest der Zeile zum Kommentar, den weder `execute` noch das CLI berücksichtigen. Endet die Zeile sofort hinter dem Punkt, bedeutet dies ebenfalls einen Kommentar — in diesem Fall eine Leerzeile, die der besseren Lesbarkeit halber eingefügt wurde.

In diesem Abschnitt haben Sie schon eine recht komfortable Kommandofolge konstruiert. Sie hat aber immer noch einige Mängel oder Schwächen.

Wäre es z.B. nicht besser, wenn die Kommandofolge auch selbst merken würde, ob die Datei, die Sie kopieren und ausdrucken will, auch existiert und eine entsprechende Fehlermeldung ausgeben würde, falls nicht? Eine solche Kommandofolge soll im nächsten Kapitel erstellt werden.

11.4 Bedingte Anweisungen und Sprünge

Die erste Kommandofolge, die Sie bis jetzt kennengelernt haben, ist extrem gradlinig. Egal was geschieht, folgt sie stur der Abfolge von CLI-Befehlen, wie Sie sie eingegeben haben. Dies ist nicht immer wünschenswert. Wenn während der Ausführung einer Kommandofolge z.B. eine Fehlersituation auftritt, so kann es sinnvoll sein, die Kommandofolge nicht einfach abubrechen. Manche Fehler können vielleicht ausgebügelt werden; zumindest sollte dem Benutzer aber in sinnvoller Form die Fehlerursache mitgeteilt werden. Unter Umständen kann es auch sinnvoll sein, zwei Fliegen mit einer Klappe zu schlagen, wenn man eine Kommandofolge schreibt — also zwei verschiedene Aufgaben zu erledigen, je nachdem welche Parameter man ihr übergibt. Beides ist nur dann möglich, wenn man Entscheidungspunkte in eine Kommandofolge einbauen kann, an denen man in Abhängigkeit vom Wert bestimmter Parameter entscheiden kann, ob man eine oder eine andere Folge von Befehlen durchführt.

Zunächst soll die Kommandofolge *ZeigeText* deshalb nachsehen, ob die auszugebende Datei überhaupt existiert. Die dafür nötige Änderung ist relativ simpel. Sie zeigt eine einfache Anwendung des *if*-Kommandos (*if*, engl. für *wenn, falls*):

```
.KEY datei
.DEF datei text
echo "Start von ZeigeText"
IF EXISTS <datei>
copy <datei> TO kopie
type <datei> OPT N
delete kopie
echo "ZeigeText erfolgreich beendet!"
ELSE
echo "Fehler: <datei> existiert nicht!"
ENDIF
```

(Die für Kommandofolgen spezifischen Befehle wie *if* werden übrigens in allen Auflistungen von Kommandofolgen von nun an immer groß geschrieben, um ihre Bedeutung hervorzuheben.) *ZeigeText* zeigt nun immer noch den Inhalt einer Datei im CLI-Fenster, falls es diese Datei überhaupt gibt. Falls Sie aber nicht existiert, bricht nicht etwa die Kommandofolge ab, sondern es wird eine sinnvolle Fehlermeldung ausgegeben, die dem Anwender das Problem verdeutlicht. Der normale Ablauf der Kommandofolge, wie wir ihn bisher kannten, wird dazu in eine *if*-Anweisung eingeschlossen. Obwohl *if*, *else* und *endif* einzelne CLI-Kommandos sind, werden sie immer zusammen verwendet und sollen deshalb auch hier zusammen als *if*-Anweisung erläutert werden. Das Format einer *if*-Anweisung lautet:

```
IF <Bedingung>
    andere CLI-Befehle...
...
ELSE
    andere CLI-Befehle...
...
ENDIF
```

Wenn die Liste der CLI-Befehle hinter *else* leer ist, so kann man *else* auch ganz wegfassen lassen und die Anweisungsfolge sofort mit *endif* beenden. Die Vorgehensweise, wenn bei der Ausführung einer Kommandofolge auf ein *if* getroffen wird, ist nun die folgende:

Die Bedingung hinter dem *if* wird überprüft. Falls sie zutrifft, so werden die Anweisungen hinter dem *if* ausgeführt, bis man auf ein *else* oder *endif* trifft, je nachdem, was zuerst eintritt. Trifft man auf ein *else*, so werden alle Kommandos dahinter bis zum *endif* ignoriert. Hinter dem *endif* wird aber auf alle Fälle wieder mit der Ausführung der Kommandos fortgefahren.

Trifft die Bedingung hinter *if* aber nicht zu, so werden alle Anweisungen ignoriert, bis man auf ein *else* oder *endif* trifft. Dahinter wird auf alle Fälle wieder mit der Ausführung der Kommandos fortgefahren. Eine *if*-Anweisung hat so also zwei Zweige, die alternativ durchlaufen werden. Entweder die Kommandos in dem Zweig hinter dem *if* oder die Kommandos in dem anderen Zweig hinter dem *else* werden durchgeführt. Fällt der Zweig hinter dem *else* ganz fort, so kann man auch von einer *bedingten Anweisungsfolge* sprechen. Nur wenn die Bedingung zutrifft, wird sie ausgeführt — sonst nicht.

Diese Erklärung ist allerdings etwas vereinfacht. *If*-Anweisungen können nämlich geschachtelt sein. Deshalb gehört nicht immer das unmittelbar nächste *else* zu einem *if*, wenn im *if*-Zweig erneut ein *if* auftaucht.


```
IF <Bedingung>
  andere CLI-Befehle...
.
IF <Bedingung> ; inneres IF
  andere CLI-Befehle...
ELSE           ; inneres ELSE
  andere CLI-Befehle...
ENDIF         ; inneres ENDIF
.
ELSE
  andere CLI-Befehle...
...
ENDIF
```

Die Zuordnung von einem *else* oder *endif* zu einem bestimmten *if* ist ähnlich, wie mit geschachtelten Klammern und wohl auch recht intuitiv zu verstehen.

In unserem Beispiel sieht die abgeprüfte Bedingung umgangssprachlich etwa folgendermaßen aus: Wenn (*if*) die auszugebende Datei vorhanden ist, kopiere sie und gib sie aus, sonst (*else*) melde einen Fehler (*endif*). Die Bedingung **EXISTS <datei>** ist dabei nur eine von vielen, die *if* kennt.

Weitere Bedingungen sind z.B. **WARN**, **ERROR** und **FAIL**, die dann zutreffen, wenn der letzte CLI-Befehl fehlschlug und der letzte Fehlercode größer oder gleich 5 (bei **WARN**), 10 (bei **ERROR**) oder 20 (bei **FAIL**) ist. Jeder fehlgeschlagene CLI-Befehl setzt einen solchen Fehlercode (für den man mittels *why* ja auch eine Erläuterung erhalten kann). Mit diesen Bedingungen kann man abprüfen, ob ein Fehler passiert ist und dann geeignete Gegenmaßnahmen ergreifen oder wenigstens den Benutzer informieren. Bei allen drei Bedingungen muß man aber berücksichtigen, daß sie abhängig von dem aktuellen Abbruch-Level sind, der mit *failat* geändert werden kann. Ist der Fehlercode größer als der aktuelle Abbruch-Level, so wird die Ausführung einer Kommandofolge immer abgebrochen. Steht der Abbruch-Level beispielsweise bei 5 und es tritt ein Fehler ein, der einen Fehlercode von 10 hat, so bricht jede Kommandofolge ab, bevor sie noch mit **IF ERROR** überhaupt den Fehler erfragen kann. Vor jedem Befehl, der einen Fehler verursachen könnte, den man selbst behandeln will, sollte man deshalb mit *failat* den Abbruch-Level hochsetzen. (Mehr dazu aber weiter unten.)

Eine weitere Bedingung erlaubt den Vergleich zweier Texte und trifft dann zu, wenn beide gleich sind. Beim Vergleich wird Groß- oder Kleinschreibung ignoriert; *A* ist gleich *a*. Hiermit wird man üblicherweise wohl die Werte von Parametern mit bestimmten Texten vergleichen und zwei Parameter

miteinander. Das Format für einen solchen Vergleich lautet: **<text1> EQ <text2>** (**EQ** steht dabei für *equal* oder auf deutsch gleich). Ein möglicher Vergleich dieser Art wäre z.B. **IF <datei> EQ "text"**.

Und schließlich kann man, indem man vor eine Bedingung das Schlüsselwort **NOT** (engl. für *nicht*) setzt, deren Bedeutung umkehren. **IF NOT "text" EQ "text"** trifft z.B. nicht zu und führt dazu, daß der erste Zweig der entsprechenden **if**-Anweisung nicht ausgeführt wird.

Eine **if**-Anweisung kann unter Umständen auch als ein Sprung bzw. ein Überspringen von Teilen einer Kommandofolge aufgefaßt werden. Trifft eine Bedingung nicht zu, werden ja die Anweisungen zwischen **if** und **else** oder **endif** übersprungen. Eine etwas direktere Möglichkeit, Anweisungen zu überspringen, stellt das **skip**-Kommando dar. Trifft man innerhalb einer Kommandofolge auf ein **skip**, so bedeutet dies ein Überspringen (engl. *skip*) aller Kommandos in der Folge, bis zu einem **lab**-Kommando (**lab** für *label*; zu deutsch *Marke*). Sowohl **skip** als auch **lab** haben einen Parameter, der aus einem beliebigen Text bestehen kann. **Skip** springt immer bis zu einem **lab**, das den selben Parameter hat. Dazwischenliegende **labs** mit anderen Parametern werden ebenfalls übersprungen, wie andere Befehle auch.

SKIP Ende

bel. CLI-Befehle...

...

LAB Ende ; Sprung bis hier

bel. CLI-Befehle...

Die letzte "Sprunganweisung" ist schließlich **quit**. Dies ist ein recht brutaler Sprung, mit dem die gesamte Kommandofolge verlassen wird, als wäre ihr Ende erreicht worden. Der Befehl **quit** besitzt einen optionalen Parameter, der ein Fehlercode (eine Zahl) sein sollte. Ist er nicht vorhanden, wird der aktuelle Fehlercode auf Null gesetzt, als wäre kein Fehler passiert. Besonders wenn die Möglichkeit besteht, daß eine Kommandofolge von anderen Kommandofolgen aufgerufen wird, sollte man im Fehlerfall immer einen Fehlercode mit **quit** "nach außen" geben, damit eine solche andere Kommandofolge eine Chance hat, auf einen eventuellen Fehler zu reagieren.

Ein Teil der eben vermittelten Informationen soll nun aber auch in die Praxis umgesetzt werden! Die aktuelle Version von ZeigeText meldet zwar korrekt, wenn die auszugebende Datei nicht existiert; andere Fehler können aber passieren, werden nicht erkannt und führen zum Abbruch der Kommandofolge. Die kritische Stelle ist dabei der **copy**-Befehl, der vor der Ausgabe stattfindet. Wenn die Diskette voll oder schreibgeschützt ist, wird er fehlschlagen. Diese Stelle soll nun etwas entschärft werden.


```
.KEY datei
.DEF datei text
echo "Start von ZeigeText"
failat 25
copy <datei> TO kopie      ; GEFÄHRLICH
IF ERROR
SKIP fehler
ENDIF
failat 10

.
.   Alles OK
type <datei> OPT N
delete kopie
echo "ZeigeText erfolgreich beendet!"
QUIT

.
.   Fehler beim Kopieren
LAB fehler
IF EXISTS <datei>
echo "Fehler: <datei> existiert nicht!"
ELSE
echo "Fehler: Kopieren von <datei> fehlgeschlagen!"
echo "Ist die Diskette voll oder schreibgeschützt ?"
ENDIF
QUIT 10
```

Die einfache kurze Kommandofolge, mit der wir begonnen haben, ist nun schon recht lang geworden. Sie nutzt fast alle Möglichkeiten, die eine Kommandofolge überhaupt nutzen kann.

Zunächst wird die kritische Stelle, der *copy*-Befehl, mit einem *failat 25* versehen. Selbst wenn der *copy*-Befehl nun fehlschlägt, läuft die Kommandofolge weiter. Unmittelbar hinter *copy* wird nun geprüft, ob der Befehl mißlungen ist und falls ja, zur Marke *Fehler* gesprungen. Falls kein Fehler auftrat, wird die schon bekannte Befehlsfolge durchlaufen und schließlich mit *quit* die Kommandofolge verlassen. Vorher aber wird der Abbruch-Level mit *failat 10* wieder auf den Normalzustand zurückgesetzt. Das ist in diesem Fall nicht unbedingt nötig. Man sollte es aber stets tun, wenn der Fehlercode zuvor hochgesetzt wurde, damit nicht spätere Fehler, die man nicht vorhergesehen hat, ignoriert werden.

Zur Marke *Fehler* kommt die Kommandofolge überhaupt nur, wenn ein Fehler aufgetreten ist. Es wird dann geprüft, ob die Datei existiert oder nicht. Wenn nicht, wird eine entsprechenden Fehlermeldung ausgegeben. Wenn

doch, ist die Fehlerursache unklar. Dem Benutzer werden dann zwei Fehlermöglichkeiten genannt. Im Fehlerfall wird schließlich noch ein Fehlercode mit *quit* ausgegeben. Dies ist im Moment nicht unbedingt nötig, aber guter Stil, den man sich "für später" angewöhnen sollte.

12

Tips für das CLI

In diesem Kapitel werden Sie einige Informationen und Tips finden, die sonst in keines der Kapitel über das CLI hineinpassen. Viele davon sind nur für fortgeschrittene Anwender gedacht und deshalb mit Vorsicht zu genießen. Aus dem CLI heraus ist es wirklich recht einfach möglich, seine Diskette dauerhaft zu beschädigen, während man sich auf der Workbench dafür schon einige Mühe geben muß.

12.1 Die Startup-Sequence

Für einen häufigen Anwender des CLI wäre es doch ganz praktisch, wenn nach dem Start bereits alle Einstellungen, z.B. das aktuelle Dateiverzeichnis, einige logische Laufwerke, usw., so wären, wie er sie üblicherweise benötigt. Gerade für Programmierer, die im CLI mit vielen Dateien jonglieren müssen, würde das einige Handarbeit nach jedem System-Start sparen.

12.1.1 Das Original

Genau dies ist möglich. Beim Start des Amiga ist zunächst einmal — vielleicht im Gegensatz zu Ihren bisherigen Annahmen — das CLI aktiv. Es führt dann aber sofort die Kommandofolge *startup-sequence* im Dateiverzeichnis *s* auf der Start-Diskette aus. Erst durch diese Kommando-folge gelangt man "normalerweise" in die Workbench. Wenn Sie einmal Ihre

frisch erworbenen Fähigkeiten mit **ed** ausprobieren und in dieser Datei hineinschauen, werden sie darin die folgenden Zeilen finden:

```
echo "Workbench disk. Release 1.1"
echo " "
echo "Use Preferences tool to set date"
echo " "
LoadWB
endCLI > nil:
```

(Wie auch bisher in diesem Buch üblich, bezieht sich dies auf die englische Version des Amiga. In der deutschen Version sind die am Bildschirm erscheinenden Texte natürlich übersetzt. Die Namen der CLI-Befehle — und die Bedeutung der Texte — bleiben aber gleich.)

Die beiden entscheidenden Zeilen in dieser Kommandofolge enthalten die Befehle **LoadWB** (*Load Workbench; Workbench starten*) und **endcli**. **LoadWB** startet die Workbench und erst in diesem Moment wird diese komfortablere Benutzerschnittstelle überhaupt erst verfügbar. Danach wird das CLI mit **endcli** geschlossen wobei durch **>nil:** die Abschiedsmeldung des CLI ins Nichts geschickt wird.

12.1.2 Modifikationen

Wie sie sehen ist das CLI wirklich die "originale" Benutzerschnittstelle des Amiga. Wenn Sie im **ed** die beiden Zeilen **loadwb** und **endcli** löschen, so bleiben Sie im CLI und kommen erst gar nicht auf die Workbench.

Alternativ kann man auch nur den Befehl **endcli** löschen, was besonders dann sinnvoll ist, wenn man häufig zwischen beiden Möglichkeiten, den Amiga zu bedienen, wechseln will. Löscht man **endcli**, aber nicht **loadwb**, so hat man von Anfang an sowohl die Workbench als auch das CLI zur Verfügung. Das CLI-Fenster verdeckt allerdings zunächst die Workbench. Sie müssen es zuerst kleiner machen, um die Disketten-Icons usw. zu sehen.

Eine andere — weniger endgültige — Möglichkeit, im CLI zu bleiben, ist das Abbrechen der Kommandofolge während Ihrer Ausführung. Jede Kommandofolge — also auch **startup-sequence** — kann mit <CTRL>-D abgebrochen werden. Drücken Sie diese Tastenkombination also rechtzeitig, bevor der **endcli**-Befehl ausgeführt wird, bleiben Sie im CLI, ohne **startup-sequence** modifizieren zu müssen.

12.1.3 Anderweitige Nutzung

Statt kleine Modifikationen an *startup-sequence* vorzunehmen, können Sie die Standardversion natürlich auch ganz löschen und durch eine eigene Version ersetzen, die die Aufgaben durchführt, die Sie üblicherweise immer beim Start des Amiga durchführen.

Falls Sie im CLI bleiben wollen, so können dies z.B. einige *cd*- und *assign*-Befehle sein. Für Benutzer des CLIs und der Workbench könnte es interessant sein, bei jedem Start *preferences* aufzurufen, um die richtige Uhrzeit und das korrekte Datum einzustellen.

Falls Sie nur auf das richtige Datum Wert legen, können Sie es sich sogar noch einfacher machen. Der *date*-Befehl im CLI kennt einige besondere Tricks, um das Datum ausgehend vom letzten gespeicherten Datum zu setzen. Diese Tricks sparen oft einige Tipparbeit.

Wenn Sie z.B. täglich Ihren Amiga einmal starten, so könnten Sie den Befehl *date tomorrow* in die *startup-sequence* setzen. Das Datum wird dann bei jedem Start automatisch einen Tag höher gesetzt (*tomorrow* = *morgen*).¹

Falls Sie über längere Zeit hinweg immer nur mit einem Werkzeug der Workbench arbeiten, z.B. mit dem Malprogramm *Graphicraft* oder der Textverarbeitung *Textcraft*, so können Sie dieses Werkzeug auch gleich direkt vom CLI aus starten und brauchen nicht erst auf der Workbench Disketten und Schubladen zu öffnen.

Dies alles waren aber nur Vorschläge. Im Laufe der Zeit werden Sie selbst bestimmt noch viele andere nützliche Anwendungen für *startup-sequence* finden.

12.2 Die RAM-Disk

Eines der nützlichsten Hilfsmittel des CLIs ist die RAM-Disk. Diese Diskette, die stets den Namen RAM: trägt, wird im Speicher des Amiga simuliert und

¹ Eine weitere Möglichkeit, das Datum schnell zu stellen ist die Verwendung eines (englischen) Wochentagsnamen statt *tomorrow*. *Date* stellt dann das interne Datum auf den nächsten Tag, der auf diesen Wochentag fällt. Der Befehl *date tuesday* stellt das Datum beispielsweise auf den nächsten Dienstag.

ist deshalb viel schneller, als es jede echte Diskette oder Festplatte sein kann. Zudem haben Sie mit der RAM-Disk ein zusätzliches Diskettenlaufwerk zur Verfügung. Wenn alle Disketten, die sich in den wirklichen Laufwerken befinden, schon voll sind, bietet die RAM-Disk oft willkommenen Ausweichplatz. Eine (vorsichtige) Anwendung der RAM-Disk kann Ihnen deshalb viel Zeit und Mühen ersparen.

12.2.1 Programme und Dateien in der RAM-Disk

Jedes Programm, das in die RAM-Disk kopiert wurde, kann ungleich schneller gestartet werden. Genauso geht es oft wesentlich schneller, wenn man Dateien, mit denen Programme arbeiten sollen, erst einmal auf die RAM-Disk kopiert, sie dort bearbeitet und dann wieder zurück auf die Diskette kopiert (der letzte Schritt ist natürlich nur dann nötig, wenn Änderungen an den Dateien vorgenommen wurden).

Auch die Kommandos des CLIs sind ja Programme. Jedesmal, wenn Sie ein solches Kommando aufrufen, muß dieses Programm erst von der Diskette geladen werden. Deshalb dauert es oft einen kleinen Moment, bis das CLI auf einen Befehl reagiert. Eine mögliche Nutzung der RAM-Disk wäre es deshalb, einen Teil der am häufigsten benutzen CLI-Befehle auf die RAM-Disk zu kopieren und dann das logische Laufwerk C: "umzulegen". Die nachstehende Folge von CLI-Befehlen (aus der man natürlich eine Kommandofolge machen könnte) leistet genau dies.

```
makedir          ram:c
copy :c/copy      ram:c/copy  ; copy nun schneller
assign C:         ram:c       ; WICHTIG !
copy :c/dir       ram:c/dir
copy :c/cd        ram:c/cd
copy :c/delete    ram:c/delete
copy :c/list      ram:c/list
copy :c/assign    ram:c/assign
copy :c/execute   ram:c/execute
copy :c/run       ram:c/run
..... ; usw.
```

Wichtig ist bei dieser Kommandofolge der *assign*-Befehl, der dem CLI sagt, wo die Kommandos zu finden sind. Würde er wegfallen, lägen die ausgesuchten Kommandos zwar auf der RAM-Disk, das CLI würde aber nach wie vor die Kommandos auf der Diskette benutzen.

Befehle auf der RAM-Disk haben nicht nur den Vorteil größerer Geschwindigkeit. Sie können nun die Workbench- (bzw. CLI-) Diskette herausnehmen und eine andere Diskette dafür einlegen, mit der Sie vielleicht arbeiten wollen. Normalerweise würde das CLI nun bei jedem *dir* oder *copy* wieder die Diskette mit den CLI-Befehlen anfordern. Dies geschieht nun nicht mehr — zumindest, wenn sich die Befehle auf der RAM-Disk befinden und auch das *assign* nicht vergessen wurde.

12.2.2 Kombination von RAM-Disk und Startup-Sequence

Es ist nun natürlich naheliegend, die oben vorgestellte Befehlsfolge in die *startup-sequence* (s.o.) zu legen. Der Startvorgang dauert dann zwar etwas länger, aber dafür geht danach alles viel schneller.

Besonders für Programmierer ist die Kombination RAM-Disk und *startup-sequence* noch in einer weiteren Hinsicht interessant. Viele Compiler-Sprachen benötigen bei der Übersetzung eines Programms eine ganze Reihe von Hilfs-Dateien (bei der Programmiersprache *C* z.B. die *.h*-Dateien). Wenn man diese bereits beim Start in die RAM-Disk legt, so führt das zu einem wesentlich beschleunigten Übersetzungsvorgang — und den weiß jeder Programmierer zu schätzen. (Auch die temporären Dateien, die von einem Compiler oft während der Übersetzung angelegt werden, schickt man, sofern dies möglich ist, am besten auf die RAM-Disk.)

12.2.3 Probleme mit der RAM-Disk

Das Hauptproblem der RAM-Disk ist naheliegend: Sie verschlingt kostbaren Speicherplatz, den man vielleicht anderweitig benötigen würde. So werden viele Programme zwar schneller, wenn Sie Dateien auf der RAM-Disk lesen und schreiben. Dafür kann es aber sein, daß nicht mehr so viele Programme gleichzeitig in den Speicher passen. Im Extremfall kann man, wie auf anderen Computern, nur mit einem Programm gleichzeitig arbeiten. Will man nun von einem Programm zum anderen gelangen, so muß man erst das eine verlassen und das andere starten. Diese Vorgänge können sehr langwierig sein und einen großen Teil der durch die RAM-Disk gewonnenen Geschwindigkeit wieder wett machen.

Die Situation wird noch verschlechtert durch die Tatsache, daß viele Programme, während sie arbeiten weiteren Speicherplatz beim

Betriebssystem anfordern. Man merkt also nicht beim Start eines Programms, ob es noch in den Speicher paßt. Vielleicht erst nach Stunden der Arbeit mit einem Programm "stürzt es auf einmal ab", weil es den benötigten Speicherplatz nicht bekommt.

Wenn ein Programm Daten auf der RAM-Disk modifiziert oder dort neue Dateien erzeugt, kommt ein zusätzliches Problem der RAM-Disk zum Tragen. Alle Daten auf ihr sind sehr vergänglich! Passiert eine System-Fehler — vielleicht gar nicht einmal in dem Programm, in dem man gerade arbeitet, sondern in einem anderen, gleichzeitig laufenden — so können alle Daten auf der RAM-Disk verloren sein. Die Verwendung der RAM-Disk selbst kann so ein Absturz-Grund sein (s.o.).

Falls irgendwie möglich, sollten Sie deshalb nur Dateien auf die RAM-Disk legen, die "nicht so wichtig sind" oder nur gelesen werden und noch einmal auf einer Diskette vorhanden sind.

12.3 Maßschneidern einer Start-Diskette

Auf einer Standard-Workbench-Diskette, wie Sie dem Amiga normalerweise beiliegt, befinden sich eine ganze Menge Dateien, die — je nach den Programmen, die Sie in der Regel verwenden — vielleicht nie benötigt werden. Je nachdem, ob Sie z.B. normalerweise mit der Workbench oder dem CLI arbeiten, welcher Drucker an welcher Schnittstelle angeschlossen ist und mit welchen Zeichensätzen Sie arbeiten wollen, können Sie eine ganze Menge dieser Dateien löschen und so kostbaren Platz auf der Diskette gewinnen. Dies ist besonders dann wichtig, wenn Sie nur ein Laufwerk besitzen.

Alle Maßnahmen zum Maßschneidern einer Diskette auf diese Art dürfen natürlich nur auf einer **Kopie** der Original-Diskette durchgeführt werden. Dies ist aber bestimmt inzwischen schon selbstverständlich für Sie, oder?

12.3.1 Drucker

Sie besitzen ja normalerweise höchstens einen Drucker (vielleicht sogar gar keinen), während im Druckerauswahl-Fenster der *Preferences* eine Vielzahl möglicher Drucker angeboten wird. Jede dieser Möglichkeiten kostet Platz auf

der Diskette. In dieser Auswahl werden Ihnen nämlich die sogenannten *Drucker-Treiber* gezeigt, die sich auf der Startdiskette im Dateiverzeichnis *devs/printers* befinden. Löschen Sie alle Drucker-Treiber in diesem Dateiverzeichnis, von denen Sie genau wissen, daß Sie sie nicht brauchen, und Sie gewinnen eine Menge Platz auf der Diskette.

Falls Sie weiterhin an der seriellen oder der parallelen Schnittstelle überhaupt keine Geräte angeschlossen haben und in naher Zukunft auch sicherlich nicht anschließen werden, können Sie zusätzlich die Datei *devs/serial.device* und/oder die Datei *devs/parallel.device* entfernen und so weiteren Platz gewinnen.

12.3.2 Zeichensätze

Wenn Sie nicht regelmäßig Textverarbeitung betreiben oder aus anderen Gründen die verschiedenen Zeichensätze des Amiga nicht oder nur selten benötigen, so können Sie auch diese löschen. Die Zeichensätze liegen alle im Dateiverzeichnis *fonts* und besitzen in diesem wiederum ein eigenes Dateiverzeichnis, in dem die verschiedenen möglichen Größen des entsprechenden Zeichensatzes eingetragen sind. Wenn Sie alle Dateien in *fonts* löschen, gewinnen Sie dadurch sehr viel Platz auf Ihrer Start-Diskette.

Natürlich brauchen Sie nicht sofort alle Zeichensätze zu löschen. Wenn Sie nur auf einige, aber nicht auf alle, verzichten wollen, können Sie diese auch selektiv entfernen. Löschen Sie dazu zunächst die Datei mit dem Zeichensatz-Namen und dem Zusatz *font* aus dem Dateiverzeichnis *fonts*. Danach müssen alle Dateien im Dateiverzeichnis mit dem Namen des Zeichensatzes gelöscht werden und auch dieses Dateiverzeichnis selbst. Die folgenden zwei Befehle löschen z.B. den Zeichensatz *opal*.:

```
delete :fonts/opal.font
delete :fonts/opal all
```

Egal, wieviele Zeichensätze übrigens gelöscht werden, einer steht Ihnen immer zur Verfügung. Der Zeichensatz *Topaz* befindet sich nämlich bereits auf der Kickstart-Diskette und kann deshalb nicht aus dem System entfernt werden.

12.3.3 Kommandos

Viele der Kommandos im Dateiverzeichnis *c* werden Sie vielleicht nie anwenden oder nur sehr selten benötigen. Wenn Sie mit einer Start-Diskette nur auf der Workbench arbeiten und eine andere für das CLI angelegt haben, so können Sie auf der Workbench-Diskette fast alle Kommandos löschen.

Zu diesen selten benötigten Kommandos gehören z.B. der *sort*-Befehl der Editor *edit* (nicht *ed*), der meistens nicht sehr hilfreiche *why*-Befehl usw.. Am besten stellen Sie selbst anhand der Beschreibungen der CLI-Befehle fest, welche Sie behalten wollen und welche nicht. Wenn Sie eine Diskette speziell für die Workbench maßschneidern, löschen Sie am besten alle Kommandos außer *cd*, *dir*, *loadwb*, *echo* und *run*. Diese fünf Befehle sollten Sie aber auf alle Fälle auf jeder Start-Diskette belassen (*run* wird z.B. von manchen Programmen intern benötigt).

12.3.4 Reine CLI-Disketten

Verwenden Sie eine Diskette nur für die Arbeit im CLI, so können Sie alle Dateien löschen, die nur für die Workbench von Interesse sind.

Dazu gehören z.B. fast alle Dateien, die auf *.info* enden. Diese *Info-Dateien* enthalten nämlich Informationen, die der Workbench helfen, diese Datei als Icon zu zeigen (s.u.). In allen Dateiverzeichnissen, die als Schubladen auf der Workbench sichtbar sind, können die *.info*-Dateien gelöscht werden. Zusätzlich sollten Sie vielleicht auch *Preferences* löschen und Datum und Uhrzeit mit *date* stellen. Mit den Info-Dateien, die zu Projekten gehören, sollten Sie vielleicht etwas vorsichtiger sein, da manche Werkzeuge beim Öffnen von Projekten nach diesen suchen. Info-Dateien von Werkzeugen selbst können aber meist gefahrlos gelöscht werden.

Die Dateien in den Schubladen *System*, *Demos* und *Utilities* auf der Original-Startdiskette dürften für die meisten Anwender des CLIs ebenfalls kaum von Interesse sein. Die entsprechenden Dateiverzeichnisse können komplett gelöscht werden.

12.4 Dateien und Icons

Zum Abschluß dieser kleinen Sammlung von Tips und Tricks wollen wir uns nun noch etwas um die Verbindung zwischen CLI und Workbench kümmern. Bis jetzt waren die Tips ja zu einem großen Teil für Anwender gedacht gewesen, die vermehrt im CLI tätig werden wollen. Nun wollen wir einmal sehen, wie wir eine Brücke vom CLI zur Workbench schlagen und z.B. "unsichtbare" Dateien sichtbar machen können.

12.4.1 Info-Dateien

In den vorangegangenen Kapiteln sind ja schon mehrfach die Info-Dateien angesprochen worden. Diese tragen den selben Namen wie jeweils eine andere Datei — nur mit der zusätzlichen Endung *.info*.

Diese Dateien sind hauptsächlich für die Workbench gedacht. Sie enthalten z.B. das Icon, das auf der Workbench für die entsprechende Datei gezeigt wird, die Position, an der dieses Icon innerhalb des Schubladen-Fensters dargestellt wird und alle anderen Informationen, die mit dem *Info*-Befehl aus dem *Project*-Menü betrachtet und geändert werden können.

Für Programmierer stehen Bibliotheken zur Verfügung, mit deren Hilfe diese Informationen zu einer beliebigen Datei ermittelt werden können, ohne dazu das Format der Info-Datei kennen zu müssen. Nur wenige Programme benötigen im Moment diese Informationen überhaupt — hauptsächlich dienen sie zu Ihrer Information.

12.4.2 Icons

Wenn Sie aber gar keine komplizierten Operationen mit einer Info-Datei vorhaben, sondern einfach nur eine Datei (oder ein Dateiverzeichnis) sichtbar machen wollen, die Sie zwar mit dem *dir*-Befehl im CLI sehen, nicht aber auf der Workbench, so geht dies relativ leicht.

Alles, was man dazu tun muß, ist, dieser Datei eine entsprechende Info-Datei zu verschaffen. Der einfachste Weg, eine solche Datei zu erzeugen, ist das Kopieren einer anderen Info-Datei. Die folgenden zwei Befehle machen z.B. eine Datei namens *emacs*, die sich im Dateiverzeichnis *c* befand, in der Schublade *Utilities* sichtbar.


```
copy :c/emacs :utilities/emacs  
copy :utilities/Notepad.info :utilities/emacs.info
```

Emacs ist nun in der Schublade **Utilities** sichtbar, sieht genauso aus wie **Notepad** (nur eben mit einem anderen Namen unter dem Icon) und liegt zunächst auch an der selben Position — also über dem Icon von **Notepad**. Diese Datei kann nun auf der Workbench kopiert, gelöscht oder geöffnet werden, wie Notepad auch. (Es handelt sich um ein Programm, das auf der Standard-Workbench-Diskette **nicht** mitgeliefert wird.)

Möchte man eine unsichtbare Schublade (also ein Dateiverzeichnis) sichtbar machen, so muß in ihr eine Datei namens ".info" liegen. Hierfür kopiert man zweckmäßigerweise ebenfalls eine solche Datei aus einer anderen Schublade. Der folgende Befehl macht z.B. das Dateiverzeichnis **c**, das die CLI-Kommandos enthält, als Schublade unter dem selben Namen sichtbar.

```
copy :System/.info :c/.info
```

Die einzelnen Kommandos werden dadurch aber noch nicht sichtbar. Es wäre nun aber z.B. möglich, sie auf einen Schlag zu löschen, indem man die Schublade **c** in den Mülleimer wirft — wovor man sich aber hüten sollte!

Selbst wenn eine Schublade eine Info-Datei besitzt, kann es trotzdem sein, daß sie nicht sichtbar wird. Dies passiert dann, wenn diese Schublade ihrerseits in einer anderen Schublade liegt und diese **nicht** sichtbar ist. Diese unsichtbare Schublade kann also nicht geöffnet werden. Somit kann die "eigentlich" sichtbare Schublade (mit der Info-Datei) darin auch nicht gesehen werden.

12.4.3 IconEd

Mit der oben vorgestellten Methode können unsichtbare Dateien und Schubladen wenigstens schon einmal sichtbar gemacht werden. Die Auswahl für die Icons, mit denen Sie auf der Workbench auftauchen, ist jedoch beschränkt — auf genau die Icons, die schon in irgendeiner Schublade zu finden sind. Aber auch für Anwender, die in dieser Hinsicht Ihren Amiga etwas persönlicher gestalten wollen, gibt es eine Lösung.

Das Programm **IconEd** in der Schublade System erlaubt es, das Icon einer Datei nachträglich zu ändern. Die Bedienung dieses Programms ist recht einfach, wenn Sie bereits mit anderen Amiga-Programmen vertraut sind, und wird deshalb nicht allzu ausführlich erläutert.

12.4.3.1 Starten und Verlassen von IconEd

Nach dem Start von IconEd werden Ihnen in einem Requester zunächst ein paar Informationen gezeigt, die Sie durch Betätigung der Taste OK verschwinden lassen können. Sie sehen 9 kleine Rechtecke, in denen sich 9 Icons befinden und ein großes Rechteck, in dem eines dieser Icons vergrößert dargestellt wird.

Das Fenster von IconEd kann nicht vergrößert oder verkleinert und damit auch nicht verschoben werden, da es so groß ist wie ein ganzer Bildschirm. Man kann es aber nach hinten legen und so wieder in die Workbench zurückgelangen.

Wollen Sie das Programm wieder verlassen, so brauchen Sie dazu nur das Fenster zu schließen, indem Sie das Schließ-Gadget betätigen.

12.4.3.2 Laden und Abspeichern von Icons

Wenn Sie das Icon einer Datei ändern wollen, so müssen Sie es zunächst einlesen. Dazu dient der Befehl **Load Data** im Menü **Disk**. Es erscheint ein Requester, in dem Sie den Namen der zu ändernden Datei in einem Text-Gadget eingeben und dann die Taste **Load Icon Image** betätigen müssen. Geben Sie am besten den vollen Pfadnamen der zu ändernden Datei ein. Sie brauchen sich dann nicht darum zu kümmern, was IconEd für das aktuelle Dateiverzeichnis hält.

Der Load-Befehl funktioniert nur dann, wenn die Datei, deren Icon eingelesen werden soll, bereits ein Icon (also eine Info-Datei) besitzt. Wie wir eine solche Datei erzeugen können, haben wir weiter oben ja schon gesehen.

Auf genau dieselbe Art und Weise können Sie ein Icon wieder in einer Info-Datei abspeichern. Klicken Sie dazu auf das kleine Bildchen des Icons in einem der 9 Rechtecke und wählen Sie dann **Save Data** im Menü **Disk** aus. Der auftauchende Requester wird entsprechende bedient, wie der bei **Load Data**.

12.4.3.3 Editieren von Icons

Das Ändern der Icons selbst läuft ähnlich ab wie das Editieren des Mauszeigers in Preferences. Sie können allerdings jederzeit eines von insgesamt 9 verschiedenen Icons bearbeiten. Immer wenn Sie eines der kleinen Icons rechts am Bildschirm anklicken, so wird dieses in vergrößerter Form links gezeigt und kann geändert werden.

In diesem können Sie nun einen Punkt anklicken und er wird mit der *aktuellen Farbe* versehen. Die *aktuelle Farbe* wählen Sie dabei zuvor aus dem Menü **Color** aus.

Alle anderen Menü-Befehle dürften durch Ausprobieren am besten zu begreifen sein. Wenn Sie Ihre Disketten also etwas persönlicher gestalten wollen: Fröhliches Probieren — mit IconEd können Sie kaum etwas falsch machen oder zerstören!

13 Chips

Im letzten Teil dieses Buches, der mit diesem Kapitel beginnt, soll ein wenig in die Innenwelt des Amiga eingestiegen werden. So werden Sie z.B. einiges über die Hardware des Amiga erfahren, die entscheidend auch zu den phantastischen Fähigkeiten der Software beiträgt, über die speziellen Chips, die ihn aus der Masse ähnlicher Computer hervorheben und zuletzt über die Schnittstellen, über die der Amiga mit der Außenwelt kommunizieren kann.

Sie benötigen diese Informationen nicht unbedingt zur Arbeit mit dem Amiga. Sie sind aber vielleicht trotzdem ganz interessant (und können Ihnen natürlich auch beim Fachsimpeln mit anderen Computer-Besitzern behilflich sein — sofern Sie daran Interesse haben).

In diesem ersten Kapitel des letzten Buchteils soll es zunächst einmal um die speziellen Chips gehen, die — mehr als alle anderen Komponenten des Systems — den Amiga erst zum Amiga machen.

Um den Amiga zu einer Supermaschine zu machen, hat Commodore/Amiga mehrere Spezialchips zur Unterstützung des eigentlichen Mikroprozessors in den Amiga eingebaut. Diese Chips geben dem Amiga seine Fähigkeiten nicht nur in Bezug auf grafische und akustische Möglichkeiten, sondern ermöglichen beispielsweise erst die Multitaskingfähigkeit des AmigaDOS.

Der Amiga hat drei verschiedene Spezialchips eingebaut. Sie heißen Paula, Denise und Agnus — sinnvolle Namen, die nicht an das Zahlengewirr erinnern, das man bei der Identifizierung anderer Chips benötigt. Welcher dieser Chips welche Aufgaben erfüllt, erfahren Sie auf den nachfolgenden Seiten.

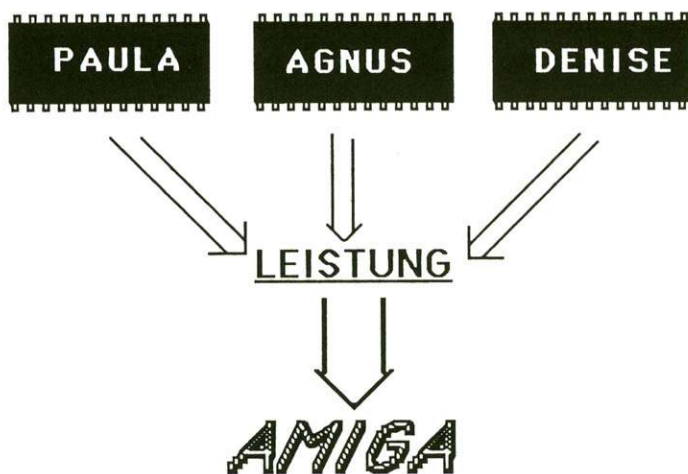


Bild 13 - 1: Die Philosophie der Amiga-Hardware

13.1 Die CPU

Der wichtigste Chip im Innern des Amiga ist zweifellos die sogenannte CPU (*Central Processing Unit* oder auf deutsch *Zentraleinheit*), wenn auch die anderen Chips natürlich spektakulärer sind. Die CPU ist derjenige Chip in einem Rechner, der den Löwenanteil der Arbeit zu leisten hat. Es ist dieser Chip, der die Programme wirklich ausführt, also rechnet und Entscheidungen trifft. Man könnte die CPU als das Herz eines jeden Computers bezeichnen.

Im Amiga wird diese Aufgabe dem 16/32-Bit-Prozessor M68 000 der Firma Motorola übertragen. Er wird auch in anderen Rechnern in dieser Preisklasse und in professionellen Systemen für 50 000 DM und mehr eingesetzt. Die Bezeichnung 16/32-Bit-Prozessor kommt daher, daß man sich nicht darüber einig werden kann, nach welchen Kriterien man einen Prozessor in welche Kategorie einordnet.

Typischerweise ist es so, daß Prozessoren, die jeweils 8 Bit auf einen Schlag verarbeiten können (z.B. zwei 8 Bit-Zahlen addieren können) 8-Bit-Prozessoren heißen, Prozessoren, die dies in einem Schritt mit 16 oder 32 Bit schaffen, dann entsprechend 16- oder 32-Bit-Prozessoren. Ein anderes Kriterium ist aber auch, wieviele Bit in einem Schritt aus dem Speicher gelesen oder in ihn geschrieben werden können. Der M68 000 erfüllt bei einigen dieser Kriterien die Bedingungen für einen 32-Bit-Prozessor und bei anderen nur die für einen 16-Bit-Prozessor.

Kommen wir nun aber zu den drei Spezial-Chips. Bei Ihnen handelt es sich um sogenannte Custom-Chips, d.h. sie wurden speziell für einen Computer-Hersteller (in diesem Falle Amiga) entwickelt und werden auch nur für ihn hergestellt. Die meisten anderen Computer-Chips sind Standard-Chips, die jeder Kunde dem Chip-Hersteller abkaufen und dann in seinen eigenen Produkten verwenden kann. Standard-Chips sind fast immer billiger als Custom-Chips, weil sie üblicherweise in größeren Stückzahlen verkauft werden. Sie stellen aber immer einen Kompromiß zwischen den Wünschen vieler Anwender dar, weswegen man für wirklich einzigartige Anwendungs-zwecke oder besonders hohe Leistungen Custom-Chips benötigt.

13.2 Funktionsweise der Chips

Die drei Zusatz-Chips Agnus, Paula und Denise arbeiten unabhängig und parallel zu der CPU. Letzteres ist eine Behauptung, die Sie vielleicht schon oft gehört haben. Was sie aber bedeutet, ist nicht ganz so leicht zu erklären und deshalb muß hier etwas weiter ausgeholt werden.

13.2.1 Koprozessoren

Es gibt viele Computer, die fast alle "Arbeit" der CPU überlassen — also die Bearbeitung von Grafikdarstellungen, Musik, Berechnungen und auch die Ein- und Ausgabe. Da all diese Arbeiten ein Prozessor übernimmt, müssen diese Tätigkeiten hintereinander ausgeführt werden. Jede Tätigkeit für eine dieser Aufgaben hält ihn eine Weile von den anderen ab.

Die Spezialchips des Amiga nehmen dem Hauptprozessor Arbeit ab. Die Grafik-, Sound- und noch andere Fähigkeiten des Amiga werden durch die drei Chips erledigt. Der Prozessor hat nicht viel damit zu tun. Um diese Verlagerung der Aufgaben aber noch nützlicher zu machen, können die Chips ihre Aufgaben parallel zu denen der CPU erledigen. Es handelt sich sogenannte *Koprozessoren*, die unabhängig von der CPU ihrer Arbeit nachgehen. Die CPU bleibt immer noch "Herr im Hause Amiga", gibt aber — wie eine gute Managerin — niedere Aufgaben an ihre Mitarbeiter weiter und wendet sich dann anderen Dingen zu.

Man kann sich das so vorstellen, daß die CPU eine Aufgabe vorbereitet und den Koprozessor mit allen nötigen Informationen darüber versorgt. Dann

"stößt sie ihn an" und kümmert sich um andere Dinge. Währenddessen wird der Koprozessor tätig und erledigt die ihm aufgetragene Arbeit. Wenn er damit fertig ist, signalisiert er dies der CPU (mit einem sogenannten *Interrupt*) und kann wieder neu mit Arbeit versorgt werden. Alternativ kann die CPU auch "zwischendurch" immer nachschauen, ob der Koprozessor schon mit seiner Arbeit fertig ist und darauf entsprechend reagieren.

Bei einer solchen Konfiguration mit CPU und Koprozessoren tritt oft nur ein Problem auf. Sowohl die CPU wie auch die Koprozessoren besitzen üblicherweise nur sehr wenig Speicherplatz, die sogenannten *Register*. Für größere Aufgaben benötigen sie deshalb den RAM-Speicher, der ungleich mehr Daten fassen kann. In vielen Computern gibt es Koprozessoren und andere Hilfschips, die diese Daten nicht selbst aus dem Speicher holen und neu dort hinterlegen können, sondern dazu immer die CPU benötigen, weil diese als einzige an den Speicher angeschlossen ist. Immer, wenn sie Daten brauchen oder ablegen wollen, müssen Sie die CPU von ihren eigentlichen Aufgaben abhalten und sie als Datenschaufel mißbrauchen. Ein wirklich paralleles Arbeiten, das solche Zusatz-Chips erst richtig ausnutzt, ist also nur solange möglich, wie der Speicher nicht gebraucht wird.

Auf dem Amiga ist das anders! Hier sind nämlich auch die Koprozessoren selbst an den Speicher angeschlossen.

13.2.2 DMA

Auf den Speicher des Amiga kann man über insgesamt 25 Wege oder *Kanäle* zugreifen. Einige dieser Kanäle nutzen auch die Spezial-Chips, andere sind für künftige Erweiterungen noch frei. Diese Technik nennt man *direkten Speicherzugriff*, im Fachjargon auch *DMA* (*Direct Memory Access*). Von Chips, die von selbst auf den Speicher zugreifen können, spricht man deshalb auch als *DMA-Chips*.

DMA ist ein recht fortschrittliches und teures Konzept, das erst vor kurzem Eingang in preiswerte Computer gefunden hat. Es ist für vielerlei Anwendungen nützlich. Neben Aufgaben wie Grafik und Klangerzeugung — die DMA benötigen, weil bei diesen Vorgängen Daten aus dem Speicher teilweise sehr schnell gelesen und in Bilder und Töne umgewandelt werden müssen — gibt es noch eine Reihe anderer Anwendungen für DMA.

Die DMA-Chips des Amiga nehmen dem Hauptprozessor aber nicht nur die Arbeit ab, sondern können Ihre Arbeiten auch viel schneller erledigen, als der Hauptprozessor dies könnte. Er ist schließlich nicht speziell für Grafik oder

speziell für das Hin- und Herschieben von Daten im Speicher gebaut worden, wie es die Spezial-Chips sind.

13.2.3 Probleme und Lösungen

Obwohl die Spezial-Chips die CPU nicht mehr für den Speicherzugriff benötigen, tritt trotzdem ein Engpaß beim Speicher auf. Die Chips und der Prozessor können normalerweise nicht gleichzeitig auf den Speicher zugreifen. Dieser kann nämlich immer nur mit einem anderen Bauteil zur gleichen Zeit kommunizieren. Es gibt zwar besondere Speicher, die mit zwei oder mehr anderen Chips gleichzeitig Daten austauschen können, diese sind aber sehr viel teurer und für einen Personal Computer wie den Amiga somit kaum geeignet.

Glücklicherweise kann der Speicher jedoch mit viel höherer Geschwindigkeit Daten aufnehmen und senden, als die CPU dies kann. Diese Geschwindigkeit, mit der Chips arbeiten können, nennt man ihre *Taktfrequenz*. Die CPU durchläuft in jeder Sekunde ca. 7,2 Millionen solcher *Taktzyklen* und hat somit eine Taktfrequenz von 7,2 Mega-Hertz (MHz). In jedem Takt-Zyklus kann die CPU einen Arbeitsschritt durchführen, also z.B. ein Wort (16 Bit) Daten vom Speicher empfangen. Der RAM-Speicher läuft aber mit einer Taktfrequenz von 14,4 MHz, hat also in der Sekunde die doppelte Anzahl von Takt-Zyklen.

Diesen Geschwindigkeitsvorsprung des Speichers haben die Amiga-Ingenieure so ausgenutzt, daß sie die CPU jeden zweiten Takt mit der CPU kommunizieren lassen und alle anderen Takte auf die restlichen DMA-Chips verteilt werden. Da diese den Speicher meist nicht so heftig brauchen wie die CPU, können so eigentlich immer CPU und Spezial-Chips ungestört voneinander gleichzeitig ihrer Arbeit nachgehen.

Nur unter besonderen Umständen benötigen die Spezial-Chips mehr Takt-Zyklen des Speichers. In diesen Fällen wird die CPU kurz angehalten, wenn Sie den Speicher zur selben Zeit benötigt wie ein anderer Chip und darf dann wieder weitermachen, wenn dieser den Speicher nicht mehr braucht. (Kurz heißt in diesem Fall wirklich kurz. Es handelt sich dabei um Zeitspannen, in denen kein Mensch reagieren oder auch nur etwas von einer solchen Pause merken könnte.)

Diese Fälle, in denen die CPU gebremst wird, kommen aber nur sehr selten oder unter ganz besonderen Umständen vor. Wenn Sie z.B. 32 verschiedene Farben (oder 16 Farben bei höchster Auflösung) gleichzeitig am Bildschirm sehen wollen, bremst Denise (der Videochip) die CPU merklich ab, weil er

wesentlich mehr Daten aus dem Speicher holen muß. Wenn Sie sich aber mit weniger Farben begnügen, üben die CPU und die Spezial-Chips friedliche Koexistenz.

13.3 Paula

Einer der drei Spezialchips des Amiga trägt den Namen "Paula". Das ist nicht nur irgendein Name, sondern leitet sich auf eine heute nicht mehr nachvollziehbare Weise aus *Peripheral/Audio* ab, was schon andeutet, wofür dieser Chip zuständig ist. Er kontrolliert die Arbeit mit den Peripheriebausteinen und die Klangerzeugung. Und er kann noch mehr, denn er verwaltet die sogenannte *Interruptverarbeitung*.

13.3.1 Interrupts

Interrupts sind, wörtlich übersetzt, Unterbrechungen. Meist andere Chips unterbrechen die CPU dabei in ihrer Arbeit, weil irgendein Ereignis eingetreten ist, das die Aufmerksamkeit der CPU verlangt. Koprozessoren schicken z.B. Interrupts, wenn sie mit ihrer Arbeit fertig sind, und Ein-/Ausgabe-Bausteine schicken Interrupts, wenn Daten bereitstehen und die CPU entscheiden muß, was damit gemacht werden soll. Es gibt verschiedene Interrupts, mit denen der CPU verschiedene Nachrichten geschickt werden können. Zusätzlich stellen Chips, die Interrupts verursachen, meist noch in bestimmten *Registern* Informationen bereit, aus denen die CPU Informationen über die näheren Umstände eines Interrupts entnehmen kann.

Bei einem Interrupt wird der gesamte Zustand des gerade laufenden Programms gesichert und ein spezielles kleines Programm gestartet, das sich um die Behandlung des Interrupts kümmert, ein sogenannter *Interrupt-Handler*.

Interrupts besitzen zudem auch *Prioritäten*, die besagen, wie wichtig diese Unterbrechung ist. Es kann ja sein, daß gerade kein "normales Programm" läuft, sondern ein Interrupt-Handler. Falls dieser wichtiger ist als der neu eingetroffene Interrupt, darf er auf keinen Fall unterbrochen werden. Genau dies ermöglichen die Prioritäten. Es gilt die Regel: Interrupts niedrigerer Priorität können keine Interrupt-Handler von gleicher oder höherer Priorität unterbrechen.

Der Paula-Chip hilft der CPU dabei, die Interrupts der verschiedenen Systemkomponenten zu erkennen und richtig einzuordnen. Das AmigaDOS nutzt diese Interruptverarbeitung auch aus, um das Multitasking zu ermöglichen. Hierzu wird in regelmäßigen Abständen ein spezieller Interrupt erzeugt, der dazu führt, daß das aktuelle Programm unterbrochen wird. Nach dem Ende des entsprechenden Interrupt-Handlers wird aber das unterbrochene Programm nicht wieder aufgenommen, sondern ein anderes, das bislang brach lag.

Paula macht im Zusammenhang mit Interrrupts folgendes: Sie verbindet die Interrupts der Ein-/Ausgabe-Bausteine, der anderen DMA-Chips und des Floppy-Controllers (=der Chip, der die Arbeit mit dem Disketten-Laufwerk steuert) und gibt sie als entsprechende Signale an die Register, in denen der Prozessor nachsieht, was er bei einer "Unterbrechung" machen soll. Paula gliedert die vielen möglichen Interrupts dabei auch gleich in viel mehr unterschiedliche Gruppen auf als die CPU dies kann. Die CPU braucht deshalb viel weniger Zeit, um zu erkennen, um was für einen Interrupt es sich handelt und wie "angemessen" darauf zu reagieren ist.

13.3.2 Ein- und Ausgabe

Eine weitere Aufgabe des Paula-Chips ist der Kontakt zwischen Außenwelt und System. Paula betreut die Ein-/Ausgabe-Bausteine, mit deren Hilfe die Schnittstellen angesteuert werden. Also Drucker, zusätzliche Floppy oder Harddisk, die serielle Schnittstelle, den Maus-Anschluß, und so weiter. Daß Paula, also ein von der CPU getrennter Chip, diese Aufgabe übernimmt, bedeutet für den Endbenutzer also wieder, daß er etwas auf den Drucker ausgeben kann, während er etwas Neues in den Computer eintippt. Der Paula-Chip ist praktisch der Mittelsmann zwischen dem Computer-System an sich und den durch die Peripheriebausteine angesteuerten Schnittstellen.

13.3.3 Klangerzeugung

Die Klangerzeugung und -ausgabe auf dem Amiga ist (nahezu) ausschließlich eine Sache des Paula-Chips. Dabei kann man aus diesem Baustein recht realistische Effekte herbeizaubern.

Jedes Geräusch hat ja seine eigenen Klang- und Wellenformen. Die verschiedenen Wellenformen machen sich für den Menschen hörbar als verschiedene Instrumente oder Geräusche, das Gehirn macht aus den

Wellenformen, mit denen der Schall ins Ohr dringt, entsprechend erkennbare Klangformen.

Der Paula-Chip ermöglicht es nun, beliebige Wellenformen als digitale Werte zu speichern und sie bei der Ausgabe analog, also als Wellenform, über den Lautsprecher klingen zu lassen. Das bedeutet, man kann praktisch die Wellenform jedes beliebigen Instruments einspeichern und abspielen lassen. Um die Musikqualität zu erhöhen, ist ein Tiefpaßfilter eingebaut, der störende Rauschgeräusche entfernt. Der Paula-Chip hat vier Soundkanäle, von denen jeweils zwei auf einen der beiden Stereokanäle ausgegeben werden. Durch die DMA-Technik ist es außerdem möglich, Musik mit dem Amiga unabhängig von anderen laufenden Programmen spielen zu lassen. Paula holt sich dazu die für die Klangerzeugung nötigen Daten einfach selbst aus dem Speicher – wo sie die CPU allerdings hinterlegen muß – sobald neue benötigt werden.

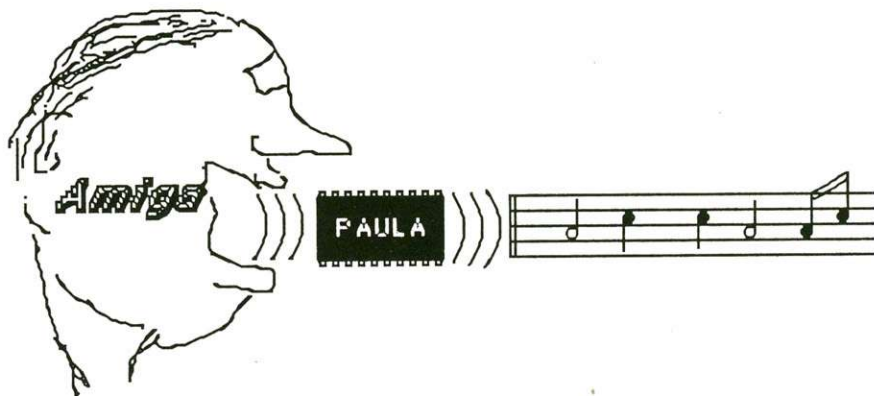


Bild 13 - 2: *Klangerzeugung auf dem Amiga*

Näheres zu Klangerzeugung mit Paula aber in einem eigenen Kapitel über dieses Thema.

13.4 Denise

Der Amiga wird vielfach auch als Grafikwunder bezeichnet. Die außergewöhnlichen Grafikfähigkeiten verdankt er zum Großteil Denise. *Denise* hat ähnlich mysteriöse Ursprünge, wie der Name *Paula* und stammt "irgendwie" von *Display Encoder* (Bild-Kodierer) ab. Denise erledigt, zusammen mit Agnus, fast alles, was mit Grafik auf dem Amiga zu tun hat.

13.4.1 Grafik und Farbe

Denise ist verantwortlich, für die Umsetzung einer Folge von Daten im RAM-Speicher in das Bild, das hinterher auf dem Monitor zu sehen ist.

Dieses Bild kann — je nach den Wünschen des Programms bzw. seines Anwenders — die verschiedensten Auflösungen besitzen (*H* und *V* in dieser Tabelle bedeuten natürlich *horizontal* und *vertikal*):

Grafikauflösung	maximal mögl. Farbenzahl
320 (H) * 200 (V)	32
640 (H) * 200 (V)	16
320 (H) * 400 (V)	32
640 (H) * 400 (V)	1

Tabelle 13 - 1: Mögliche Grafik-Auflösungen von Denise

Jede dieser oben genannten Farben kann aus einer Palette von 4096 verschiedenen Farbtönen ausgewählt werden. Der Denise-Chip hat 32 Farbregister (Register=Speicherzellen, aus denen sich der Chip spezielle Informationen holt), in denen jeweils einer dieser Farbtöne hinterlegt werden kann. In jedem dieser Register sind die Werte gespeichert, die die Rot-, Grün- und Blau-Intensität der entsprechenden Farbe angeben.

Außer den oben genannten Grafikauflösungen verfügt der Denise-Chip über einen sogenannten *Hold And Modify*-Modus (kurz: *HAM*). Damit (und einigen Kunstgriffen bei der Programmierung) ist es möglich, alle 4096 Farben auf einmal am Bildschirm zu zeigen.

Mit diesen Fähigkeiten hat man nun schon das reinste Grafikwunder vor Augen; doch der Amiga kann noch mehr.

13.4.2 Bewegung ins Bild!

Denise ermöglicht gleich zweierlei Arten der Bewegung. Zum Einen kann der Hintergrund einer Grafik (das sogenannte *Playfield*) in Wirklichkeit eine sehr große Fläche, größer als der Bildschirm, sein, aus dem nur ein Teil gezeigt wird. Dieser gezeigte Ausschnitt kann stufenlos und schnell über dem gesamten Playfield verschoben werden. Die zweite Möglichkeit, Bewegung ins Bild zu bringen, sind die sogenannten *Sprites*, kleine farbige Bilder, die

sich "über" dem Playfield befinden und sehr schnell unabhängig von diesem bewegt werden können. *Sprites* erfreuen sich auch auf anderen Computern schon großer Beliebtheit und lassen sich (gegenteilig lautenden Meinungen zum Trotz) auch außerhalb von Video-Spielen sinnvoll einsetzen.

Playfields sind hochauflösende Grafiken, die entsprechende Grafik - auflösungen und Farben, wie die in Tabelle 13-1 beschriebenen, haben können. Das Besondere an diesen Playfields ist eigentlich nur die Größe. Der Denise-Chip hat nun Register, die angeben, welcher Teil einer solchen Riesen-Grafik auf dem Bildschirm dargestellt wird; durch ständige Änderung dieser Register kann quasi mit dem Bildschirm über die Grafik gefahren werden, also genau so, als würde man eine Kamera herumbewegen und einen bestimmten Ausschnitt der Landschaft auf den Bildschirm bringen. Diese Art von Bewegung einer Hintergrundgrafik nennt man auch *Scrolling*, am Amiga speziell *Playfield-Scrolling*.

Diese Fähigkeit des Denise-Chips erlaubt so beispielsweise Videospiele, wie man sie bisher nur von den Spielhallen kennt, aber noch nicht auf Computern gesehen hat. Für den Endanwender des Amiga könnte diese Fähigkeit aber auch im Zusammenhang mit *Computer Aided Design* (CAD) interessant werden. Ein Standard im CAD-Bereich ist z.B. eine Bildschirm-Auflösung von ca. 1024 mal 1024 Punkten. Der Amiga bietet dies zwar nicht in voller Größe an, aber mit Hilfe des Playfield-Scrollings kann man sich diese Grafikauflösung "künstlich" schaffen.

Zusätzlich bietet der Denise-Chip eine weitere Animationsfähigkeit: *Sprites*. Sprites sind kleine Objekte (Bilder), deren Anzeige unabhängig vom Hintergrund erfolgt. Sprites erscheinen üblicherweise noch "über" der Hintergrundgrafik (dem Playfield). Ein Amiga-Sprite kann 16 Punkte breit und eine beliebig hohe Zahl von Punkten hoch werden. Der Amiga hat 8 solcher Sprites. Jedes dieser 8 Sprites kann — unter bestimmten Voraussetzungen und Einschränkungen — mehrmals verwendet werden, so daß eine weitaus größere Zahl von unabhängig bewegbaren Objekten möglich ist.

Sprites sind im Zusammenhang mit Videospielen besonders interessant, sie können dem Anwender aber auch eine freundlichere Bedienung bringen. Auch der Mauszeiger ist z.B. ein Sprite.

Der Denise-Chip erkennt Kollisionen verschiedener Sprites mit Teilen des Playfields und mit anderen Sprites. Programme können diese Kollisions - abfrage benutzen und eine entsprechende Reaktion darauf einleiten. Hierzu ein (etwas triviales) Beispiel: Man schießt in einem Spiel ein Ufo ab und trifft es. Die Kollision zwischen Schuß und Ufo wird festgestellt, als entsprechende Reaktion verschwindet das Ufo. Sie sehen schon, der Denise-Chip macht aus dem Amiga eigentlich eine richtige Spielhöllenmaschine. Sie

müssen den Computer aber deswegen nicht als solche benutzen, denn schließlich ist er ja ein Allround-Talent.

13.5 Agnus

"Agnus" ist eine etwas kryptische Abkürzung für *Adreß-Generator*. Agnus beinhaltet noch mehr als einen "Adreß Generator". Weitere wichtige Bestandteile dieses Chips sind der *Blitter* und der *Copper*. Näheres dazu weiter unten.

13.5.1 Was ist ein Adreß-Generator?

Der Adress Generator ist ein Hauptbestandteil des Agnus-Chips. Er beinhaltet alle DMA-Kanäle (s.o.) und steuert ihren Zugriff auf den Speicher. Die Custom-Chips des Amiga greifen — wie oben gesehen — unabhängig von der CPU auf den RAM-Speicher zu. Das geschieht über sogenannte *DMA-Kanäle*. Keiner der beiden anderen Chips kann jedoch selbst auf den Speicher zugreifen, alle DMA-Kanäle laufen über Agnus und werden so von Agnus kontrolliert und koordiniert.

Insgesamt hat der Amiga 25 dieser DMA-Kanäle. Der Adreß-Generator regelt nun den Zugriff der über die verschiedenen DMA-Kanäle laufenden Aufgaben (Musik, Grafik, Disk, etc.) auf den Datenkanal, über den der direkte Speicherzugriff läuft. Der Adress-Generator wartet immer, bis der Prozessor den Speicher sowieso nicht benötigt und greift dann darauf zu. Solange ein DMA-Zugriff aktiv ist, gibt Agnus nun ein Signal an den Prozessor, sich vom Speicher fernzuhalten (sofern er ihn benötigt, bevor der andere DMA-Chip damit fertig ist). Wie also bereits im Kapitel über die grundlegende Funktionsweise der DMA-Chips erläutert, kommt so kein Chaos ins System.

13.5.2 Der Blitter

Der Amiga wurde von vornherein mit der Hauptaufgabe konzipiert, eine schnelle Verarbeitung von Grafik zu ermöglichen. Um diese schnelle Verarbeitung überhaupt ermöglichen zu können, mußte eine Einheit geschaffen werden, die schnelle Manipulationen von Daten im RAM-Speicher

möglich machte, die der Video-Chip ja in das Monitor-Bild umwandelt. Diese Einheit ist der Blitter.

Der Name "Blitter" beruht auf einer Prozedur zum Verschieben und Kombinieren von Speicherblöcken für die Benutzung dieser Blöcke als Grafik. Diese Prozedur war als Befehl "BitBlit" in einem älteren Computer der Firma XEROX installiert. Und da die für den Amiga entwickelte Einheit nichts anderes tat, als eben diese Prozedur, lag wohl nichts näher, als einen ähnlichen Namen für diese Einheit zu finden: den Blitter.¹

Der Blitter ist eine recht eigenständige Komponente innerhalb von Agnus. Er ist ein echter Koprozessor (s.o.). Der Blitter beschert dem Amiga einige Leckerbissen in Bezug auf die Grafik. Der erste davon ist das Füllen von Flächen (wie es z.B. Graphicraft und Deluxe Paint demonstrieren). Flächen können mit einer Million Bildpunkte pro Sekunde ausgefüllt werden. Mit der selben Geschwindigkeit ist er auch in der Lage, gerade Linien zu ziehen.

Die Hauptanwendung des Blitters ist aber in der Animation (Bewegung) von Objekten zu sehen. Der Blitter kann nämlich sehr schnell (so schnell, wie es die Speichertechnologie überhaupt zuläßt) Daten im Speicher von einem Ort zu einem anderen bewegen. Dies kann die CPU auch — aber bei weitem nicht so schnell wie der Blitter. Während die CPU solche Speicherbewegungen zudem nur mit ganzen Gruppen von Bytes (8 Bits) und Worten (16 Bits) einfach (und damit schnell) durchführen kann, können die vom Blitter bewegten Daten an beliebigen Bit-Positionen im Speicher liegen, ohne daß dies Geschwindigkeitseinbußen mit sich bringen würde.²

¹ Jay Miner, der Designer der drei Amiga-Chips, ist in dieser Hinsicht anderer Meinung. Der Blitter des Amiga kann auch wirklich einiges mehr, als nur Daten hin und herschieben. Jay Miner nennt ihn deshalb Bimmer für *Bit Image Manipulator* (zu deutsch etwa *Bit-Bild-Manipulierer*). Dies soll andeuten, das dieser Prozessor nicht nur Daten durch den Speicher schaufeln, sondern zugleich komplexe Operationen damit durchführen kann.

² Weil der Blitter solche Operationen soviel schneller als die CPU kann, hat er in bestimmten Situationen auch "das Recht", diese völlig lahm zu legen, bzw. vom Speicher abzuschneiden. Für eine gewisse Zeit monopolisiert der Blitter dann den Speicher (und schneidet dabei auch die meisten anderen Chips ab) und gibt ihn erst dann wieder frei, wenn seine Operationen beendet sind. Dies bedeutet aber in Wirklichkeit keine Geschwindigkeitseinbußen für ein Programm (da die CPU die entsprechenden Operationen einfach nicht so schnell schaffen würde).

Statt Daten einfach nur zu bewegen, kann der Blitter auch bis zu drei Daten-*Quellen* nehmen, beliebige logische Operationen damit ausführen und die so kombinierten Daten wieder in den Speicher schreiben. Dies wird vor allem für die Fensterverwaltung wichtig (über die Sie noch mehr in einem folgenden Kapitel erfahren).

Diese schnelle Datenverschiebung wird auch für eine zusätzliche Gruppe von bewegten Objekten genutzt. Solche Objekte können wesentlich größer sein als Sprites (sie können fast beliebig groß werden) und man nennt sie im Amiga-Fachjargon BOB (kurz für *Blitter Object*).

Der Blitter ist also die Einheit, die dem Benutzer die außerordentlich schnelle Grafikdarstellung und zusätzliche Animationsfähigkeiten beschert.

13.5.3 Der Copper

Copper ist ein kurzes Wort für Co-Processor. Auch der Blitter ist ein solcher Koprozessor — für den Copper hatte man aber wohl offensichtlich keinen deskriptiveren Namen zur Hand. Der Copper ist allerdings ein ganz besonderer Koprozessor. Er ist nämlich eng an den Elektronen-Strahl gekoppelt, mit dem der Monitor das Bild zeichnet.³

Der Copper hat die Fähigkeit, zu erkennen, wo (in welcher Zeile und Spalte des Bildschirms) dieser Elektronenstrahl gerade ist. Er kann z.B. auf bestimmte Positionen warten (er hat dazu einen sogenannten *Wait*-Befehl) und kann dann bei Erreichen dieser Position die Register anderer Chips verändern. Durch diese Fähigkeit ist es zum Beispiel möglich, in der Mitte des Bildschirms die Farbreister zu ändern, so daß man praktisch auch ohne den Hold&Modify-Modus alle 4096 Farben gleichzeitig darstellen könnte.

Es bedeutet aber für den Benutzer auch, daß er mit der Maus einfach einen Bildschirm nach unten ziehen kann und dahinter etwas ganz anderes erscheint, ein anderer virtueller Bildschirm.⁴ Daß man durch Herunterziehen der einzelnen Bildschirme jeweils ein Drittel eines jeden Programms sehen kann, macht der Copper möglich. Als geschickter Programmierer kann man aber noch eine Menge anderer Tricks mit dem Copper anstellen, die Sie

³ Mehr zum Prinzip der Bilderzeugung in Computern übrigens im nächsten Kapitel.

⁴ Mehr zu den Vorteilen und zum Verhalten von *Screens*, virtuellen Bildschirmen, ist im Kapitel *Intuition* zu finden.

zweifellos auch bald in Grafik-Anwendungen und guten Video-Spielen sehen werden.

13.6 Begriffs-Ver- und Entwirrungen

Vielleicht ging es Ihnen in diesem Kapitel wie mir, als ich das erste Mal etwas über die Spezial-Chips des Amiga las und Sie leiden nun unter extremer Begriffsverwirrung. Ich hoffe, diese aber zum Abschluß dieses Kapitels noch etwas "entwirren" zu können. Vor allem möchte ich zwei Begriffe klären, die in den folgenden Kapiteln immer wieder verwendet werden, Chip und Prozessor.

Vielen dürfte im Moment nicht ganz klar sein, was denn nun ein Chip ist und wo die Koprozessoren des Amiga liegen. In der Berichterstattung über den Amiga in der Computer-Presse ging es meist auch wild durcheinander. Manchmal wurden der Blitter und der Copper als Chips bezeichnet und Denise und Agnus als Koprozessoren und manchmal war es umgekehrt. In Wirklichkeit ist aber (hoffentlich) alles ganz einfach.

Paula, Denise und Agnus sind drei Chips im Innern des Amiga. Sie arbeiten aber so eng zusammen, daß Sie eigentlich ein großes elektronisches Bauteil bilden und man sie deshalb am besten auf einem Chip zusammenfassen würde. Im Moment ist die VLSI-Technologie⁵ aber noch nicht so weit, daß man einigermaßen preiswert einen so großen Chip bauen könnte. Aus rein praktischen Erwägungen sind die für den Amiga benötigten Funktionen deshalb auf drei getrennte Bausteine verteilt worden.

Innerhalb der großen Funktionseinheit des *Superchips*, den Denise, Paula und Agnus zusammen bilden, existieren einzelne Untereinheiten, die sich um ganz spezielle Funktionen kümmern. Die Ansteuerung von externen Geräten, die Erzeugung eines Videobildes und die Manipulation einzelner Bits im Speicher sind s.B. solche Funktionen, die von jeweils einer Untereinheit des Superchips erledigt werden. Diese Untereinheiten nennen wir *Prozessoren*. Ein Prozessor ist eine *logische* Funktionseinheit, während ein Chip eine

⁵ VLSI (für *Very Large Scale Integration* oder auf deutsch *Integration im ganz großen Maßstab*) bezeichnet eine Technologie, bei der sehr viele der elektronische Bauteile (meist sogenannte *Transistoren*), aus denen ein Computer aufgebaut ist, auf einem Baustein oder *Chip* zusammengefaßt werden.

physikalische, greifbare Einheit ist (eben ein kleiner, schwarzer Klotz mit vielen Metallbeinchen).

Ein Prozessor **kann** komplett innerhalb eines der drei Chips liegen, tut es meist aber nicht, sondern setzt sich aus Komponenten zusammen, die auf mehrere Chips verteilt sind. Den Speicherzugriff für alle Aufgaben der verschiedenen Prozessoren erledigt z.B. immer Agnus. Keine der anderen Funktionseinheiten in den anderen beiden Chips ist deshalb komplett ohne den Adreßgenerator in Agnus. Deshalb kann man eigentlich von keinem der Prozessoren, die oben besprochen wurden, sagen, daß er in **einem** Chip liegt.

Der Einfachheit halber spricht man aber meist davon, daß ein Prozessor in dem Chip liegt, der die "wesentlichen" Teile des Prozessors enthält. Denise wird deshalb nicht nur als *Video-Chip*, sondern manchmal auch als *Video-Prozessor* bezeichnet.

"Andersrum" können aber theoretisch durchaus mehrere Prozessoren innerhalb eines Chips liegen. Dies ist z.B. beim Blitter und beim Copper der Fall, die sich im wesentlichen beide komplett in Agnus befinden. Wie Sie sehen, haben Chips und Prozessoren zwar viel miteinander zu tun, sind aber nicht identisch, sondern Begriffe, die sich auf sehr unterschiedliche Eigenschaften beziehen. (Auch ein *Microprozessor* muß nicht auf einem *Microchip* liegen und ein Chip, den man als *Microprozessor* **bezeichnet**, kann mehrere verschiedenen Prozessoren enthalten.)

Damit sollen die pingeligen Hardware-Details dann aber endgültig abgeschlossen sein. Ich hoffe, dem Leser in diesem Kapitel zumindest einen kleinen Eindruck von der Komplexität und Vielseitigkeit der Amiga-Hardware vermittelt zu haben, ohne dabei allzu viel Vorwissen vorausgesetzt zu haben.

14

Grafik

Der Amiga erstaunt immer wieder durch seine außergewöhnlichen Grafikfähigkeiten. Das ist denn auch kein Wunder, denn er ist der erste Computer seiner Preisklasse, der im Hinblick auf grafische Leistungsfähigkeit und Geschwindigkeit mit professionellen Grafiksystemen konkurrieren kann.

Im letzten Kapitel haben wir ja bereits die Spezial-Chips etwas kennengelernt, die für diese Grafikfähigkeiten verantwortlich sind. In diesem Kapitel wollen wir etwas weiter ausholen und ein wenig auf die Grundlagen moderner Computer-Grafik (natürlich speziell im Fall des Amiga) eingehen.

Zunächst wird es dabei um die statischen Aspekte der Bildschirmgrafik — die sogenannten Playfields — gehen und dann um die Möglichkeiten für bewegte Grafik.

14.1 Bitmapped Grafik

Die Grafik des Amiga ist *bitmapped*, d.h. irgendwo im Speicher des Amiga gibt es eine sogenannte Bitmap, die von der Videologik (innerhalb von Denise) in ein Bild umgesetzt wird. Wörtlich übersetzt ist eine Bitmap eine "Speicherlandkarte".

Dies bedeutet, daß die Farbe jedes Punktes auf dem Bildschirm durch ein oder mehrere Bits im Speicher bestimmt wird. Im simpelsten Fall gibt es eine 1:1-Zuordnung, bei der jedem Punkt auf dem Bildschirm genau ein Bit im

Speicher entspricht. Da der Amiga mehrere Farben gleichzeitig in einer Grafik verwenden kann, liegt der Fall natürlich etwas komplizierter. Dazu aber mehr weiter unten. Zur Veranschaulichung der wichtigsten Grundlagen will ich jetzt aber einmal annehmen, daß es sich um eine reine Schwarzweiß-Grafik handeln würde. Der Leser möge die folgenden Ausführungen bitte nicht für "Bit-Popelei" halten. Ein Verständnis dieser Vorgänge ist entscheidend für einige der raffinierteren Möglichkeiten der Amiga-Grafik.

Während der Elektronenstrahl über die Bildröhre fährt, holt der Videochip nacheinander die entsprechenden Bits aus dem Speicher und modifiziert den Strahl entsprechend. Ist das Bit gleich 1, so läßt er den Strahl in voller Stärke auf die Phosphor-Schicht der Bildröhre prallen, so daß ein heller Punkt entsteht. (In der Computergrafik werden Bildschirmpunkte übrigens oft auch *Pixel* genannt.) Ist das Bit gleich 0, so drosselt er den Strahl, so daß der Punkt schwarz bleibt. Die Abfolge, in der der Elektronenstrahl über den Bildschirm fährt, entspricht im simpelsten Fall genau der Reihenfolge der Bits im Speicher. Während der Strahl aber von links nach rechts und von oben nach unten schwenkt und dabei einen rechteckigen Bereich abdeckt, liegen die Bits im Speicher natürlich alle hintereinander bei steigenden Adressen. Eine rechteckige Anordnung der Bits müssen wir uns vorstellen; sie entspricht nicht dem Aufbau der Speicher-Chips.

Im allgemeinen kann man sagen, daß bei einer bitmapped Grafik eine eindeutige Zuordnung von einem Bit im Speicher (bei Schwarzweiß-Grafik oder genauer gesagt zweifarbigem Grafik) zu einem Punkt am Bildschirm besteht. Die Bits bilden eine Karte (engl. "map"), die der Videologik zeigen, an welchen Stellen der Bildschirm welche Farbe haben soll.

14.2 Farben

Schön und gut, aber wo bleibt die Farbe? — Es gibt mehrere Möglichkeiten, Farben mit diesem Bitmap-Prinzip zu kodieren. Um mehrere Farben darzustellen, reicht ein Bit nicht mehr aus. Ein Bit kann nur zwei Zustände haben (0 und 1), die zwei Farben zugeordnet werden können. Zwei Bits reichen aber schon aus um vier verschiedene Zustände darzustellen (00, 01, 10 und 11), drei reichen für acht Zustände usw.. Im allgemeinen gilt: N Bits reichen zur Kodierung von 2^N Farben. Diese N Bits entsprechen natürlich auch einer Zahl, wenn man sie sich als Zahlennotation zur Basis Zwei (duale Zahl) vorstellt. Mit drei Bits kann man so die Farben 0 bis 8 (dezimal) kodieren. Jede dieser Zahlen wird nun einer Farbe zugeordnet. Die

Videologik eines Computers bildet nun nicht nur ein Bit auf einen Bildschirmpunkt in einer von zwei Farben (z.B. schwarz und weiß), sondern 3 Bits auf einen Punkt ab, der insgesamt 8 verschiedenen Farben haben kann.

Die Speicherorganisation von farbiger Grafik im Amiga ist so ausgelegt, daß nicht mehrere im Speicher hintereinanderliegende Bits für einen Punkt verwendet werden, sondern "übereinanderliegende". Wie ist das gemeint? *Übereinanderliegend* soll in diesem Fall bedeuten, daß man mehrere Bitmaps gleicher Größe miteinander kombiniert, also quasi übereinanderlegt. Die Kombination der übereinanderliegenden Bits ergibt dann eine Zahl bzw. einen Farbwert für einen Punkt auf dem Bildschirm.

Bei vielen Computern wird heute ein ähnliches oder identisches Prinzip angewendet. Unterschiede gibt es nur dabei, wie die acht Zahlen (bei 3 Bits) als Farben interpretiert werden. Man kann sich natürlich acht beliebige Farben aussuchen und sie fest diesen acht Zahlen zuordnen. Bei einigen Modellen macht man sich auch die Tatsache zunutze, daß Farbfernsehbilder aus den drei Grundfarben rot, grün und blau durch additive Farbmischung zusammengesetzt werden. Jedes der drei Bits, die durch die Videologik auf einen Bildschirmpunkt abgebildet werden, wird deshalb einer dieser Grundfarben zugeordnet. Jede Farbe, deren Bit gleich 1 ist, wird dann bei dem jeweiligen Punkt "beigemischt". Ist das Bit gleich 0, bleibt die Farbe weg. Die acht Farben, die auf diesem Wege gemischt werden können, entsprechen dabei den acht möglichen Kombinationen von drei Bits.

Ziemlich genau dieses Prinzip wird standardmäßig z.B. bei IBM PCs verwendet. Es ist einfach zu realisieren, aber auch recht eingeschränkt. Mehr als acht Farben kann man nämlich so nicht darstellen, und auch diese acht Farben liegen eben "fest" und können nicht geändert werden.

14.3 Mehr Farbe ins Bild!

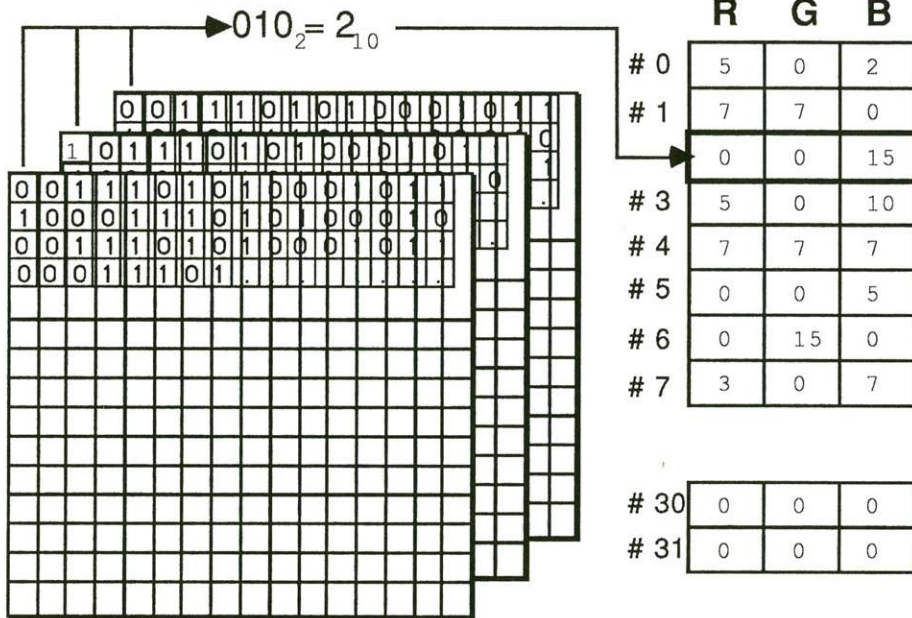
Um diese Feststellung zu vermeiden, wurde der Amiga so entwickelt, daß man die rote grüne und blaue Farbe nicht nur an- und abstellen, sondern sogar die "Intensität" selbst einstellen kann. Der Denise-Chip hat 32 *Farbregister*. Jedes dieser Farbregister ist 12 Bit breit. Davon werden vier Bit der Rot-Intensität, vier Bit der Grün-Intensität und weitere vier Bit der Blau-Intensität zugeordnet. Wie oben bereits erwähnt, ergeben N Bit 2^N Möglichkeiten. Jede Farbkomponente kann also in 16 Abstufungen der resultierenden Farbe beigemischt werden. In jedem Farbregister kann somit

eine von insgesamt 16 mal 16 mal 16 gleich 4096 möglichen Farben eingestellt werden.

Warum können wir am Amiga nun nicht alle 4096 Farben gleichzeitig darstellen? Wir bräuchten doch nur 12 Bitmaps "übereinander" zu legen. Berechnen wir dazu einmal den Speicheraufwand, der für eine solche tausendfarbige Grafik bei der kleinstmöglichen Auflösung benötigt würde. 320 Punkte horizontal mal 200 Punkte vertikal ergeben insgesamt 64000 Bildschirm-Punkte. 64000 Punkte sind 64000 Bit und somit 8000 Byte (1 Byte gleich 8 Bits). 12 Ebenen mit 8000 Byte wären dann bereits 96 000 Byte und somit rund mehr als ein Drittel des Speichers, den der Amiga mit 256 KByte Speicherausbau zur Verfügung hat.

Deswegen wurde der Denise-Chip des Amiga so entwickelt, daß er maximal 5 übereinandergelagerte Bitmaps verwalten kann.¹ Es gibt damit also 32 verschiedene Farbmöglichkeiten. Denise greift bei der Darstellung der Grafik ganz einfach, je nach der Kombination der Bits in der verschiedenen Bitmaps, auf das Farbbregister mit der entsprechenden Nummer zu. So ist es möglich, von 4096 verschiedenen Farben maximal 32 davon gleichzeitig darzustellen. Verwendet man weniger Bitmaps, kann man natürlich entsprechend weniger Farbbregister benutzen. Wie dieses Prinzip mit drei Bitmaps (also 8 Farben) funktioniert, sehen Sie in der folgenden Grafik.

¹ Diese Einschränkung hat allerdings nicht nur Sparsamkeitsgründe. Es ist nämlich bei den heutzutage verwendeten Speichern leider nicht möglich, mehr als eine bestimmte Menge Daten pro Zeiteinheit aus dem Speicher zu lesen. Gäbe es mehr Bitmaps, würde dies aber auch eine größere Menge Daten bedeuten. Die physikalische Leistungsgrenze der im Amiga verwendeten Speicherchips liegt aber ziemlich genau bei fünf oder sechs mal 320 Bits (=40 Byte) pro Zeiteinheit, in der der Elektronenstrahl eine Zeile am Bildschirm zeichnet. Mehr ist nicht "drin", ohne daß der Amiga wesentlich aufwendiger und teurer würde.

BitMap aus drei Ebenen32 Farb-Register
(davon 8 verwendbar)**Bild 14 - 1: Die Zuordnung von Farben zu Bildschirmpunkten**

Manchmal spricht man in diesem Zusammenhang auch von mehreren *Bit-Ebenen* (engl. *bitplanes*) die zusammen eine (mehrschichtige) Bitmap bilden. Wir wollen aber so weit wie möglich versuchen, mit dem Wort Bitmap alleine auszukommen.

14.4 Noch mehr Farbe ins Bild!

Die bitmapped Grafik des Amiga benötigt — wie oben gesehen — maximal 5 Bitmaps, um 32 verschiedene Farben gleichzeitig am Bildschirm darzustellen. Es gibt aber auch einen Modus, der 6 Bitmaps benötigt und theoretisch alle 4096 möglichen Farben gleichzeitig am Bildschirm zeigen. Dieser Modus heißt *Hold-And-Modify-Modus*, oder kurz *HAM*. Der Name bedeutet soviel

wie "festhalten und modifizieren" und genau das macht die Video-Hardware in diesem Modus.

Die sechs Bits, die für jeden Bildschirmpunkt in diesem Modus zur Verfügung stehen, werden dabei in einer speziellen Weise genutzt, die vom üblichen Bitmap-Prinzip etwas abweicht. Die Bitmaps Nummer 5 und 6 bestimmen dabei, wie die restlichen Bitmaps (Nummer 1 bis 4) interpretiert werden.

Ist die Bitkombination von Bitmap 5 und 6 gleich 00, so werden die restlichen 4 Bits als die Nummer eines Farbregisters interpretiert, das die Farbe dieses Punktes enthält. In diesem Fall wirkt der HAM-Modus als ein etwas speicherverschwendender Grafikmodus mit 4 Bit-Ebenen (= 16 Farben).

Ist die Bitkombination aber 01, tritt der eigentliche HAM in Aktion. Dabei wird erst einmal die Farbe *festgehalten (Hold)*, die der unmittelbar vor (links von) dem neuen Punkt liegende Bildpunkt hat. Die Blauintensität dieser Farbe wird jedoch verworfen und durch den Wert ersetzt, der in den Bitmaps 1 bis 4 für diesen Punkt eingetragen ist. Bits 4 bis 1 *modifizieren (Modify)* also den Blauwert des vorangegangenen Punktes und ergeben damit die neue Farbe für den aktuellen Punkt.

Ist die 6-5-Bit-Kombination 10, geschieht das Gleiche, diesmal allerdings ersetzen die Bitmaps 1 bis 4 die Rotintensität. Und wenn die Kombination gleich 11 ist, werden die Bitmaps 1 bis 4 dazu benutzt, die Grünintensität zu ersetzen.

Wie leicht zu sehen ist, können auf diese Weise wirklich alle 4096 Farben erzeugt werden. Es ist zwar nicht möglich jeden beliebigen Punkt auf jede beliebige Farbe zu setzen, aber mit etwas Geschick kann man diese Einschränkung recht gut umschiffen. Nach spätestens drei Punkten ist man immer bei der Farbe, die man haben will. Dazu lädt man einfach nacheinander alle drei Farb-Komponenten Rot, Grün und Blau einzeln. Setzt man die 16 Farben, die man in jedem Punkt direkt aus einem der Farbregister anwählen kann, geschickt ein, so werden die Einschränkungen noch geringer.

14.5 Der Interlace-Modus

Die Auflösung in vertikaler Richtung beträgt auf dem Bildschirm normalerweise 200 Punkte. Dies ist die Grenze, die von einigermaßen

preiswerten Monitoren und Farbfernsehern gesetzt werden, bei der noch ein relativ flimmerfreies Bild möglich ist. Die *Bildwiederholfrequenz* beträgt bei dieser Auflösung 60 Hz. D.h. 60-mal pro Sekunde wird das Bild neu gezeichnet — schneller, als das Auge einzelne Bilder trennen könnte. Der Interlace-Modus ist ein Mittel, die horizontale Grafikauflösung auf dem sichtbaren Bildschirm zu verdoppeln. Mit Hilfe des *Interlace* (engl. für *verschränkt, überlagert*) ist es möglich, die Anzahl der Zeilen des Videobildes von 200 auf 400 zu erhöhen.

Das funktioniert folgendermaßen: Der Bildschirm wird statt in 200 in 400 Zeilen aufgeteilt. In diesen 400 Zeilen werden zwei Bilder mit je 200 Zeilen gezeigt, wobei jedes dieser beiden Bilder aber nur in jede zweite Zeile gezeichnet wird. Das erste in die ungeraden, das zweite in die geraden Zeilen. Zusammen ergeben die beiden Bilder ein Bild von 400 Zeilen. Da jedoch jede Zeile nur bei jedem zweiten Durchlauf des Elektronenstrahls neu gezeichnet wird und zwischendurch schwarz bleibt, liegt die *effektive* Bildwiederholfrequenz nur noch bei 30 Hz, was das Bildflimmern in den Bereich des deutlich Wahrnehmbaren absenkt.

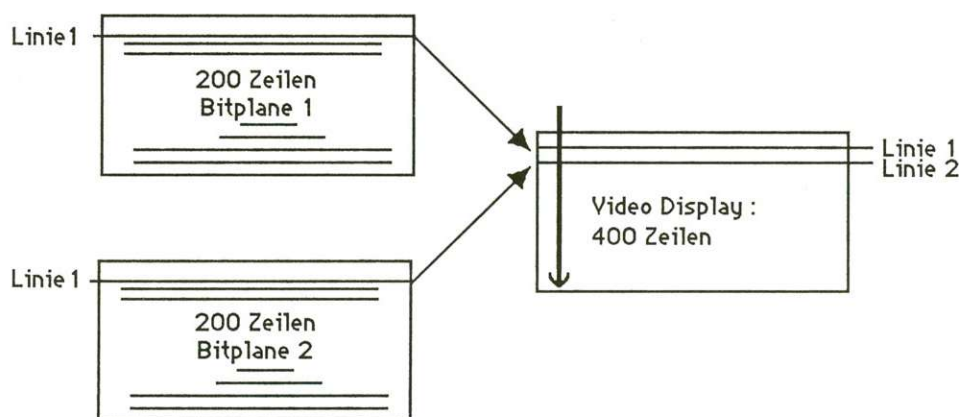


Bild 14 - 2: Das Interlace-Prinzip

Ein ganz normaler Farbfernseher tut übrigens genau das selbe. Die Fernsehanstalten senden ebenfalls jede 50stel Sekunde eine von zwei sogenannten Halb-Bildern, die nacheinander — um eine Zeile versetzt — am Bildschirm gezeichnet werden. Das träge menschliche Auge kombiniert diese beiden Halbbilder dann zum kompletten Fernseh-Bild, das bei der effektiven Bildwiederholfrequenz von nur noch 25 Hz dann aber ebenfalls deutlich flimmert.

14.6 Übergroße Grafiken

Der Amiga bietet die Möglichkeit, Grafiken darzustellen, die größer sind als der gesamte Bildschirm. Ein *Scrolling*, also ein stufenloses Bewegen dieser Riesengrafik, wird durch Änderung entsprechender Register innerhalb des Videochips Denise erreicht. Dies ist beim Amiga einfacher, als man vielleicht denkt.

Oben hatten wir ja schon die Bitmaps beschrieben, nach denen Denise die Bildschirmgrafik zeichnet. Diese liegen aber nicht an festen Stellen im Speicher, sondern können irgendwo angelegt werden, wo es dem zuständigen Programm gerade gefällt.² Denise muß nur mitgeteilt werden (indem man eine entsprechende *Adresse* in eines ihrer Register schreibt), wo die Bitmaps liegen.

In anderen Registern kann man Denise nun auch mitteilen, wie breit die Grafik "wirklich" ist, was nichts mit der Anzahl von Punkten in einer Bildschirmzeile zu tun haben muß, und welcher Teil der Grafik wo am Bildschirm dargestellt werden soll. Wenn man will, kann man auch weniger als 320 mal 200 Punkte darstellen, wobei dann aber nicht mehr der ganze Bildschirm genutzt wird.

Will man den sichtbaren Ausschnitt verändern, so muß man einfach nur die Werte in den entsprechenden Denise-Registern modifizieren. Es brauchen keine Daten im Speicher verschoben und keine komplexen Kalkulationen durchgeführt werden, wie auf andern Computern. Deshalb ist die Verschiebung auch sehr fließend und schnell.

14.7 Der Dual-Playfield-Modus

Der Amiga kann aber auch in der Hinsicht Playfields noch mehr! Durch Aktivieren des Dual-Playfield-Modus kann man mehrere Playfields übereinanderlagern. Dabei sieht man durch gewisse "transparente" Flächen in

² Diese Aussage stimmt nicht ganz. Alle Daten, auf die die drei Spezial-Chips zugreifen — also auch eine Bitmap — müssen in den unteren 512 KByte des Speichers liegen. Diese Einschränkung erlaubte es, diese Chips (vor allem Agnus) einfacher und damit billiger zu halten.

einem Playfield auf ein dahinterliegendes hindurch. Man kann also zwei voneinander unabhängige Playfields gleichzeitig darstellen und mit entsprechenden Prioritäten über- oder untereinander legen. Grafiken, die aus zwei logisch getrennten Ebenen bestehen, können so wesentlich leichter dargestellt werden.

Teile von Zeichnungen im unteren Playfield können in diesem Modus nur dann sichtbar werden, wenn das obere Playfield an diesen Stellen durchsichtig ist. Man kann Grafiken so sehr leicht auf bestimmte Bereiche beschränken. Auch können beide Playfields unabhängig voneinander verschoben werden (s.o.), was ebenfalls für einen *Blick-aus-dem-Fenster-Effekt* sehr praktisch ist.

Verwendet man zwei Playfields, muß man allerdings einige Einschränkungen in Kauf nehmen. Jedes der beiden Playfields kann nur aus einer, zwei oder drei Bitmaps bestehen. Die Farbregister 0 bis 7 werden dabei dem ersten Playfield zugewiesen, die Register 8 bis 15 dem zweiten. Zusätzlich können die Farben in den Registern 0 und 8 nicht verwendet werden, da in beiden Playfields die jeweils erste Farbe als "transparent" gilt, durch das darunterliegenden Bild durchscheint. (Hinter dem unteren Playfield scheint der Bildschirm-Hintergrund durch.) Der Dual-Playfield-Modus bietet jedoch für bestimmte Anwendungen genügend Vorteile, um diese Einschränkungen zu rechtfertigen.

14.8 Sprites

Sprites sind bewegbare Objekte, die durch die Hardware auf den Bildschirm gebracht werden. Das heißt, der Benutzer muß nicht erst — wie bei den Softwaresprites an manchen Computern — sein Objekt mit dem Grafikspeicher mischen. All diese Aufgaben erledigt bereits die Hardware.

Ein Sprite kann am Amiga 16 Pixels breit und beliebig hoch sein. Innerhalb der Spritegröße kann jede beliebige Form enthalten sein, zum Beispiel so:

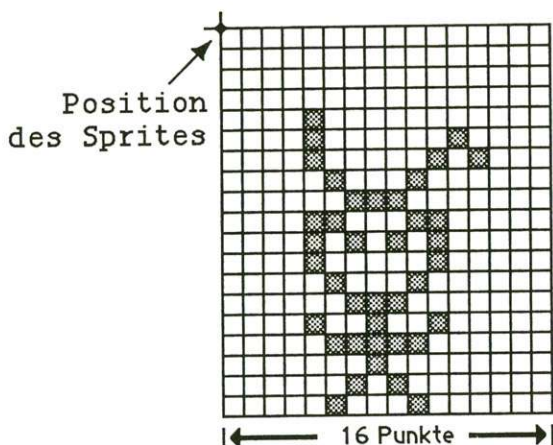


Bild 14 - 3: Ein Sprite

Um ein Sprite an einer bestimmten Stelle des Bildschirms erscheinen zu lassen, braucht man nur die X- und Y-Position der Stelle, an der die obere linke Ecke des Sprites sitzen soll, in dafür bestimmte Register innerhalb von Denise schreiben. Ändert man die Werte in diesen Registern, erscheint das Sprite automatisch an der neuen Stelle und verschwindet an der alten. Eine "Bewegung" dieses kleinen Bildes ist so leicht und sehr schnell möglich.

Die Größe der Punkte, mit denen die Sprites dargestellt werden, ist dabei unabhängig von der Auflösung der Grafik, die gerade im Playfield dargestellt wird. Ein Sprite-Punkt ist immer so groß wie ein Punkt in der niedrigsten Grafik-Auflösung (320 * 200) des Amiga.

Genauso wie bei Playfields erfolgt auch die Darstellung und Färbung von Sprites. Bei Sprites werden allerdings (fast) immer nur zwei Bits verwendet um die Farbe für einen Punkt anzugeben, weswegen ein Sprite nur drei Farben (plus transparent) haben kann. Statt Bitmaps werden bei Sprites aber *Worte* kombiniert. Ein Wort besteht aus 16 Bit und jede Zeile eines Sprites ist durch zwei Worte im Speicher festgelegt. Die ersten 16 Bit und damit das erste Wort (*Low-Word*) stellen praktisch die erste Bitmap für diese Sprite-Zeile und das zweite Wort (*High-Word*) die zweite Bitmap dar. Das erste Bit des ersten Wortes bestimmt also zusammen mit dem ersten Bit des zweiten Wortes die Farbe des ersten Punktes in einer Sprite-Zeile, usw..

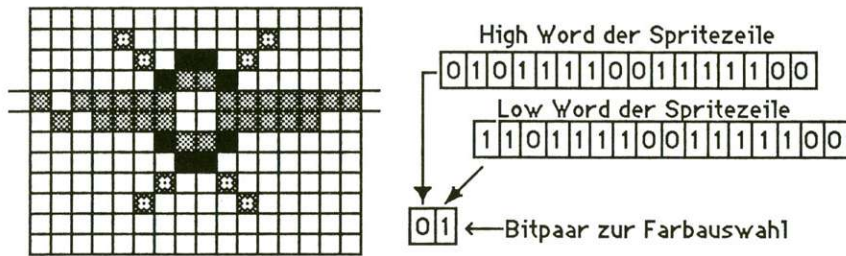


Bild 14 - 4: Farben in einem Sprite

Auch die Farben der Sprites kommen aus den 32 Farbregistern innerhalb von Denise (s.o.). Für die Sprites werden die Farbregister 16 bis 31 verwendet, womit für acht Sprites nur 16 verschiedene Farben zur Verfügung stehen. Jeweils zwei aufeinanderfolgende Sprites müssen sich deshalb vier Farbregister "teilen". Diese Einschränkung dürfte für die meisten Anwendungen aber nicht allzu tragisch sein.

Vier Farben pro Sprite sind für manche Zwecke jedoch ein bißchen zu wenig. Deshalb ließ man sich beim Amiga etwas einfallen, diese Einschränkung – unter Opferung anderer Vorteile – zu umgehen. Um eine größere Farbwahl zu haben, werden zwei Sprites zu einem verbunden. So kann man Sprite 0 mit Sprite 1 verbinden, Sprite 2 mit Sprite 3, usw.. Damit stehen jetzt vier Bits pro Punkt für die Farbauswahl zur Verfügung und somit insgesamt 15 Farben plus Transparenz.

Eine andere gravierende Einschränkung der Sprites ist ihre Zahl. Acht bewegliche Objekte sind für viele Zwecke zu wenig. Deshalb gibt es zwar nur acht (oder vier) *Hardware-Sprites* oder *Sprite-Kanäle* innerhalb von Denise, man kann jedes(n) aber mehrmals benutzen, während ein Bild gezeichnet wird. Hierfür wird die Tatsache genutzt, daß die Videologik innerhalb von Denise, die für ein bestimmtes Sprite zuständig ist, ja nicht mehr gebraucht wird, sobald das untere Ende des Sprites auf dem Bildschirm vom Elektronenstrahl gezeichnet wurde. Es wäre doch eine glatte Verschwendung, diesen Sprite-Kanal des Chips nun brach liegen zu lassen. Wenn ein Sprite "zu Ende ist", kann Denise deshalb blitzschnell auf die Beschreibung eines anderen, auch völlig anders aussehenden Sprites, überwechseln und zeigt dieses an. Dieses zweite Sprite muß dazu aber vollständig unterhalb des ersten Sprites auf dem Bildschirm zu sehen sein. Man kann zwei Sprites also nur untereinander, aber nicht nebeneinander über denselben Sprite-Kanal darstellen.

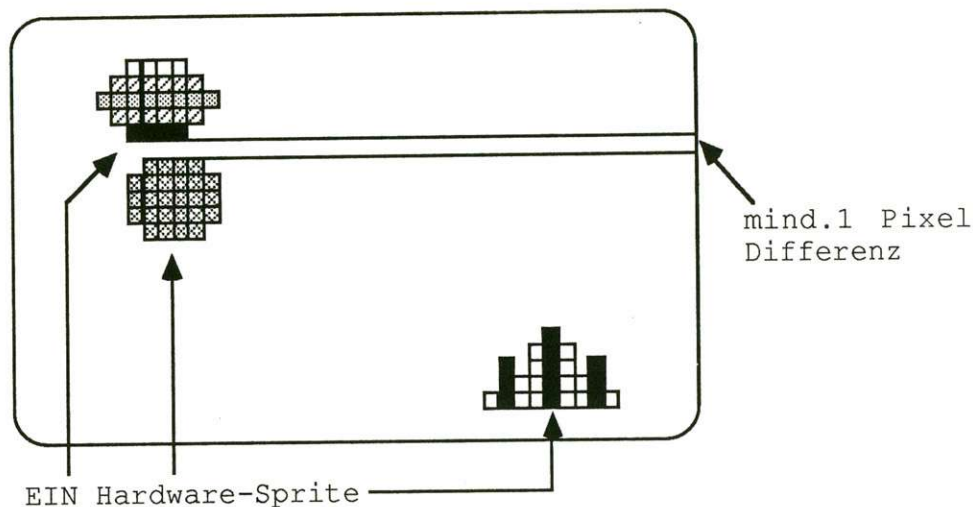


Bild 14 - 5: Wiederverwendung eines Sprite-Kanals

Wenn mehrere Sprites gleichzeitig zu sehen sind, kann es natürlich auch vorkommen, daß sie sich überlappen. Dafür haben die verschiedenen Sprites *Prioritäten* voreinander. Beim Amiga sind diese Prioritäten festgelegt, man kann sie also nicht ändern. Wenn Sprite 1 und Sprite 2 auf den selben Koordinaten liegen, überlappen sie sich so, daß Sprite 1 vor Sprite 2 erscheint, also das zweite Sprite überdeckt. Die Prioritätsfolge aller Sprites zeigt das folgende Bild.



Bild 14 - 6: Sprite-Prioritäten

Wenn Sprites sich überlappen (*zusammentreffen*), oder sich über bestimmten Gebieten des Playfields befinden, so kann das eine wichtige Information für ein Programm sein. Denise enthält deshalb Hardware, die feststellen kann, ob zwei oder mehr Sprites an einem Bildschirm Punkt übereinanderliegen oder über einem Punkt des Playfields, der nicht die Farbe 0 hat. Dazu werden

bestimmte Signal-Bits (Flags) in Denise auf 1 gesetzt, sobald eine der genannten Bedingungen eintritt.

14.9 BOBs

Trotz vieler Tricks und Kniffe bleiben Sprites in ihren Möglichkeiten aber doch recht eingeschränkt. Es gibt nur 8 Stück davon, sie sind 16 Punkte breit (schmal) und haben normalerweise nur 3 Farben. Alle diese Einschränkungen kennen *BOBs* (*Blitter Objects*) nicht. Im Gegensatz zu Sprites sind die BOBs aber kein Konzept der Amiga-Hardware, sondern der Grafik-Bibliotheken, also der Amiga-Software. Diese Software nutzen allerdings intensiv ein besonderes Stück Hardware, den Blitter, von dem die BOBs auch ihren Namen haben.

BOBs sind sogenannte *Softwaresprites*. Es sind ganz einfach hochauflösende Grafiken, die man verschieben kann, ohne die "darunterliegende" Grafik dabei zu stören. BOBs kann man sich vorstellen wie einen Ausschnitt aus einem Playfield, der zu einem eigenständigen Objekt geworden ist. BOBs können deshalb genauso groß und farbig werden, wie das Playfield auch, sind im allgemeinen (jedoch nicht zwangsläufig) aber kleiner als ein Playfield.

Bei BOBs wird die kleine Bitmap, die ihr Aussehen bestimmt, wirklich in die Playfield-Bitmap hineinkopiert. Vorher wird aber der alte Inhalt der Playfield-Bitmap an dieser Stelle in einem Puffer gesichert. Wird das BOB bewegt, so wird diese gesicherte Kopie des alten Playfields wieder in die Playfield-Bitmap zurückgeschrieben und das BOB an der neuen Stelle gezeichnet. Beim Bewegen eines BOBs müssen also wirklich Daten im Speicher hin- und hergeschoben werden. Je größer das BOB ist, desto mehr Daten sind es. Das kostet wesentlich mehr Zeit als das Laden eines Denise-Registers mit einem neuen Wert für die Bewegung eines Sprites. Gäbe es nicht den Blitter, der speziell für solche "Datenschaufeleien" geschaffen ist, wären BOBs viel zu langsam um überhaupt praktikabel zu sein.

BOBs sind, wie hoffentlich zu sehen war, wesentlich flexibler und vielseitiger als Sprites, da sie behandelt werden können wie der Teil einer beliebigen Playfield-Grafik. Dieser Vorteil ist aber auch ihr Nachteil. Da Sie nicht durch Hardware dem Playfield überlagert werden können, müssen sie durch Speicheroperationen (von teilweise großen Datenmengen) dargestellt und bewegt werden. Dies kostet Zeit! BOBs sind deshalb bei weitem nicht so beweglich wie Sprites.

14.10 AnimObjects

BOBs und Sprites eignen sich ideal für die Darstellung bewegter Objekte in einem Programm. Mit beiden kann man Bilder (mit verschiedenen großen Einschränkungen) recht mühelos über den Bildschirm schieben. Es reicht jedoch für die Simulation der Bewegung wirklicher Bilder meist nicht aus, einfach ein "starrtes Bild" zu verschieben.

Die nächste Stufe der Komplexität bei der Erzeugung bewegter Grafiken im Amiga sind die *AnimObjects*. AnimObjects sind eine Reihe von BOBs, die mit einigen anderen zusätzlichen Daten zu einem Objekt zusammengefaßt werden. AnimObjects enthalten unter anderem mehrere verschiedene BOBs. Ein AnimObject kann nun aber nicht nur über den Bildschirm bewegt werden, sondern bei der Bewegung wird auch ein BOB gegen das nächste in der Liste ausgetauscht.

Die Reihenfolge und Geschwindigkeit, in der die verschiedenen BOBs gegeneinander ausgewechselt werden, ist vom Programmierer frei wählbar. Gleichzeitig kann er dem AnimObject noch eine Geschwindigkeit mitteilen, mit der die BOBs sich über den Bildschirm bewegen sollen und eine Beschleunigung, die angibt, mit welcher Rate sich diese Geschwindigkeit ändern soll.

Ein AnimObject könnte also die verschiedenen Bewegungsphasen einer gehenden Person enthalten. Je mehr Phasen man verwenden würde (und je mehr Speicher man dafür verwenden würde), desto flüssiger würde diese Bewegung wirken. Gibt man dem AnimObject dann eine Geschwindigkeit, so könnte diese Person z.B. langsam oder schnell über den Bildschirm schreiten.

Das bekannte Robocity-Demo, das sich ebenfalls auf den Demo-Disketten befindet, die jeder Amiga-Händler bekommt, nutzt exakt diese Möglichkeit. Zwei Roboter, zwei Roboter-Hunde und eine Art Springmaus bewegen sich darin gleichzeitig und recht flüssig vor einem festen Hintergrund.

Die Software, die die AnimObjects bewegt, registriert dabei sogar, wenn zwei solche Objekte zusammentreffen und das Programm kann dies abfragen und darauf reagieren. AnimObjects sind also echte *Supersprites*, die die Programmierung sich ändernd und dabei gleichzeitig bewegender Bilder extrem leicht macht und fast alle dabei anfallenden Aufgaben automatisch erledigt.

14.11 Der PAL-Amiga

Alle bisherigen Erläuterungen und Zahlen in diesem Kapitel stimmen nicht! Sie meinen, es wäre ein wenig spät, Ihnen das zu sagen? Na ja, ganz so schlimm ist es auch wieder nicht mit den falschen Zahlen. Man könnte vielleicht besser sagen, daß die Zahlen *nicht ganz* stimmen, da sie für die US-amerikanische Version des Amiga gelten, in denen der Video-Chip Denise nach der amerikanischen Fernsehnorm NTSC ausgelegt ist.

Die deutschen Amigas werden ca. ab Sommer mit einem geänderten Video-Chip ausgeliefert, der dann der PAL-Farbfernsehnorm entspricht, die hierzulande üblich ist. Daß die PAL-Norm in verschiedener Hinsicht leistungsfähiger ist als die uralte NTSC-Norm, wollen die Amiga-Ingenieure ausnutzen. So wird die maximale vertikale Auflösung für Grafiken bei dieser neuen Version von Denise um ca. 25% auf 256 Zeilen bzw. 512 Zeilen im Interlace-Modus gesteigert.

Keine Angst! Alle amerikanischen Programme werden nach wie vor lauffähig sein (wenn es auch vorkommen kann, daß sie die höhere Auflösung nicht nutzen können). Bei Bedarf stehen aber mehr Zeilen für detailliertere bzw. größere Zeichnungen zur Verfügung.

Zudem werden die neuen Versionen von Denise einen komplett neuen Betriebsmodus beinhalten, der es erlaubt — ohne die Einschränkungen des HAM-Modus — 64 verschiedene Farben gleichzeitig in einem Screen zu zeigen. Dieser zusätzliche Modus trägt den Namen *Extra Halfbright* und benötigt wie der HAM-Modus 6 Bit-Ebenen.

Der PAL-Amiga ist aber nicht — wie mancher von der Computer-Industrie verunsicherte Leser jetzt vielleicht meinen wird — eine Vision der Amiga-Ingenieure, sondern Wirklichkeit. Es gibt ihn bereits und er wurde vom Autor dieser Zeilen auch bereits in Aktion gesehen. Die Großserien-Produktion soll laut Planung von Commodore USA aber erst im Frühjahr 1986 anlaufen.

Ein zusätzlicher Vorteil der PAL-Amigas wird natürlich auch sein, daß man diese an einen preiswerten Farbfernseher anschließen kann und keinen teuren Monitor mehr benötigt. Der Einstiegspreis für die billigste Konfiguration des Systems dürfte dadurch mindestens 1000 DM billiger werden.

14.12 Zusätzliche Grafik-Hardware

Alle Fähigkeiten der Grafik-Hardware des Amiga, die bislang erwähnt wurden, dienten eigentlich nur (?) dazu, Grafiken hoher Qualität **aus dem Innern des Computers nach außen** (auf den Bildschirm) zu bringen. Für viele Anwendungen ist es jedoch auch nötig, Bilder **von außen in den Computer** zu schaffen, wo sie dann verarbeitet werden – um sie danach vielleicht wieder in veränderter Form auf dem Bildschirm zu zeigen. Auch hierzu ist der Amiga fähig, allerdings nur mit zusätzlicher Hardware. Der FrameGrabber und das Genlock-Interface sind zwei solche Zusatzbauteile, die über die Erweiterungs-Schnittstelle an der rechten Seite der Systemeinheit angeschlossen werden können.

14.12.1 Das GenLock-Interface

Die einfachste dieser beiden Erweiterungen ist das *GenLock-Interface*. Es dient dazu, das Bild aus einer externen Video-Quelle (z.B. einen Video-Recorder, eine Video-Kamera oder ein Bildplattenspieler) mit dem Bild, das Denise erzeugt, zu mischen. Das Video-Bild ersetzt dabei auf dem Bildschirm nur die Flächen, die mit der Hintergrundfarbe (aus Farbregister Null) gefüllt sind. Alle anderen Farben überdecken das Video-Bild.

Aus einer solchen Kombination eines von außen eingespielten Hintergrunds mit überlegten Computergrafiken ergeben sich interessante Anwendungsmöglichkeiten. Wer zum Beispiel ungern die Tagesschau verpaßt, aber weiter programmieren möchte, blendet einfach das Fernsehbild im Hintergrund ein!

Abgesehen von solchen Spielchen existieren aber auch ernsthafte Anwendungen, vor allem auf den Gebieten Ausbildung, Werbung und Spiel. Hierbei würde man z.B. zunächst eine Videoplate oder -Kassette mit dem Lehrgang bzw. einer Spielszene erstellen und diese dann später überlegt mit Texten (oder Skizzen) aus dem Computer zeigen. Hierbei würde das Videobild den größten Anteil des Bildes einnehmen und die Computergrafiken nur ergänzende und sich schnell jeder Situation anpassende Zusatzinformationen oder Fragen bieten.

Je nach Reaktion des Betrachters könnte der Amiga nun dafür sorgen, daß eine andere Lektion gezeigt wird, die letzte wiederholt wird, bzw. ein interaktives Spiel einen anderen Verlauf nimmt. Zu solchen Anwendungen

würde allerdings gehören, daß der Amiga gleichzeitig auch den Videorecorder oder Bildplattenspieler steuert. Dies ist aber das geringste Problem.

14.12.2 Der FrameGrabber

Während das GenLock-Interface nur zum Mischen bzw. Überlagern von Bildern verwendet werden kann, holt der *FrameGrabber* (zu deutsch etwa *Bild-Fänger*) wirklich ein Video-Bild in den Speicher des Amiga. Man nennt dies auch *digitalisieren* eines Bildes. Es liegt dann als Bitmap am RAM und kann natürlich auch sofort wieder am Bildschirm gezeigt werden.

Mit dem FrameGrabber wäre es z.B. aber auch möglich, ein Bild zunächst mit einer Video-Kamera aufzunehmen und dann mit einem Mal-Programm wie Deluxe Paint nachzubearbeiten oder zu verfremden. Viele der Abbildungen im Handbuch des Amiga sind auf diese Weise entstanden. Gerade auch für Werbezwecke eröffnet dies ungeahnte Möglichkeiten.

Statt ein Bild von einem Menschen (unter Mithilfe eines Programms) bearbeiten zu lassen, kann man aber natürlich auch versuchen, es gleich mit einem speziell dafür geeigneten Programm zu analysieren oder zu modifizieren. Im Extremfall könnte ein Programm sogar versuchen, zu "erkennen", was das Bild zeigt. Dies würde dann aber schon in den Bereich der künstlichen Intelligenz hineingehen und den Amiga in den meisten Fällen wahrscheinlich etwas überfordern. Einfache Überwachungsaufgaben könnte man aber mit dem FrameGrabber durchaus dem Amiga überlassen.

14.13 Anwendungen der Amiga-Grafik

Zum Abschluß dieses kleinen Überblicks über die Möglichkeiten der Amiga-Grafik sollen Sie aber auch noch einige praktische Anwendungen dieser Möglichkeiten kennenlernen. *Kennenlernen* ist eigentlich nicht das richtige Wort dafür. Sie kennen diese Anwendungen nämlich schon aus den vorangegangenen Kapiteln. Nur wußten Sie bislang wahrscheinlich noch nicht, welche "Ursachen" diese Eigenschaften des Amiga hatten.

Die folgenden Abschnitte zeigen nur einige wenige Beispiele für Anwendungen der Grafik-Konzepte in der Hardware und Software des Amiga auf. Wenn Sie sich aber ein wenig näher mit der Amiga-Grafik

beschäftigen, werden Sie bestimmt bald noch weitere Anwendungen entdecken.

14.13.1 Fenster

Die Flexibilität des Fenster-Konzeptes auf dem Amiga und die Geschwindigkeit, mit der Fenster manipuliert werden können, hat zu einem großen Teil seine Ursprünge in den oben beschriebenen Grafik-Konzepten. So kann der Blitter z.B. auf verschiedene Weise dafür eingesetzt werden, Fensterinhalte *aufzufrischen*.

Die in einem Window dargestellte Grafik liegt ja innerhalb eines bestimmten Speicherbereichs — so, wie im Zusammenhang mit Bitmap-Grafik erklärt wurde. Legen wir jetzt ein anderes Window über das erste, so verschwindet die Grafik darunter. Die darunterliegende Grafik des verdeckten Fensters wird dabei gelöscht. Sobald ein Teil dieses verdeckten Gebietes aber wieder freigelegt wird, muß er aufgefrischt werden, d.h. der alte Inhalt dieser Fläche muß restauriert werden.

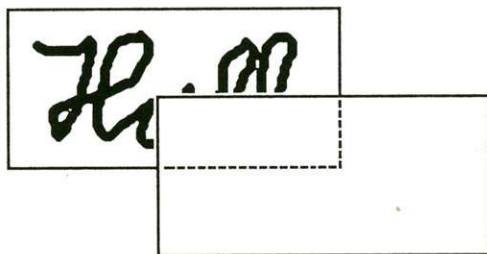


Bild 14 - 7: Ein Fenster wird verdeckt

Der Amiga unterscheidet grundsätzlich drei Techniken für dieses Auffrischen: **Simple Refresh**, **Smart Refresh** und **Super Bitmap**. Diese Methoden stellen verschiedene Kompromisse zwischen Speicherplatzbedarf und Einfachheit des Auffrischungsvorgangs dar. Welche bei einem bestimmten Fenster angewendet wird, entscheidet der Programmierer des für das jeweilige Fenster zuständigen Programms. Die Workbench selbst arbeitet mit der Smart Refresh Methode, die eine sinnvolle Zwischenlösung darstellt.

Die SimpleRefresh-Technik hätte man auch *No Refresh* nennen können. Die Grafikfähigkeiten des Amiga spielen hier keine große Rolle. Dabei wird nämlich ganz einfach der Grafikspeicher des unten liegenden Windows gelöscht, wenn ein anderes darübergelegt wird. Sobald ein solches teilweise verdecktes Fenster aufgefrischt werden muß, ergeht eine Nachricht an das

zuständige Programm, das dann selbst dafür sorgen muß, daß der Inhalt restauriert wird. Der entscheidende Vorteil dieser Methode ist, daß kein zusätzlicher Speicherplatz dafür benötigt wird.

Methode 2 ist die SmartRefresh-Technik. Bei ihr werden die verdeckten Teile eines Fensters durch den Blitter in einen Puffer gelegt, bevor sie verdeckt werden.

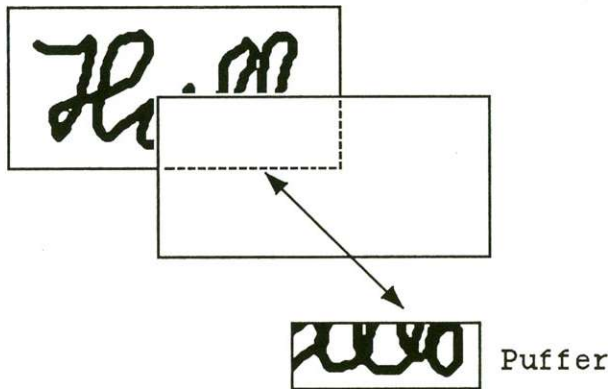


Bild 14 - 8: *Ein Smart-Refresh-Fenster*

Auch wenn die Grafik geändert wird, während sie teilweise verdeckt ist, wird dieser Puffer modifiziert. Sobald verdeckte Teile wieder freigelegt werden, wird ihr Inhalt aus dem Puffer wiederhergestellt. Auch hierfür kommt natürlich wieder der blitzschnelle Blitter zum Einsatz, weshalb man oft gar nichts von diesem Vorgang merkt und meint, das verdeckte Bild hätte wirklich unter dem anderen Fenster gelegen. Das verbraucht mehr Speicherplatz als die Simple-Refresh-Methode, ist aber schneller und einfacher für das Programm.

Wird das Fenster jedoch vergrößert, kann auch die Smart-Refresh-Methode nicht mehr helfen. Die dadurch entstehenden leeren Flächen muß das Programm wieder selbst ausfüllen, da diese ja nie verdeckt und deshalb auch nicht gepuffert wurden.

Die Super-Bitmap-Technik merzt auch diese letzte Schwäche aus, ist allerdings enorm speicherintensiv. Zudem wird das Zeichnen von Grafiken dadurch etwas langsamer. Bei diesem Verfahren wird ein Speicherbereich reserviert, der die komplette Zeichnung, von der ein Ausschnitt innerhalb eines Fenster erscheint, gespeichert werden kann. Dieser Speicherbereich heißt *Super-Bitmap*. Alle Zeichen-Operationen werden dann zunächst einmal in der Super-Bitmap durchgeführt, die immer den aktuellen Zustand des kompletten Bildes enthält. Nach Beendigung jeder Änderung wird ein

fenstergroßer Ausschnitt aus dieser Super-Bitmap dann in das Fenster kopiert. Auch hierfür wird natürlich wieder der Blitter eingesetzt, das Arbeitstier im Amiga.

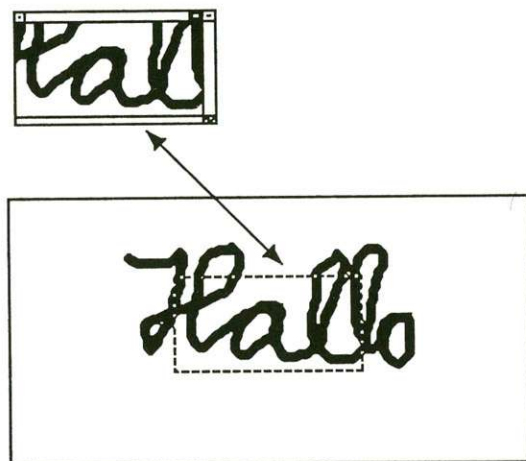


Bild 14 - 9: *Ein Super-Bitmap-Fenster*

Als Kompromiß dieser drei verschiedenen Methoden ist die SmartRefresh-Methode wohl die beste. Sie benötigt zwar etwas zusätzlichen Speicher, aber bei weitem nicht soviel wie das Super-Bitmap-Prinzip und spart dem Programm doch eine Menge Arbeit.

Die drei Demo-Programme in der Schublade Demos auf der Original-Workbench-Diskette verwenden übrigens alle verschiedene Methoden zur Fensterauffrischung. Probieren Sie sie ruhig einmal aus und versuchen Sie herauszufinden, welches Programm welche der drei Methoden anwendet. Es ist recht deutlich zu sehen.

14.13.2 Screens

Die virtuellen Bildschirme oder Screens des Amiga sind eine trickreiche Anwendung des Coppers, wie ja auch schon einige Male angedeutet wurde.

Der Copper ist ja fähig, jedesmal ein kleines Programm auszuführen, wenn der Elektronenstrahl des Monitors bei einer ganz bestimmten Bildschirm-Position vorbeikommt. Die Programmiersprache des Coppers ist allerdings sehr simpel. Alles, was er kann, ist Daten in Register der drei Spezial-Chips

zu schreiben. Dies reicht für die Realisierung von Screens aber schon vollkommen aus.

Immer wenn der Elektronenstrahl nämlich an eine Bildschirmzeile kommt, in der ein Screen zu Ende ist, schaltet er zunächst den Elektronenstrahl ganz aus und beginnt dann zu arbeiten. Er lädt dann die Register von Denise mit neuen Werten, und verändert so z.B. die horizontale und vertikale Bildschirm - auflösung, die Inhalte der Farb-Register und natürlich auch die Adresse, die Denise sagt, wo die Bitmap liegt, aus der sie ein Bild machen soll. Sind diese Umstellungen beendet, schaltet der Copper den Elektronenstrahl wieder an und voilà: ein neuer Screen beginnt.

Um einem Screen völlig andere Eigenschaften zu geben, benötigt der Copper aber mehr Zeit, als der Elektronenstrahl normalerweise braucht, um eine Zeile zu zeichnen. Deshalb bleiben unterhalb eines jeden Screens immer zwei Zeilen leer (oder genauer: sie erscheinen mit der aktuellen Hintergrundfarbe).

14.13.3 Pinsel

Die letzte Anwendung der Grafik-Konzepte des Amiga, die beschrieben werden soll, ist die Definition eigener Pinsel im Programm Deluxe Paint. Mit diesen selbstdefinierten Pinseln sind phantastische Effekte möglich und doch sind sie mit den Möglichkeiten, die die Amiga-Hardware und System-Software bietet, sehr leicht zu realisieren.

Ein selbstdefinierter Pinsel in Deluxe Paint ist zunächst einmal ein rechteckiger Bereich aus einem beliebigen Bild. Das Programm kopiert diesen Teil der Bitmap nun einfach (natürlich mit Hilfe des Blitters) in einen separaten Puffer, ergänzt einige zusätzliche Daten und übergibt der Amiga-System-Software diese Datenstruktur dann als ein BOB.

Wenn Sie nun die Maus bewegen, braucht Deluxe Paint nur eine kleine Routine im Amiga-ROM aufzurufen, und dieses BOB (also der Pinsel) bewegt sich mit.

Drückt man auf die Maustaste, so muß der Pinsel ja malen. Auch dies ist sehr leicht zu erreichen. Wann immer die Maus bei gedrückter Taste bewegt wird, wird dazu einfach eine Kopie des BOBs in die Bitmap gelegt, die das eigentliche Bild enthält. Mit dem Blitter ist dieser Kopiervorgang nur "ein Klacks" und verlangt wieder nur einige wenige Aufrufe innerhalb eines Programms.

Die obigen Beispiele konnten — glaube ich — in eindrucksvoller Weise demonstrieren, daß die Amiga-Ingenieure nicht nur einen leistungsfähigen

Computer gebaut haben, sondern sich auch Mühe gemacht haben, diese Leistung von jedem Programm aus erreichbar zu machen.

15

Klangerzeugung

Nicht ganz so große Publizität wie die Grafikfähigkeiten haben die Möglichkeiten des Amiga zur Klangerzeugung gefunden. Sie sind nichts - destotrotz aber auch recht beeindruckend. Der Amiga besitzt insgesamt 4 unabhängige *Tonkanäle*, von denen jeweils 2 zu einem Stereo-Kanal zusammengefaßt werden können. Jeder der 4 Kanäle kann eine völlig beliebige Wellenform mit beliebiger *Hüllkurve* ausgeben. Er benötigt dazu nur in sehr geringem Maße die CPU bzw. ein Eingreifen des Programms, das den Ton irgendwann einmal "angestoßen" hat.

Um zu verstehen, wie ein Computer aber überhaupt dazu eingesetzt werden kann, Musik (und natürlich auch andere Geräusche zu machen) müssen wir einen kleinen Ausflug in die "digitale Musik" machen.

15.1 Töne, wie sie der Amiga sieht

Ton ist nichts anderes als ständige Luftdruckänderungen. Diese Luftdruck - änderungen bewegen die menschlichen Trommelfelle. Das Nervensystem des Menschen macht aus diesen Bewegungen des Trommelfells dann den Klang, den wir wahrnehmen. Ähnliche Luftdruckschwankungen muß der Amiga also zwangsläufig auch erzeugen, um einen (einigermaßen) natürlichen Klang zu simulieren. Da es aber Verstärker und Lautsprecher gibt, die aus Spannungsschwankungen Luftdruckschwankungen machen können, reduziert sich das Problem auf die Erzeugung einer im "richtigen" Rythmus schwankenden Spannung an den beiden Lautsprecherausgängen des Amiga.

(Mit Spannung kommt ein Computer naturgemäß etwas besser klar als mit Luftdruck.)

Sie können einen Klang nun entweder völlig "frei erfinden" oder einen natürlichen Klang *digitalisieren*. Zum Digitalisieren wird einfach der Luftdruck eines natürlichen Klangs in sehr kurzen Zeitabständen gemessen. Diesen Vorgang des Abtastens nennt man üblicherweise *Sampling*, und die Frequenz, mit der die Lautstärke abgetastet wird, heißt dementsprechend *Sampling-Frequenz* oder *Sampling-Rate*. Es läßt sich zeigen, daß es für die einigermaßen naturgetreue Digitalisierung eines Klangs nötig ist, mindestens eine doppelt so hohe Sampling-Frequenz zu verwenden, wie die Frequenz, die der Klang selbst hat.

Leider kann ein Computer aber Werte nicht beliebig genau messen und abspeichern wie z.B. ein Kassettenrecorder, sondern muß jedem gemessenen Wert eine Zahl zwischen einem ganz bestimmten Minimum und Maximum zuordnen. Deswegen heißt dieser Vorgang auch *digitalisieren*. Aus einer kontinuierlichen (*analogen*) Schwingung wird eine (*digitale*) Folge von diskreten Zahlenwerten. Die folgende Abbildung zeigt in vereinfachter Form ein solches Sampling, bei dem jedem der gemessenen Werte eine Zahl zwischen -128 und +127 zugeordnet wird.

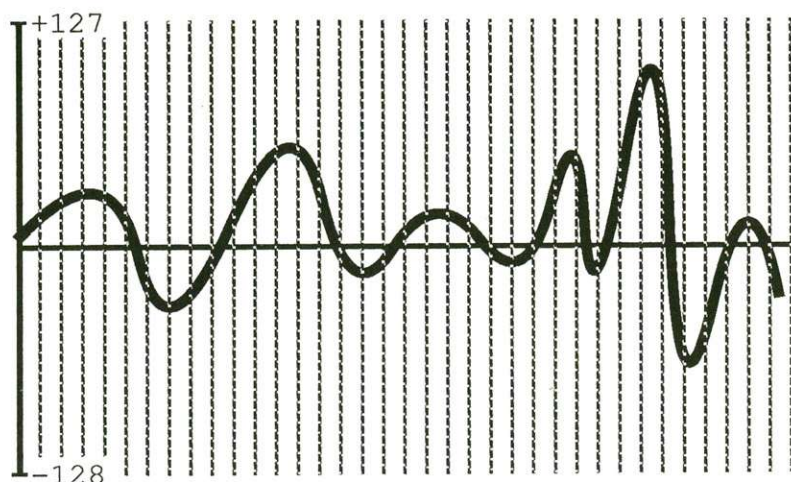


Bild 15 – 1: Ein Klang wird digitalisiert

Ein solches Sampling, wie es Bild 15–1 zeigt, ergibt eine Folge von Werten, die dem Schwanken des Luftdrucks um die Nullachse entsprechen. Die folgende Abbildung zeigt, wie diese Zahlenfolge dann wieder in eine

Spannungs- (und damit Luftdruck-) Schwankung umgesetzt wird, wenn dieser Klang ausgegeben wird.

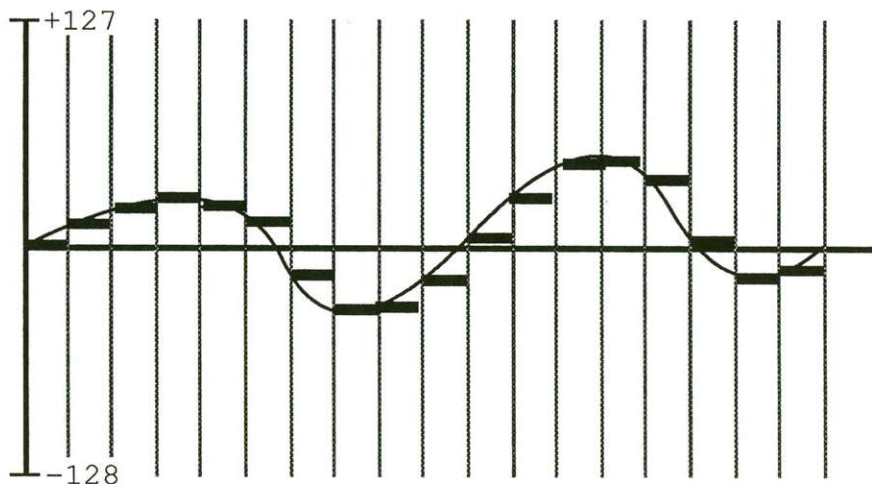


Bild 15 – 2: Ein digitalisierter Klang, wie er ausgegeben wird

Die dicken schwarzen Linien bezeichnen dabei den Verlauf der Spannungsschwankung und die gestrichelte Linie den Verlauf der Luftdruckschwankungen des "Originalklangs" vor der Digitalisierung. Die Abbildung ist allerdings nur eine Annäherung an die tatsächlichen Gegebenheiten. Sie gibt nur das Prinzip wieder. Der Amiga enthält Schaltkreise, die die plötzlichen Sprünge etwas glätten und somit dem echten Klang wieder recht nahe kommen. Trotzdem sieht man hoffentlich, daß auf diese digitale Weise nur eine Annäherung an einen wirklichen Klang erreicht werden kann, der "der Natur" um so näher kommt, je kleiner die Abstände zwischen den einzelnen Messungen sind und je genauer der Wert gemessen wird.

Man darf von der Sound-Hardware des Amiga natürlich nicht zuviel verlangen. Es handelt sich nicht um einen Synthesizer und die Klangqualität kann HiFi-Tests gewiß nicht standhalten. Für die Darstellung eines Klangs werden im Amiga nur 8 Bit breite Werte verwendet. HiFi-Geräte, die ebenfalls die Digital-Technik nutzen (z.B. CD-Plattenspieler) verwenden üblicherweise 16 Bit, was zu einer rund 250-mal besseren Klangqualität führt — von anderen klangverbessernden Konstruktionen in diesen Geräten ganz abgesehen.

Es wäre aber selbst bei dieser verhältnismäßig niedrigen Qualität sehr aufwendig, mehrere Minuten Musik in dieser Form zu digitalisieren. Jeder

Wert in der Sampling-Folge ist ja ein Byte (8 Bit). Bei einer Sampling-Frequenz von nur 10 KHz fallen also jede Sekunde 10 KByte Daten an. In einer Minute also schon 600 KByte oder fast eine volle Diskette. Glücklicherweise ist dies aber für viele Anwendungen gar nicht notwendig. Die meisten Töne — z.B. die musikalischer Instrumente — wiederholen sich nach kurzer Zeit, der sogenannten *Periode*. Man braucht deshalb nur eine Periode zu digitalisieren und kann schon stundenlang ein und denselben Ton "halten". Zudem kann man die Tonhöhe (Frequenz) variieren, in dem die Zahlenfolge einer Periode langsamer oder schneller wiederholt wird. Ohne für jede Frequenz eine Sampling-Folge eines bestimmten Instruments zu haben, kann man es so trotzdem in ganz verschiedenen Tonhöhen wiedergeben.

15.2 Die Hüllkurve eines Tons

Eine einzelne Note hat bei jedem Instrument typischerweise auch einen sehr charakteristischen Lautstärkenverlauf. Zu Beginn steigt sie sehr schnell zu einer Maximal-Lautstärke an, fällt etwas ab, bleibt dort einen Moment auf einem Plateau stehen und klingt dann langsam aus. Diesen Verlauf nennt man Hüllkurve, weil er die eigentlichen Schwingungen des Klangs einhüllt. Die Länge und Steilheit der verschiedenen Plateaus, Steigungen und Gefälle in der Hüllkurve ist von Instrument zu Instrument sehr verschieden und macht viel von dessen typischem Klang aus.

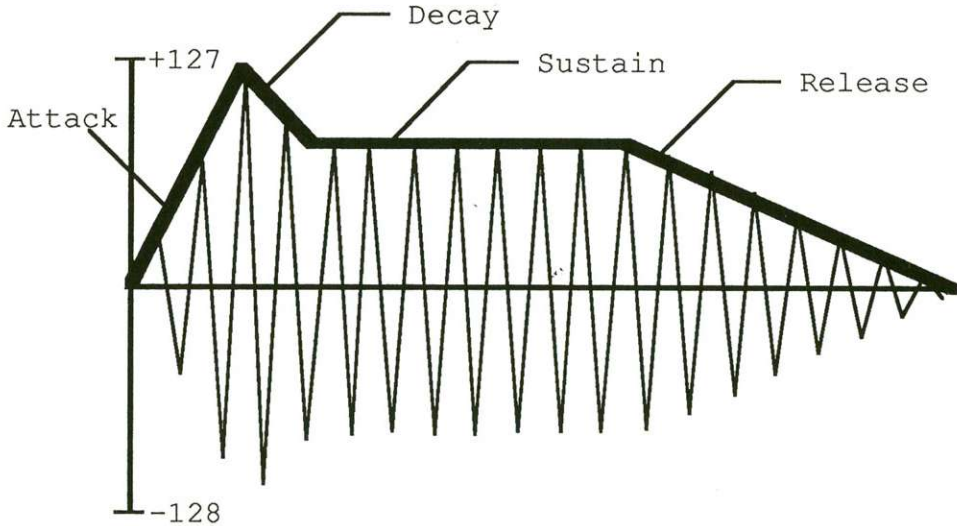


Bild 15 – 3: Die Hüllkurve eines Tons

Die Hüllkurve kann normalerweise durch sehr wenige Zahlenwerte schon genau beschrieben werden und zwar durch drei Zeiten und drei Lautstärken. Dies sind die Zeiträume für das Erreichen der Maximal-Lautstärke, das Abfallen zum ersten "Plateau", die Dauer des Plateaus, die Zeit, bis der Ton völlig verklungen ist und die Lautstärke, die der Ton jeweils am Ende jeder dieser Phasen hat. Der Fachmann nennt diese Zeiten *Attack-* (Anstiegs-), *Decay-* (Abfalls-), *Sustain-* (Halte-) und *Release-* (Auskling-) Zeiten und -Levels. Die Hüllkurve wird nach den Anfangsbuchstaben dieser Namen manchmal auch *ADSR-Kurve* genannt.

15.3 Klangerzeugung mit Paula

Nach dieser mehr theoretischen Vorrede wollen wir uns nun einmal anschauen, wie man Paula wirklich dazu bringt, einen Klang zu erzeugen. Man benötigt dazu vor allem eine digitalisierte Periode des gewünschten Klangs (also eine Zahlenfolge), die im Speicher hinterlegt werden muß. Diese Zahlenfolge kann man entweder bei einem wirklichen Ton messen oder mit Bleistift und Papier erstellen. Die Zahlenfolge für einen mathematisch beschreibbaren Klang, wie z.B. die Sinuswelle, kann man z.B. berechnen. In ein Register von Paula schreibt man danach die Adresse und die Länge der

Zahlenfolge und die Geschwindigkeit, mit der Paula sich neue Werte holen soll. Letztere bestimmt, wie schnell eine Periode ausgegeben wird und damit auch, welche Frequenz der erzeugte Ton hat.

Ist man mit diesen Vorarbeiten fertig, muß man in ein anderes Register einen speziellen Wert schreiben, wodurch Paula beginnt, sich die Zahlenwerte für den Verlauf der Kurve zu holen und so Töne zu erzeugen. Wichtig ist auch hier wieder das DMA-Prinzip. Wenn die Klangerzeugung erst einmal begonnen hat, braucht sich das Hauptprogramm nicht mehr darum zu kümmern. Paula holt sich selbstständig neue Werte aus der digitalisierten Zahlenfolge und wandelt diese in Spannungen am Lautsprecher ausgang um.

Die oben beschriebene Hüllkurve wird aber leider nicht durch die Hardware eingehalten. Hier muß die Systemsoftware helfend eingreifen und während der Erzeugung des Tons die Lautstärke regeln. Dies kostet natürlich ein bißchen der Leistung der CPU — allerdings recht wenig, da diese Lautstärke-Änderungen immer nur nach Zeiträumen anfallen, die einem schnellen Mikroprozessor wie dem 68000 als halbe Ewigkeiten erscheinen.

Leider läßt sich aber nicht jeder Klang durch eine kurze Periode beschreiben, die dann mit einer bestimmten Frequenz von Paula abgespielt werden kann. Viele Geräusche besitzen keine solche Regelmäßigkeit. In diesem Fall muß man entweder die Länge der digitalisierten "Periode" auf die gesamte Dauer des gewünschten Tons ausdehnen oder einen anderen Betriebsmodus von Paula verwenden.

In diesem zweiten Modus macht die Paula fast gar nichts mehr alleine. In diesem Modus muß das Programm die Spannung, die am Lautsprecher - ausgang anliegen soll, und die Dauer, während der sie konstant gehalten werden soll, direkt in ein Register schreiben. Ist die angegebene Zeitdauer vorbei, meldet der Paula-Chip dies dem Programm (durch einen Interrupt) und verlangt nach einem neuen Wert. Das eigentliche Programm wird so immer wieder (und zwar in sehr kurzen Abständen) unterbrochen und muß die Hardware mit neuen Daten versorgen.

Für sehr unregelmäßige Klänge, wie sie nicht bei Musik, aber bei aufgezzeichneten Geräuschen auftauchen, wird die zweite Möglichkeit unter Umständen nötig sein. Im allgemeinen ist die erste Methode der Sounderzeugung aber immer vorzuziehen, da sie weniger Speicherplatz benötigt und auch die CPU in viel geringerem Maße belastet.

Bisher wurde nur beschrieben, wie **ein** Ton erzeugt wird. Paula besitzt aber vier Tonerzeugungs-Kanäle. Für jeden dieser Kanäle gibt es einen Satz der oben beschriebenen Register und jeder Kanal kann so völlig unabhängig von den anderen betrieben werden. Verwendet man allerdings alle vier Kanäle im zweiten Betriebsmodus (s.o.), verlangsamt das alle laufenden Programme

schon recht beachtlich. Die so erzeugten vier Stimmen werden auf zwei Lautsprecherausgänge ausgegeben. Der Programmierer kann selbst wählen, welche der Stimmen auf welchen Lautsprecherausgang geleitet wird. Es müssen aber immer zwei Stimmen auf einem Ausgang liegen; drei auf einen und eine auf den anderen Ausgang zu legen geht nicht.

15.4 Spezialeffekte

Der Amiga erlaubt noch einige spezielle Tricks in Bezug auf die Klangerzeugung. Hierzu können jeweils zwei der Klangkanäle miteinander gekoppelt werden. Der eine (der sogenannte *Oszillator*) erzeugt dann den "eigentlichen" Ton, die vom zweiten Kanal (dem sogenannten *Modulator*) erzeugte Schwingung wird zur Modulation des ersten Tons verwendet. Der Modulator besitzt dabei meist eine wesentlich niedrigere Frequenz als der Oszillator. Je nach der Einstellung der verschiedenen für diese Tricks zuständigen Register kann der Modulator nun die Frequenz des Oszillators oder seine Lautstärke ändern. Damit kann man die seltsamsten Effekte erzielen. Vibratos, Tremolos, Glissandos oder Triller-Effekte können so nahezu ohne jedes weitere Eingreifen eines Programms erreichen.

Eine solche Kopplung zweier Kanäle für diese Spezialeffekte ereignet sich übrigens in genau gleicher Weise auch in professionellen Synthesizern. Nur besitzen diese einige Klangkanäle mehr, so daß die Verwendung eines Kanals zur Modulation eines anderen die restlichen Möglichkeiten (vor allem die Anzahl unabhängiger Musikstimmen) kaum einschränkt. Beim Amiga muß man sich schon sehr genau überlegen, ob man diese Möglichkeiten wirklich braucht. Man hat dadurch schließlich nur noch zwei verschiedene Stimmen für die Klangerzeugung zur Verfügung, auch wenn diese vielleicht nun raffiniertere Effekte beherrschen.

15.5 Sprachsynthese

Eine akustische Fähigkeit des Amiga darf man auf keinen Fall vergessen: die Sprachausgabe. Der Amiga dürfte so ziemlich der erste Computer sein, in dessen standardmäßig vorhandener Systemsoftware Sprachsynthese sofort

mitgeliefert wird. Sie kann in jedem Programm eingesetzt werden und ist sehr problemlos zu programmieren. In der simpelsten Form übergibt man einfach einen ganz normalen Text an eine bestimmte System-Routine, die ihn daraufhin in akustischer Form ausgibt.

Die Aussprache kann man zusätzlich in verschiedenster Form beeinflussen. Die Stimmlage kann geändert werden; der Sprecher kann entweder im tiefen Baß oder als helle Frauenstimme sprechen. Die Geschwindigkeit der Sprache kann innerhalb weiter Grenzen geändert werden (in weiteren Grenzen, als sinnvoll ist). Und schließlich kann die Stimme entweder sehr ausdrucksvoll, mit betonten Worten und Satzzeichen oder monoton, computerhaft sprechen.

So steht es zumindest in der Dokumentation des Amiga. In Wirklichkeit ist die Sache nicht ganz so leicht — zumindest wenn man wirklich verständliche Sprache erzeugen will. Die Umsetzung von geschriebenen Buchstaben in zusammenhängend ausgesprochene Worte und Sätze ist nämlich eine komplexe Aufgabe, die durch viele Ausnahmeregeln erschwert wird.¹ (Der Leser kann ja einmal für sich selbst ausprobieren, wie viele Möglichkeiten es gibt einen Vokal wie *i* auszusprechen, je nachdem, in welcher "Nachbarschaft" er sich befindet). Konsonanten und Wortbetonungen sind — zumindest für einen Computer — noch schwieriger zu behandeln.

Glücklicherweise kann man auch "eine Ebene tiefer" ansetzen. Die Sprachsynthese-Software des Amiga kann nicht nur Klartext aussprechen, sondern auf Wunsch auch "Phoneme". Phoneme kennt jeder, der schon einmal in ein Wörterbuch geschaut hat. Es sind die Zeichen, mit denen die Aussprache von Worten beschrieben wird. Phoneme sind die "Urbausteine" der Sprache, elementare Laute, aus denen man jedes Wort (fast) jeder Sprache zusammensetzen kann.

Mit etwas Mühe — d.h. bei manchen Worten muß man die Phonem-schreibweise im Wörterbuch nachschauen — kann man den Amiga auch dazu bringen, deutsche Sätze auszusprechen. Ein unverkennbarer (amerikanischer) Computer-Akzent bleibt allerdings. Trotzdem ist die Sprachausgabe für viele Anwendungszwecke — z.B. Fehlermeldungen — bestimmt recht brauchbar. Auch ohne daß man gerade auf den Bildschirm schaut, kann einem ein Programm auf diese Weise mehr Informationen mitteilen als mit einem kurzen Piepston, der sonst für solche Zwecke verwendet wird.

¹ Die augenblickliche Systemsoftware des Amiga ist zudem nur fähig, englische Text auszusprechen. Versucht man deutsche Sätze mit dieser Software auszusprechen, so ergibt dies ein zwar interessantes, aber ziemlich unverständliches Kauderwelsch.

15.6 MIDI

Wie es inzwischen scheinbar schon guter Ton bei neuen Computern zu werden scheint, ist der Amiga natürlich auch für die Kommunikation mit MIDI-Geräten (MIDI = "Musical Instruments Digital Interface"); der wohl wichtigste Standard für die Verbindung digitaler Musikinstrumente, wie Synthesizer und Keyboards) vorbereitet. Hierzu benötigt man ein kleines Kästchen namens *MIDI-Interface*, das an den seriellen Ausgang des Amiga angeschlossen wird und die Signale des Amiga dem MIDI-Standard anpaßt. Dann braucht man "nur noch" einen guten Synthesizer und ein MIDI-fähiges Keyboard.

Wem also die Klangqualität des Amiga auch dann noch nicht ausreicht, wenn er ihn an seine Stereoanlage angeschlossen hat, der sollte sich die Anschaffung des MIDI-Interfaces und der entsprechenden MIDI-Geräte überlegen.

Man mag sich nun vielleicht fragen, wozu der Computer überhaupt noch gut ist, wenn die Musik auf einer separaten Tastatur eingegeben und auf einem Synthesizer erzeugt wird. Die Antwort darauf ist einmal mehr bei der Software zu finden. Wie man vielleicht schon bei der Beschreibung des Programms Musicraft (im Kapitel Arbeiten mit dem Amiga) gesehen hat, ist es wesentlich komfortabler, Musik auf einem Computer zu entwickeln als auf einem Blatt Papier. Dies wird noch deutlicher, wenn Musik mit noch mehr als nur vier Stimmen erzeugt werden soll und die Bedienung eines leistungsfähigen (und komplizierten) Synthesizers durch den Musiker nötig wird.

Man könnte den Amiga deshalb sehr sinnvoll als Oberaufsicht in einem System MIDI-fähiger Komponenten einsetzen, auf dessen Bildschirm man einen Überblick über die verschiedenen Vorgänge und Verbindungen gezeigt bekommt, mit dem man aber gleichzeitig komponiert und neue Klänge entwickelt.

16

Die Programmierung des Amiga

In den vorangegangenen Kapiteln haben wir ja nun einen ungefähren Eindruck davon bekommen, welche Möglichkeiten die Hardware des Amiga bietet. Falls der eine oder andere unter den Lesern dadurch Appetit bekommen hat, auch einmal selbst ein wenig mit dieser Möglichkeiten "herumzuspielen", kann ihm vielleicht dieses und das nächste Kapitel helfen. Es soll nämlich nun um die Programmierung des Amiga gehen. Hierzu werden in diesem Kapitel zunächst etwas allgemeiner die Möglichkeiten zur Programmierung erläutert und im nächsten Kapitel dann beschrieben, was die Programmiersprache zu bieten hat, die jedem Amiga beiliegt: Amiga-Basic.

Die Programmierung des Amiga spielt sich nie "im leeren Raum" ab. Alle Amiga-Programme nutzen intensiv die System-Software, die sich auf der Kickstart-Diskette und der Workbench-Diskette befindet. Die "Kunst der Amiga-Programmierung" besteht unter anderem auch darin, diese "Bibliotheken" von Funktionen, die einem viel Arbeit abnehmen können, richtig zu nutzen.

16.1 Architektur der Amiga-Software

Im Zusammenhang mit Software von *Architektur* zu reden, mag vielleicht etwas hochtrabend klingen. In Wirklichkeit ist die Entstehung eines so komplexen Systems, wie es die Systemsoftware des Amiga darstellt, ein mindestens genauso komplexer Vorgang wie der Bau eines Hauses. Sie muß sorgfältig geplant werden, und verlangt ein reibungsloses Zusammenspiel

vieler Beteiligten. Auch die Teile des entstehenden Gebäudes müssen nahtlos zusammenpassen und sich gegenseitig in genau berechneter Weise stützen.

Eine der größten Stärken des Amiga liegt in diesem reibungslosen Zusammenspiel der einzelnen Komponenten der Systemsoftware mit anderen Software-Komponenten und mit den Hardware-Komponenten. So arbeiteten z.B. von Anfang an die beiden Teams, die die Hardware und die Software entwickelten, eng zusammen und beeinflussten sich immer wieder gegenseitig. Dies beschleunigte nicht gerade die Entwicklung, garantiert jetzt aber ein sehr effizientes System mit ideal aufeinander abgestimmten Bestandteilen.

Die "Kunst der Amiga-Programmierung" besteht zu einem großen Teil aus der geschickten Nutzung der Systemsoftware. Diese stellt eine Art "Software-Bibliothek" dar, in der Programme sich die Funktionen aussuchen können, die sie benötigen — und somit nicht noch einmal selbst realisieren brauchen.

16.1.1 Exec

Die wichtigsten Komponenten der Hardware haben wir ja in den vergangenen Kapiteln kennengelernt. Der wichtigste Bestandteil der Amiga-Systemsoftware ist **Exec**. Exec ist ein Paket von kleinen, sehr effizienten Routinen im sogenannten ROM-Kern des Amiga (also in der Software, die im Moment von der Kickstart-Diskette eingelesen wird, später aber vielleicht irgendwann einmal in einen neuen ROM kommt). Diese Routinen werden von allen anderen Komponenten der Software andauernd benötigt. Sie werden für eine Vielzahl von sehr elementaren Aufgaben bei der Verwaltung der verschiedenen "System-Ressourcen" des Amiga benötigt.

16.1.1.1 Listen

Exec bietet z.B. Hilfsroutinen für die Verwaltung von Listen an. Solche Listen werden von fast jedem Programm und auch von anderen Teilen Execs andauernd benötigt. In ihnen werden Namen, Dateien, Programme, Fenster, usw. gesammelt und verwaltet.

Da die Listenverwaltung sich innerhalb von Exec befindet, ist sie sehr schnell (in Maschinencode geschrieben) und steht jedem zur Verfügung, der sie benötigt. (Exec befindet sich ja immer im Speicher.) Andere einzelne Programme brauchen so nicht eigene Pakete zur Listenverwaltung zu beinhalten und können dadurch kompakter werden.

16.1.1.2 Tasks

Eine besonders wichtige Sorte von "Objekten", die Exec in Listen verwaltet, sind die Tasks, die verschiedenen Programme im Amiga, die scheinbar alle gleichzeitig laufen. Exec erzeugt und verwaltet solche Tasks, teilt ihnen Speicher und CPU-Zeit zu und zerstört sie auch wieder.

Die Verteilung der CPU-Zeit (und anderer Ressourcen) auf die verschiedenen darum konkurrierenden Tasks ist eine der Hauptaufgaben von Exec. Auch hierbei zählt vor allem die Geschwindigkeit. Besonders der Wechsel von einem Task zu einem anderen muß sehr schnell vor sich gehen, da solche Wechsel jede Sekunde zig-mal vorkommen können.

16.1.1.3 Nachrichten

Ebenfalls zur Task-Verwaltung im weiteren Sinn gehört der Nachrichten-Mechanismus von Exec. Er dient zu einer reibungslosen Kommunikation zwischen unterschiedlichen Tasks. Oft ist es nämlich nötig, daß verschiedene Tasks miteinander kommunizieren müssen, um z.B. Daten auszutauschen. Dies ist nicht ganz so einfach, da kein Task vom anderen weiß, in welchem Zustand sich dieser gerade befindet. Ein Kommunikationsversuch zum falschen Zeitpunkt kann aber dazu führen, daß Daten verloren gehen oder den empfangenden Task bei einer wichtigen Aufgabe stören.

Der Nachrichten-Mechanismus von Exec umgeht diese Schwierigkeiten und sorgt dafür, daß Tasks nur dann Meldungen empfangen, wenn sie dazu bereit sind. Exec weiß natürlich als einziger immer ganz genau, ob ein Task gerade bereit ist, eine Nachricht zu empfangen oder nicht, da es die Tasks ja kontrolliert. Kann ein Task eine Nachricht nicht sofort empfangen, legt Exec sie erst einmal in eine Warteschlange (eine Liste!), wo sie der Task entnehmen kann, sobald er Zeit hat. Keine Nachricht geht bei diesem Mechanismus verloren.

16.1.1.4 Speicherverwaltung

Exec kümmert sich zusätzlich auch noch um die Speicherverwaltung im Amiga. Es ist naheliegend, einem übergeordneten Paket wie Exec diese Aufgabe zu überlassen. Der Speicher ist schließlich eine sehr "globale" Ressource des Amiga, die von allen Programmen (Tasks) benötigt wird. Würden alle Tasks gleichzeitig und unkontrolliert im Speicher herumfuhrwerken, wäre Chaos unvermeidlich.

Wann immer ein Programm zusätzlichen Speicherplatz benötigt (z.B., weil ein Benutzer eine neue Zeile in der Textverarbeitung eingegeben hat), kann es diesen bei Exec anfordern. Exec sucht dann nach einem Stück im Speicher, das die passende Größe hat und noch frei ist, reicht dieses dem anfragenden Task und markiert es dann als belegt. Währenddessen werden alle anderen Programme angehalten, damit sie nicht zur selben Zeit den selben Speicher anfordern können.

"Brave" Programme geben einen so angeforderten Speicherblock natürlich auch sofort, wenn sie ihn nicht mehr benötigen, an Exec zurück, damit er anderen Programmen zugeteilt werden kann, die ihn vielleicht dringend brauchen.

16.1.1.5 Bibliotheken

Die wichtigste Aufgabe von Exec ist neben den Tasks aber wahrscheinlich die Verwaltung von Programmbibliotheken. Diese können in ähnlicher Weise von vielen Programmen gleichzeitig genutzt werden wie Exec selbst. Sie brauchen sich aber nicht in der Kickstart-Software zu befinden. Bibliotheken sind nichts anderes als besondere Dateien, die die Routinen einer Programmsammlung (Programm-Bibliothek) enthalten. Solche Dateien können jederzeit auf die Workbench-Diskette kopiert und danach von allen Programmen genutzt werden.

Exec sorgt dafür, daß Programme, die eine Bibliothek nutzen wollen, nicht zu wissen brauchen, wie diese Datei aussieht, wo sie liegt und ob sie bereits von einem anderen Programm benutzt wird oder neu in den Speicher geladen werden muß. Ein Programm braucht eine Bibliothek nur mit einem entsprechenden Exec-Aufruf und dem Namen der Bibliothek zu "öffnen" und kann danach die Routinen der Bibliothek benutzen, als wären es Bestandteile des Programms selbst.

16.1.1.6 Geräte

Bei der Beschreibung des CLIs haben wir ja gesehen, daß die verschiedenen Ein- und Ausgabegeräte, die man am Amiga anschließen kann, behandelt werden können wie Dateien. Für diese Einheitlichkeit der Behandlung auch unterschiedlichster Geräte sorgt ebenfalls Exec.

Für jedes angeschlossene Gerät richtet Exec nämlich zusammen mit dem sogenannten *Treiber* des Geräts einen separaten Task ein. Um Daten zu einem Gerät zu schicken oder welche von diesem zu holen, schickt Exec dem entsprechenden Task eine Nachricht, deren Format unabhängig von dem

Gerät selbst ist. Diese Nachricht enthält nur die Art der Anforderung an das Gerät (also z.B., ob Lesen oder Schreiben eines Datenblocks gewünscht wird). Welche Disketten nun gedreht oder welche Schalter umgelegt werden müssen, um diese Anforderung zu erfüllen, ist Exec egal. Der Geräte-Task ist alleine zuständig für die Beantwortung der Nachricht und somit für die exakte Ansteuerung der Hardware zu diesem Zweck.

Wenn ein neues Gerät an den Amiga angeschlossen wird, benötigt Exec nur einen passenden Treiber dazu. Dieser besteht einfach aus einer Datei, die der Geräte-Hersteller üblicherweise mitliefert und die einfach auf die Workbench-Diskette gelegt wird. Und schon können auch Programme Daten auf einem Gerät ausgeben, bei deren Entstehung es dieses Gerät vielleicht noch gar nicht gab. (In dieser Hinsicht sind Treiber durchaus ähnlich wie Bibliotheken zu verwenden.)

16.1.2 Die Grafik-Bibliothek

Eine dieser von Exec verwalteten Bibliotheken, die von fast allen Programmen genutzt wird, ist die Grafik-Bibliothek (engl. *Graphics Support Library*). Mit ihr hat direkt oder indirekt jedes Programm etwas zu tun, das irgendwelche Ausgaben auf dem Bildschirm zeigen will — egal ob Text oder Grafik.

Die Grafik-Bibliothek dient vor allem dazu, die Komplexität der Amiga-Grafik-Hardware vor dem Programm zu verbergen. Denn obwohl die Fähigkeiten der Grafik-Hardware beachtlich sind, verlangen sie doch auch die Erledigung einiger Vorarbeiten und Haushaltungsaufgaben, bevor auch nur ein kurzer Strich auf dem Bildschirm erscheint.

Durch die Routinen der Grafik-Bibliothek sieht der Bildschirm auch für ein Programm so simpel aus wie eine Malfläche, über der ein Pinsel oder Stift bewegt wird. Nur die Größe dieser Fläche und die Anzahl der Farben in der Palette müssen vor der ersten Zeichenoperation festgelegt werden. Um danach z.B. einen Strich zu zeichnen, wählt man zunächst eine Farbe aus. Der Zeichenstift kann dann durch Aufruf der entsprechenden Routinen zu jedem beliebigen Punkt der Fläche bewegt werden. Bei jeder solchen Bewegung des Stiftes kann man wählen, ob der Stift über der Zeichenfläche "schwebt" und somit nur bewegt wird, oder ob er "abgesenkt" wird und Spuren — also eine Linie — hinterläßt.

Andere Grafik-Routinen sind fähig, Rechtecke zu zeichnen und beliebige Flächen, die durch Punkte einer anderen Farbe begrenzt sein müssen, mit einer beliebigen neuen Farbe zu füllen. Letzteres funktioniert wie der

Farbeimer in Graphicraft (und ist auch genauso empfindlich gegenüber "Lecks" in der Umrandung).

Zuletzt ist die Grafik-Bibliothek auch noch für das "Malen" von Text auf dem Bildschirm zuständig. Dies ist bei bitmapped Grafik ja nicht so einfach wie bei Computern, die einen reinen Text-Modus besitzen. Erschwerend kommt noch hinzu, daß der Amiga viele verschiedene Zeichensätze, Text-Stile und -Größen kennt. Bei Verwendung der entsprechenden Routinen aus der Grafik-Bibliothek ist ein Text aber so einfach wie ein Strich zu zeichnen. Zunächst wird der gewünschte Zeichensatz mit seinem Namen ausgewählt und die Größe der Buchstaben sowie der Textstil (fett, unterstrichen, kursiv oder eine Kombination davon) festgelegt. Dann wird der Stift zu der Position bewegt, wo der Text beginnen soll und eine Routine der Grafik-Bibliothek aufgerufen, der der Text übergeben wird. Diese zeichnet ihn dann selbständig auf den Bildschirm, wobei für die Buchstaben die aktuelle Farbe des Zeichenstiftes (s.o.) verwendet wird.

16.1.3 Intuition

Auch Intuition ist eine Bibliothek. Was Intuition so alles leistet, ist ja — allerdings bei weitem nicht vollständig — bereits im Kapitel mit dem treffenden Namen *Intuition* aufgelistet worden. Intuition ist der Teil der Systemsoftware, mit dem ein Programm wahrscheinlich am meisten kommuniziert. Exec und die Grafik-Bibliothek werden zwar andauernd benötigt, ein Programm merkt davon aber im allgemeinen recht wenig und hat auch nicht direkt mit Exec zu tun.

Intuition enthält z.B. Routinen zum Öffnen und Schließen von Screens, Fenstern und Requestern. Auch alle Gadgets und Menüs werden von Intuition gezeichnet und bearbeitet. Die Routinen in Intuition achten auch darauf, daß alle Ausgaben eines Programms in dem ihm zugeteilten Fenster oder Screen bleiben. Eine lange Linie, die von einem Programm gezeichnet wird, stoppt Intuition z.B. automatisch am Fensterrand. Durch Intuition wird so die (pseudo-)parallele Arbeit mit mehreren gleichzeitig aktiven Programmen erst möglich. Intuition sorgt dafür, daß sich die Ein- und Ausgaben gleichzeitig laufender Programme in sauberlich begrenzten Gebieten auf dem Bildschirm abspielen und sich deshalb nicht gegenseitig stören.

Intuition nimmt einem Programm zusätzlich auch noch sehr viel Arbeit bei der Kommunikation mit dem Anwender des Programms ab. Alle Aktionen des Anwenders — vor allem Mausklicks und Tastendrucke — laufen nämlich

zunächst einmal durch Intuition, wo sie einer Vorverarbeitung unterzogen werden. Mausklicks in ein Gadget, die Titel-Leiste eines Screens oder Fensters, usw. bearbeitet Intuition zunächst einmal selbst. Vom Bewegen und Verkleinern eines Fensters, vom Herunterklappen eines Menüs merkt ein Programm herzlich wenig. Und dies ist auch gut so, weil die damit verbundenen Operationen recht komplex sind (obwohl alles so mühelos aussieht) und das Programm nur von der Erledigung seiner eigentlichen Aufgaben abhalten würden.

Erst, wenn eine Reaktion des Programms nötig wird — z.B. wenn der Anwender einen Menü-Befehl ausgewählt hat — wird die entsprechende Aktion in Form einer Exec-Nachricht verpackt und an den Programm-Task des gerade aktiven Fensters geschickt. Diese Nachricht enthält z.B. die Nummern des ausgewählten Menüs und des Befehls darin oder den Buchstaben, der einer soeben gedrückten Taste entspricht. Was ein Programm mit diesen Nachricht macht, ist dann seine Sache.

16.2 Grundlagen der Programmierung

Zur Programmierung des Amiga benötigt man, wie für jeden anderen Computer auch, eine Programmiersprache. In Wirklichkeit versteht der Amiga allerdings nur eine (Programmier-) Sprache, den sogenannten *Maschinen-Code*. Dies ist eine Folge von Bits, die zu "Worten" von jeweils 16 Bit zusammengefaßt werden. Da ein Mensch normalerweise nicht in Bit-Folgen denkt und redet, spielt sich die Kommunikation mit dem Computer zwangsläufig etwas schleppend ab, wenn nicht zusätzliche Hilfsmittel eingesetzt werden.

16.2.1 Sprachen

Ein solches Hilfsmittel stellen die sogenannten *höheren Programmiersprachen* dar — *höher* deshalb, weil man sich nicht mehr auf das "niedere" Niveau der Maschine hinabgeben muß. Die wesentliche Idee bei allen höheren Programmiersprachen ist es, ein Programm zunächst einmal in einer einigermaßen lesbaren Sprache zu formulieren, die aus "richtigen" Worten besteht, welche aus "richtigen" Buchstaben gebildet werden. Wie nahe die so entstehende Sprache der menschlichen Sprache kommt, ist verschieden. Es gibt sehr primitive Programmiersprachen, die wohl nur von Fachleuten

verstanden werden und sehr "hohe" Programmiersprachen, in denen sich Programme fast wie Gebrauchsanweisungen aus dem vielzitierten "richtigen Leben" lesen.

Falls Sie bisher noch keine Programmiersprache kennen, lassen Sie sich also nicht vom Getue der Programmierer abschrecken, eine zu erlernen. Alle Programmiersprachen sind wesentlich simpler und leichter zu erlernen als auch die einfachste menschliche Sprache.

Was alle Programmiersprachen bei all ihren Unterschieden gemeinsam haben, ist, daß immer ein Verfahren definiert ist, wie man von einer Anweisung in der höheren Programmiersprache zur *äquivalenten* Bit-Folge im Maschinen-Code kommt. *Äquivalent* heißt in diesem Fall, daß die Bit-Folge das tut, was der Programmierer gemeint hat, als er das Programm in der höheren Sprache aufgeschrieben hat. (Eines der Hauptprobleme bei der Programmierung ist, daß sich Programmierer und Computer meist nicht so ganz darüber einig werden können, was der Programmierer denn nun eigentlich "gemeint" hat!)

Das Verfahren, das ein Programm in Maschinen-Code umwandelt, ist relativ (!) simpel und geradlinig. Es ist zumindest so simpel, daß man es wiederum einem Computer-Programm überlassen kann. Es gibt zwei große Gruppen solcher Programme, die zwei grundsätzliche Strategien repräsentieren, mit denen man diesen Übersetzungsprozeß angehen kann: *Interpreter* und *Compiler*.

Eine Ursache für viel Verwirrung bei Programmieranfängern ist oft, daß diese *Programme*, die andere Programme in Maschinen-Code umsetzen, ebenfalls *Programmiersprache* genannt werden. Dies ist eigentlich falsch! Genau wie ein Dolmetscher für Französisch nicht die *Sprache Französisch* ist, ist auch ein Interpreter für Basic nicht die *Sprache Basic*. Behalten Sie dies immer im Auge, wenn Ihnen jemand das nächste Mal erzählen will, diese oder jene Computer-Sprache sei gut oder schlecht. Meist will er damit in Wirklichkeit sagen, das der Compiler oder Interpreter gut oder schlecht ist.

16.2.2 Interpreter

Ein *Interpreter* ist die einfachste Möglichkeit, eine Umsetzung von höherer Programmiersprache zum Maschinencode zu realisieren. Er enthält üblicherweise für jede Anweisung (jedes Wort) der höheren Programmiersprache ein bestimmtes Stück Maschinencode, das dieser Anweisung entspricht (ihr *äquivalent* ist). Um ein Programm auszuführen, liest ein Interpreter es Zeile für Zeile, sucht die entsprechenden Maschinencode-Stückchen für die einzelnen Anweisungen heraus und übergibt diese dem

Computer zur Ausführung. Diesen Vorgang nennt man dann sinnigerweise auch *Interpretieren* eines Programms.

Wen man einen Vergleich mit menschlichen Sprachen ziehen würde, könnte man sagen, daß ein Interpreter einem Dolmetscher entspricht, der ein fremdsprachiges Buch vorliest und dabei simultan übersetzt.

16.2.3 Compiler

Ein *Compiler* hingegen nimmt den Programmtext (in der höheren Programmiersprache) und erzeugt daraus ein äquivalentes Programm in Maschinencode. Der gesamte Übersetzungs-Vorgang heißt *Compilieren* oder auch einfach *Übersetzen*. Dabei wendet er raffinierte Techniken an, um den Maschinencode möglichst klein und schnell zu machen. Im allgemeinen produzieren diese Techniken nicht aus einer Programmanweisung immer ein und dasselbe Stückchen Maschinencode, sondern verwenden Wissen über die vorangehenden und nachfolgenden Anweisungen, um den erzeugten Code zu optimieren.

Wollte man wieder den Vergleich mit dem fremdsprachigen Buch bemühen, könnte man sagen, daß ein Compiler einem Dolmetscher entspricht, der eine schriftliche Übersetzung des Buches anfertigt, die man danach selbst lesen kann, ohne dazu den Dolmetscher selbst noch zu benötigen.

16.2.4 Vergleich von Compiler und Interpreter

Die Vorteile des Compiler-Prinzips liegen auf der Hand. Der erzeugte Maschinencode ist selbständig und benötigt kein zusätzliches Programm (den Interpreter) zur Ausführung. Zudem sind compilierte Programme üblicherweise immer schneller, da kein Interpreter mühselig Zeile für Zeile den Programmtext durchlaufen, analysieren und interpretieren muß.

Compiler haben aber auch gravierende Nachteile, weshalb sie sich bisher trotz ihrer Vorteile noch nicht generell für alle Sprachen durchgesetzt haben. Der Vorgang des Compilierens eines Programms dauert z.B. oft sehr lange. Das dabei entstehende Programm ist zwar schnell, dafür muß man zunächst aber erst einmal eine Wartezeit in Kauf nehmen. Dies ist besonders während der Programmentwicklung sehr lästig. Nach jeder kleinen Änderung muß das Programm neu übersetzt werden und man muß oft mehrere Minuten warten, bis die Änderungen am laufenden Programm getestet werden können. Bei

Verwendung eines Interpreters ist ein Programm auch nach einer Änderung sofort wieder lauffähig.

Compiler sind zudem meist schwieriger zu bedienen als Interpreter. Während es bei diesen in der Regel ausreicht, einen Start-Befehl (z.B. RUN) zu geben, um ein Programm mit der Arbeit beginnen zu lassen, verlangen Compiler mehrere Arbeitsschritte bis zu diesem Punkt (bei denen man natürlich auch jedesmal Fehler machen kann). Interpreter sind deshalb auch heute noch unschlagbar, wenn es darum geht, eine Programmiersprache zu erlernen oder ein Programm wirklich **interaktiv** zu entwickeln.

16.3 Programmiersprachen für den Amiga

Im Gegensatz zu vielen anderen neuen Computern gab es für den Amiga schon kurze Zeit nach der Markteinführung eine Vielzahl von Compilern und Interpretern. Eigentlich werden alle weitverbreiteten Sprachen bereits unterstützt. Kein Programmierer wird also durch fehlende Werkzeuge davon abgehalten, sich mit den phantastischen Möglichkeiten des Amiga näher zu beschäftigen.

16.3.1 Assembler und Maschinencode

Die Programmiersprache, die dem Maschinencode noch am nächsten kommt, ist die **Assembler-Sprache**. Bei einem Assembler entspricht jede Zeile des Programms ziemlich genau einer Instruktion (einem Wort) im Maschinencode. Der Compiler (in diesem Fall ebenfalls Assembler genannt) hat es also mit dieser Sprache recht leicht.

Mit einem guten Assembler kann man wirklich alle Möglichkeiten nutzen, die ein Computer zu bieten hat, muß sich dabei aber auf ein recht niedriges Niveau der Programmierung begeben. Selbst einfache Dinge — wie das Ausgeben eines kurzen Textes auf dem Bildschirm — erfordern in Assembler-Sprache oft nicht-triviale, größere Programme.

Ein weiteres Problem der Assembler-Sprache ist, daß es sehr leicht ist, beliebig viele logische Fehler in ein Programm einzubauen; der Assembler merkt nichts davon. Höhere Programmiersprachen schützen in stärkerem

Maße vor den eigenen Fehlern und nehmen damit die lästige Arbeit der Fehlersuche zumindest zum Teil ab.

Nur wenige Programme werden heute noch in Assembler-Sprache entwickelt, da es inzwischen andere Programmiersprachen gibt, mit denen die meisten Programmieraufgaben genauso gut erledigt werden können wie in Assembler-Sprache. Nur wenn Geschwindigkeit und Kompaktheit eine extrem große Rolle spielen (oder der Programmierer einfach Spaß am "Bit-Popeln" hat) greift man heute noch auf Assembler-Sprache zurück.

Ein Assembler gehört zum Standard-Lieferumfang des *Entwickler-Paketes*, das Software-Entwickler bei Commodore erwerben können. Beim Amiga für "normale Sterbliche" liegt kein Assembler bei. Er kann aber jederzeit mit den dazugehörigen Unterlagen auch separat erworben werden. Bei diesem Assembler handelt es sich ansonsten um einen recht komfortablen Vertreter seiner Gattung (für Fachleute: es ist ein vollausgebauter Macro-Assembler).

16.3.2 Basic

Basic dürfte heute die am weitesten verbreitete Computersprache auf der ganzen Welt sein. Sie gehört standardmäßig zum Lieferumfang vieler Computer, so auch des Amiga, oder wird sogar gleich in die Computer eingebaut (in Form eines ROMs). Über Amiga-Basic erfahren Sie aber mehr im folgenden Kapitel, das sich der Sprache und dem zugehörigen Interpreter ausführlich widmet.

16.3.3 C

Eine Sprache, die sich in den letzten Jahren gerade auf Microcomputern wachsender Beliebtheit erfreut, ist auch **C**. Sie stellt einen — nach Meinung vieler Fachleute sehr gelungenen — Kompromiß zwischen Maschinennähe (à la Assembler) und Programmierkomfort dar. C wird fast nur im Zusammenhang mit Compilern verwendet und ist dementsprechend auch nicht gerade eine Sprache für Anfänger.

Ein zusätzlicher Nachteil von C aus der Sicht eines Anfängers ist, daß es, ähnlich wie beim Assembler, wenig Schutz vor den eigenen Fehlern gibt. Man kann viele Fehler in ein Programm einbauen, die so "dumm" sind, daß sie ein derart aufwendiges Programm (wie es ein guter C-Compiler darstellt) eigentlich finden könnte — nur: er sucht gar nicht danach!

Dafür erzeugen C-Compiler aber meist sehr "guten" Maschinencode, der recht nahe an die Geschwindigkeit und die Kompaktheit herankommt, die man erreichen könnte, wenn man dasselbe Programm mit einem Assembler erstellen würde — was ungleich schwieriger und lanwieriger wäre. Zudem sind die C-Compiler vieler Hersteller so raffiniert konstruiert, daß man sie sehr leicht an einen neuen Computer anpassen kann. Dies gilt selbst dann, wenn dieser einen ganz anderen Maschinencode versteht. Es gibt deshalb für neue Computer meist sehr schnell C-Compiler, während Compiler und Interpreter für alle anderen Sprachen oft lange auf sich warten lassen.

Genau dies war auch beim Amiga der Fall. Das Amiga-Entwickler-Paket (für Software-Entwickler), das es schon gab, als der Amiga noch nicht einmal offiziell verkauft wurde, enthielt von Anfang an einen C-Compiler. Es handelt sich dabei um einen Compiler der US-Firma *Lattice*, die auch auf anderen Computern bekannt für ihre guten und leicht übertragbaren C-Compiler ist. (Der Lattice-C-Compiler kann aber auch außerhalb des Entwickler-Pakets erworben werden, falls man sich zunächst den Amiga gekauft hat und erst später das Interesse an der Sprache C hinzu gekommen ist.)

Der Lattice-C-Compiler ist allerdings nicht uneingeschränkt für jeden Anwender zu empfehlen, da er verhältnismäßig viel Platz auf der Diskette benötigt und sehr langsam ist. Der zügige Umgang mit ihm verlangt einige Tricks und Kniffe, die man erst herausfinden muß, wofür man aber wiederum einige Vorkenntnisse über die Arbeitsweise des Amiga-Betriebssystems und des C-Compilers besitzen muß.

Wesentlich kompakter, schneller und zudem noch leichter zu bedienen ist der *Aztec-C-Compiler* von der US-Firma *Manx Software Systems*. Er läßt sich z.B. problemlos auch mit nur einem Diskettenlaufwerk betreiben, was beim Lattice C nicht der Fall ist (Lattice C braucht eigentlich sogar ein drittes Diskettenlaufwerk). Auch die Firma Manx bietet ähnliche C-Compiler bereits mehrere Jahre lang für andere Computer an; hat also ausreichend Erfahrung, um ein ausgereiftes Produkt bieten zu können. Der Aztec-C-Compiler kann — mit gewissen Einschränkungen — auch für C-Anfänger empfohlen werden.

16.3.4 Lisp und Logo

Eine für Ausbildungszwecke geradezu berühmte Sprache ist **Logo**. Sie wurde speziell für die Ausbildung von Kindern (!) in der Programmierung geschaffen. Entsprechend diesem Anwendungsgebiet gibt es auch eigentlich kaum Logo-Compiler, sondern fast nur Logo-Interpreter.

Logo ist eine Sprache, die mit sehr wenigen Worten auskommt und deshalb sehr schnell zu erlernen ist. Trotzdem lassen sich — entgegen anders lautenden Gerüchten — auch große und anspruchsvolle Programme damit entwickeln. Logo ist also **keine** Sprache nur für Kinder! Was sie aber für Kinder und andere Programmier-Anfänger so attraktiv macht, ist, daß sie sehr schnell erlernbar ist und man sehr schnell eigene Programme entwickeln kann. Die ersten Erfolgserlebnisse beim Programmieren kommen sehr schnell.

Logo ist zudem sehr grafisch orientiert und kann auch sehr gut dazu verwendet werden, Programmstrukturen durch entsprechende grafische Strukturen darzustellen. Dies paßt natürlich hervorragend zur stark grafisch/visuellen Orientierung des Amiga.

Im Moment ist leider jedoch noch kein funktionsfähiges Logo-System für den Amiga auf dem Markt. Mindestens drei Hersteller haben aber Logo-Interpreter für den Amiga angekündigt, so daß diese traurige Situation bestimmt nicht mehr allzu lange so bleiben wird.

Logo hat allerdings viel mit der Sprache **Lisp** gemeinsam. Lisp wird vor allem auf dem Gebiet der *künstlichen Intelligenz* oder allgemeiner gesagt für die *symbolische Datenverarbeitung* eingesetzt. Mit Lisp und Logo ist es wesentlich einfacher, Sätze, Worte und Bilder zu analysieren und zu verarbeiten, als z.B. mit C oder Pascal.

Das bekannte *Cambridge-Lisp* wird für den Amiga bereits von dem britischen Software-Haus *Metacomco*, das auch einen Pascal-Compiler für den Amiga anbietet (s.u.), vertrieben. Im Gegensatz zu den meisten anderen Lisp-Realisierungen auf Microcomputern ist Cambridge-Lisp eine recht vollständige Realisierung der Sprache. Der relativ beschränkte Speicher des Amiga setzt der Arbeit mit Lisp aber natürliche Grenzen, weshalb man von diesem System nicht allzu viel verlangen sollte. (Computer, auf denen professionelle Lisp-Systeme laufen, haben üblicherweise mindestens ein bis zwei **Mega-Byte** mehr Speicher als der Amiga.) Für das Erlernen von Lisp stellt der Amiga aber eine völlig ausreichende Umgebung dar.

16.3.5 Pascal und Modula

Ebenfalls für die Ausbildung — in diesem Fall aber die Ausbildung von Informatikern an der Universität (genauer ETH) Zürich — wurde die Programmiersprache **Pascal** konzipiert. Pascal ist nicht ganz so leicht zu erlernen wie z.B. Logo, zwingt den Programmierer aber mehr oder weniger zu strukturierten und leserlichen Programmen. In Pascal sind zudem diverse

Konzepte "eingebaut", durch die viele Fehler, die die Compiler anderer Sprachen einfach "durchgehen lassen", schon bei der Übersetzung des Programms erkannt werden können.

Jedem, der nicht nur "schnell mal eben" eine Programmiersprache, sondern das "Programmieren an sich", in fundierter Weise lernen möchte, kann Pascal nur wärmstens empfohlen werden. Wer mit Pascal das Programmieren gelernt hat, hat meist auch nur wenig Schwierigkeiten, eine weitere Sprache zu lernen — was man vom umgekehrten Weg leider nicht behaupten kann.

Auch für den Amiga gibt es bereits einen Pascal-Compiler von der britischen Firma *Metacomco*. Dieser entspricht hundertprozentig dem internationalen Sprach-Standard, wie ihn die ISO (International Standards Organisation) definiert hat und hat den zusätzlichen Vorteil, Programmstücke in Pascal, C und Assembler-Sprache innerhalb eines Programms kombinieren zu können.

Nach einer gewissen Zeit der Verwendung von Pascal stellt sich heraus, daß es bei allen Vorteilen dieser Sprache doch an einigen wichtigen Fähigkeiten für bestimmte Gruppen von Programmen mangelte. So wurde z.B. die Entwicklung großer Programmprojekte durch ganze Gruppen von Programmierern kaum unterstützt und es gab kaum Möglichkeiten auf die Hardware des verwendeten Computers zuzugreifen.

Die Entwickler von Pascal nahmen dies zum Anlaß, eine neue Sprache zu entwerfen, die zwar nicht als Ersatz, aber auf vielen Gebieten als Ergänzung zu Pascal gesehen werden kann: Modula (bzw. Modula-2). Sie beseitigt weitgehend alle als schwerwiegend empfundenen Mängel von Pascal und hat dabei aber genügend Ähnlichkeiten mit Pascal, um vom Programmierer nur minimale Umstellungen zu verlangen.

Die ETH Zürich verbreitet selbst eine Reihe von recht preiswerten Modula-2-Compilern für andere Computer. Ein Compiler für den Amiga befindet sich gerade in der Entwicklung und kann vermutlich ab Herbst 1986 direkt bei der ETH Zürich bezogen werden (umsonst bzw. gegen Erstattung der Versandkosten)!

17

Amiga-Basic

Die Programmiersprache des Amiga heißt Amiga-Basic. Das zugehörige Werkzeug befindet sich auf der Extras-Diskette, die jedem Amiga beim Kauf beiliegt und gibt so jedem Besitzer des Amiga die Gelegenheit, eigene Programme zu entwickeln oder auch nur ein wenig mit den Möglichkeiten der Computerprogrammierung herumzuspielen.

Der Leser möge es verzeihen, wenn er hier keine Einführung in die Computer-Programmierung finden wird und noch nicht einmal eine deutsche Anleitung für Amiga-Basic. Programmierung ist eine Sache, die man nicht in einem Buchkapitel abhandeln kann. Und Amiga-Basic ist eine so umfangreiche und vielseitige Sprache, daß ein Buch vom Umfang des hier vorliegenden, würde es sich ausschließlich damit beschäftigen, Amiga-Basic zu erläutern, sich wahrscheinlich noch recht knapp fassen müßte.

Die folgenden Abschnitte geben deshalb nur einen kurzen Überblick über die Möglichkeiten der Sprache Basic und des speziellen Dialekts Amiga-Basic. Sie werden so z.B. unter anderem erfahren, welche Möglichkeiten des Amiga sie in eigenen Programmen nutzen können (fast alle!) und wie Amiga-Basic gegenüber anderen Dialekten der Sprache Basic auf anderen Computern einzuordnen ist.

Falls Sie wirklich eigene Programme für den Amiga entwickeln wollen, so erhalten Sie Informationen dazu zusammen mit dem Amiga selbst im ausführlichen (aber zunächst leider englischen) Handbuch für Amiga-Basic. Falls Sie noch keinerlei Erfahrung mit der Programmierung von Computern haben, wird dieses Handbuch allerdings nicht ausreichen. Für die Einführung in die Sprache Basic existieren aber bereits einige hervorragende Lehrbücher, die Ihnen in diesem Fall helfen sollten. Viele Volkshochschulen bieten zudem inzwischen ebenfalls Basic-Kurse an. Diese sind zwar nicht speziell für den

Amiga ausgelegt, wenn Sie aber erst einmal ein Basic kennengelernt haben, werden Ihnen die anderen Dialekte keine allzu großen mehr Schwierigkeiten bereiten. Es könnte sogar sinnvoll sein, zunächst eine ganz andere Programmiersprache wie z.B. *Pascal* zu erlernen. Die Unterschiede zwischen den verschiedenen Computer-Sprachen zu bewältigen, ist bei weitem nicht so schwer wie ein Wechsel von Deutsch nach Französisch.

17.1 Basic

Basic ist mehr oder weniger die Computersprache, die fast jeder mit dem Microcomputer verbindet. Sie hatte aber schon eine lange Geschichte hinter sich, bevor die erste Zeile eines Basic-Programms auf einem Microcomputer ausgeführt wurde.

Das Wort Basic ist eine Abkürzung für *Beginners All Purpose Symbolic Instruction Code* (Allzweck-Instructionscode für Anfänger) und deutet damit schon auf den Zweck hin, zu dem die Sprache ursprünglich am Dartmouth-College entwickelt wurde: der Ausbildung von Anfängern in der Computer-Programmierung (in einem *Instructionscode*, wie man damals noch sagte). Auch heute noch wird Basic vielfach für diese Zwecke eingesetzt. Nach wie vor ist einer der Hauptvorteile dieser Sprache nämlich die schnelle Erlernbarkeit, was dazu führt, daß jeder Anfänger relativ schnell eigene Programme schreiben und so seine ersten Erfolgserlebnisse am Computer haben kann.

Die ersten Basic-Realisierungen für Microcomputer waren aber (aus Gründen des damals noch sehr knappen Speicherplatzes) notwendigerweise sehr beschränkt. Basic-Dialekte auf Microcomputern waren lange Zeit recht primitive Sprachen, die wirklich nur die allernötigsten Hilfsmittel der Programmierung zur Verfügung stellten. Das führte in der Folgezeit dazu, daß Basic in Kreisen der Computer-Wissenschaft (in Deutschland heißt sie *Informatik*) einen sehr schlechten Ruf bekam, weil es mit diesen primitiven Basic-Formen sehr leicht war, Programme zu schreiben, die nach kurzer Zeit niemand mehr verstehen und vor allem ändern konnte. Gerade das Verstehen und Ändern alter Programme ist aber eine der Hauptaufgaben vieler Programmierer und kostet die Programm-Hersteller auf Dauer mehr als das Erstellen einer ersten Programmversion.

Einer der Vorwürfe, die man Basic oft machte, war z.B., daß die Sprache zu sogenanntem *Spaghetti-Code* verführt. Viele Programme in den frühen Basic-

Versionen waren nämlich lange Ketten von unstrukturierten Programmzeilen, zwischen denen das Programm bei der späteren Ausführung hin und her hüpfen mußte. Wenn man diese Programmsprünge mit einem Bleistift nachzuvollziehen versuchte, ergab dies ein wildes Durcheinander von Linien – eben den berühmten Spaghetti-Code.

Zudem kannten frühe Basic-Versionen keine Namen für Teilprogramme (Subroutinen oder Prozeduren), sondern nur Nummern. Stand in einem anderen Programmteil nun der Aufruf einer solchen Prozedur, mußte man erst die Bedeutung dieser Nummer nachschlagen (sofern sie überhaupt irgendwo dokumentiert war und nicht nur der Programmierer allein wußte, was das entsprechende Unterprogramm tun sollte), wenn man wissen wollte, was das Programm in diesem Teil tun sollte. In anderen Programmiersprachen war es zu dieser Zeit längst üblich, Unterprogrammen sinnvolle Namen zu geben und sie auch mit diesen Namen aufzurufen. Statt **GOSUB 5390** in Basic konnte man schreiben **Zeige_Daten_Von(Müller)**.

Im Laufe der Zeit wurde Basic aber auch auf Microcomputern immer "erwachsener", was nicht zuletzt damit zusammenhängt, daß der Speicher nicht mehr so teuer ist, und die Basics größer werden und so auch mehr Möglichkeiten enthalten konnten. Das erste Basic auf einem Microcomputer war z.B. gerade 4 KByte groß; Amiga-Basic ist immerhin schon 80 KByte groß und ist nicht einmal das größte Basic auf Microcomputern.

Viele heutige Basic-Dialekte (wie z.B. Amiga-Basic) enthalten alle wesentlichen Merkmale einer modernen Programmiersprache. Sie erlauben mühelos die sogenannte *strukturierte Programmierung*, mit Hilfe derer Programme so übersichtlich geschrieben werden können, daß man sie auch noch Jahre, nachdem sie geschrieben wurden, verstehen und ändern kann. Unterprogramme und Programmvariablen können sinnvolle Namen erhalten und eine Vielzahl wichtiger mathematischer und andere Funktionen sind fest eingebaut und brauchen nicht erst mühselig von jedem Programmierer neu geschrieben zu werden.

Basic hat im Laufe der Jahre viele gute und erprobte Fähigkeiten anderer Programmiersprachen aufgenommen und läßt auch hinsichtlich des Komforts bei der Programmentwicklung und Fehlersuche heute nicht mehr viele Wünsche offen. Im nächsten Abschnitt wollen wir uns den Dialekt "Amiga-Basic" einmal genauer betrachten, um zu sehen, wie er im Vergleich abschneidet.

17.2 Der Basic-Dialekt "Amiga-Basic"

Das Basic des Amiga stammt von der Firma Microsoft®. Dieses Softwarehaus ist bezüglich der Sprache Basic praktisch der "Hauslieferant" für fast alle Hersteller von Microcomputern. Das erste Basic für einen der ersten Microcomputer stammte von Microsoft, und bis heute gibt es kaum einen Microcomputer, für den es kein Microsoft- (MS-) Basic gibt. Es ist meist die erste Programmiersprache für jeden neuen Computer.

17.2.1 Microsoft Basic

Diese weite Verbreitung von MS-Basic ist ein enormer Vorteil für den Programmierer in dieser Sprache. Alle Basic-Versionen der Firma Microsoft sind sich nämlich (in ihrer jeweils neuesten Version) weitgehend ähnlich. Wenn man Microsoft-Basic auf einem Computer kennt und dann auf einen anderen überwechselt, gibt es deshalb sehr wenig Umstellungsschwierigkeiten.¹ Viele Programme können sogar mit sehr wenigen Änderungen direkt von einem Computer zum anderen übernommen werden, so daß man besonders bei neuen Computern auch zu Anfang gleich Software zur Verfügung hat.

Auch wenn Sie ein gutes Lehrbuch für Amiga-Basic suchen, hilft Ihnen die weite Verbreitung von MS-Basic. Es dauert nämlich naturgemäß einige Zeit, bis gute Programmierbücher für einen neuen Computer auf den Markt kommen. Falls Sie aber ein gutes Basic-Buch für den IBM-PC® (PC-Basic® oder MS-Basic®) oder noch besser den Apple® Macintosh® finden, zögern Sie nicht, zuzugreifen. Alle "wesentlichen" Eigenschaften des Basic für den IBM-PC und den Amiga sind gleich. Die Ähnlichkeiten bezüglich der Verwaltung von Menüs, Maus und Fenstern zwischen Amiga-Basic und MS-Basic für den Apple Macintosh sind sogar noch größer.²

¹ Dies betrifft natürlich nicht die speziellen Fähigkeiten der verschiedenen Computer, die natürlich in den verschiedenen Basics auch unterschiedlich unterstützt werden müssen. Viele andere Computer haben nun einmal keine Maus, Fenster und hochauflösende Farbgrafik wie der Amiga.

² Amiga Basic ist in Wirklichkeit eine speziell auf den Amiga zugeschnittene und angepaßte Version des Macintosh-MS-Basic.

17.2.2 Moderne Features von Microsoft Basic

Einige der wichtigsten Eigenschaften des MS-Basic für den Amiga sollen nun stichwortartig aufgelistet werden. Es gibt sie fast alle in jeder neueren Version von MS-Basic und sie sind deshalb nicht Amiga-spezifisch. Es sind aber Möglichkeiten, die durchaus nicht in jedem Basic zu finden sind, und die deutliche Fortschritte gegenüber den primitiven frühen Basic-Versionen (s.o.) darstellen.

- **Optionale Zeilennummern:** Alle Zeilen eines Programms **können, müssen aber nicht**, mit Zeilennummern versehen werden. MS-Basic benötigt intern diese Zeilennummern höchstens noch für die Programmzeilen, die Ziel eines Programmsprungs (GOTO) sind. Nur die wenigsten Zeilen eines Programms sind aber normalerweise Ziel eines Programmsprungs. Diese Zeilen können mit einer Nummer versehen werden. Noch besser ist es aber, sie mit einer lesbaren Marke zu versehen (s.u.).

- **Programm-Marken:** Wichtige Stellen eines Programms, z.B. die Ziele von Sprunganweisungen, können mit einem lesbaren Namen, einer sogenannten *Marke* (engl. *Label*), versehen werden. Diese wichtigen Stellen werden so deutlich hervorgehoben und das Programm in überschaubare Abschnitte gegliedert.

- **Anweisungs-Blocks:** Eine beliebig lange Folge von Zeilen eines Programms kann zu einem *Block* zusammengefaßt werden. Solche Blöcke finden z.B. bei bedingten Anweisungen (IF-THEN-ELSE-ENDIF), Programmschleifen (FOR-NEXT, WHILE-WEND) oder Unterprogrammen (SUB-END SUB) Anwendung. Die Verwendung von unübersichtlichen Programm-Sprüngen (GOTO-Anweisungen) für diese Zwecke ist nicht mehr nötig.

- **Lange Namen:** Alle Bezeichnungen für Variablen, Marken, Unterprogramme usw. können lange selbsterklärende Namen sein und so jedes Programm wesentlich lesbarer machen. Alle diese Namen können bis zu 40 Buchstaben lang werden, und alle diese 40 Buchstaben werden auch zur Unterscheidung der verschiedenen Namen eingesetzt.

- **Reiche Auswahl an Datentypen:** Für Variablen und die Ergebnisse von Funktionen stehen insgesamt 5 verschiedene *Datentypen* zur Verfügung. Amiga-Basic kennt zwei Genauigkeitsstufen für ganze Zahlen (16 Bit und 32 Bit), zwei Genauigkeitsstufen für Fließpunktzahlen (32 Bit und 64 Bit) und Strings (Texte) die bis zu 32 768 Buchstaben enthalten können. Arrays (Felder bzw. Matrizen) können bis zu 255 Dimensionen haben und in jeder Dimension bis zu 32 768 Elemente enthalten.

- **Subprogramme:** Neben den auch aus anderen Basic-Dialekten bekannten *Subroutinen* gibt es auch echte *Subprogramme* oder Prozeduren, die die Übersichtlichkeit großer Programme wesentlich verbessern und ihre Fehleranfälligkeit verringern können. Subprogramme haben stets einen (hoffentlich aussagekräftigen) Namen, können Parameter besitzen, die beim Aufruf mit einem Wert belegt werden und können eigene Variablen besitzen, die von denen des Hauptprogramms verschieden sind, selbst wenn sie genauso heißen. Subprogramme können sich sogar selbst aufrufen, was gar nicht so selten ist, wie man vielleicht meint (der Informatiker nennt solche Subprogramme dann *rekursiv*).

- **Programm-Chaining:** Jedes Basic-Programm kann andere Basic-Programme aufrufen und diesem sogar (über die COMMON-Anweisung) Daten übergeben. Dies nennt man *Programm-Verkettung* oder *-Chaining*. Auf diese Weise kann man sehr große Programme entwickeln, von denen aber immer nur die gerade benötigten Teile im Speicher sind. So bleibt mehr Platz für Daten frei.

- **Komfortable Dateiverwaltung:** MS-Basic besitzt reichhaltige Möglichkeiten für die Verarbeitung von Dateien, die auf der Diskette oder einer Festplatte gespeichert sind. Solche Dateien können in Datensätze gleicher Struktur aufgeteilt werden, bei denen jeder Datensatz aus einer Reihe von Feldern mit aussagekräftigen Namen besteht. Solche Dateien können *sequentiell* (also Datensatz für Datensatz) oder auch *mit wahlfreiem Zugriff* (engl. *Random Access*) auf jedem beliebigen Datensatz bearbeitet werden.

17.3 Spezielle Fähigkeiten des Amiga

Obwohl Amiga-Basic sehr ähnlich zu den anderen MS-Basic-Dialekten auf anderen Computern ist, hat Microsoft es aber doch auch sehr sorgfältig an die speziellen Fähigkeiten des Amiga angepaßt. Fast jede dieser Fähigkeiten, die den Amiga ja erst zu einem so außergewöhnlichen Computer machen, kann von Basic aus genutzt werden. Es ist also problemlos möglich, eigene Programme zu schreiben, die wie "richtige Amiga-Programme" aussehen; mit eigenen Screens, Fenstern, Menüs und Maus-Bedienung. Zugleich werden diese Möglichkeiten aber auch für die Programmentwicklung selbst genutzt und machen dem Programmierer so das Leben leichter.

17.3.1 Umgang mit Amiga-Basic

Amiga-Basic ist, wie die meisten Basic-Systeme, eine sogenannter *Interpreter*. Dies bedeutet, daß für die Ausführung eines Programms immer ein kleines Programm (eben der Interpreter) benötigt wird, das Zeile für Zeile durch den Programmtext läuft, jeweils eine Anweisung liest und sie ausführt. Dies ist ein verhältnismäßig (für einen Computer) langsamer Prozeß. Schnellere Programme erhält man meist durch einen *Compiler*, der einen Programmtext nimmt und in einen Code übersetzt, der direkt von der CPU des Computers ausgeführt werden kann, den sogenannten *Maschinen-Code*. Auch für Amiga-Basic wird sicherlich über kurz oder lang ein solcher Compiler auf den Markt kommen.

Die Arbeit mit einem Interpreter hat aber unschätzbare Vorteile in Hinsicht auf den Bedienungskomfort. Während die oben beschriebene Übersetzung eines Programms in Maschinen-Code ein langwieriger und vor allem für Anfänger fehleranfälliger Vorgang ist, braucht man Programme für Amiga-Basic nur einzutippen und kann sie sofort starten. Auch jede Programmänderung kann sofort ausprobiert werden.

"Normalerweise" arbeitet man mit Amiga-Basic in zwei verschiedenen Fenstern. In einem, dem sogenannten *Output-Fenster*, erledigt das laufende Programm seine Ein- und Ausgaben und man kann hier auch über die Tastatur Befehle an das Basic-System erteilen. Das zweite Fenster, das sogenannte *Listing-Fenster*, enthält den Text des gerade bearbeiteten Programms. Basic-Programme können darüberhinaus beliebig viele zusätzliche Fenster öffnen, für deren Inhalt sie dann auch selbst zuständig sind. Aber auch die beiden ersten Fenster müssen nicht immer sichtbar sein. Wenn ein Programm fehlerfrei läuft, wird man üblicherweise das Listing-Fenster schließen und Programme, die selbst eigene Fenster öffnen, werden oft auch das erste Output-Fenster schließen und alle Ein- und Ausgaben in den eigenen Fenstern erledigen.

17.3.1.1 Eingabe und Editieren von Programmen

Die Eingabe und das Ändern von Programmen im Listing-Fenster ist relativ komfortabel. Die Bearbeitung des Programmtextes selbst geht ähnlich wie im Notepad oder in einem Textcraft-Dokument. Ein blinkender Strich, die Schreibmarke (engl. *cursor*) markiert immer die Stelle, an der neu eingegebene Buchstaben erscheinen. Er kann mit der Maus oder mit den vier Cursor-Tasten beliebig im ganzen Programmtext verschoben werden. Ein Verschieben des Fenster über ein großes Programm geschieht jedoch nicht mit Rollbalken, wie aus der Workbench oder von Textcraft bekannt, sondern

ebenfalls mit den Cursor-Tasten (indem man gleichzeitig mit einer Cursor-Taste die <SHIFT>-Taste festhält).

Indem man eine beliebige Stelle im Text mit der linken Maustaste anklickt und bei gedrückter Taste zu einer anderen Stelle zieht, können größere Bereiche im Text selektiert werden. Einzelne Worte können auch durch einen Doppelklick über einer beliebigen Stelle des Wortes selektiert werden. Ein so selektierter Bereich kann mit dem Menü-Befehlen aus dem Menü **Edit** dann im ganzen bearbeitet werden. **Cut** entfernt ihn, legt ihn aber zugleich in eine Zwischenablage (engl. *clipboard*). **Copy** kopiert ihn nur in die Zwischenablage. **Paste** fügt schließlich den Text aus der Zwischenablage bei der Schreibmarke in den Programmtext ein. Die Arbeit mit diesen drei Befehlen ist sehr ähnlich wie im Programm Graphicraft oder der Umgang mit Schere, Kamera und Klebstofftube in Textcraft. Mit Hilfe des **Edit**-Menüs kann man sehr leicht auch größere Umstellungen im Programm vornehmen.

Ein großer Vorteil von Amiga-Basic gegenüber primitiveren Versionen dieser Sprache sind die Möglichkeiten zur Formatierung eingegebener Programme. So darf das Innere von Schleifen oder IF-THEN-ELSE-Anweisungen eingerückt werden, wodurch solche Blöcke natürlich gleich viel mehr ins Auge stechen. Die <TAB>-Taste ist ideal für solche Einrückungen. Sie fügt jeweils 3 Leerzeichen bei der Schreibmarke ein.

Neben diesen Formatierungen, für die man selbst sorgen muß, gibt es aber auch automatische Formatierungen, die Amiga-Basic selbst durchführt. Wenn Sie z.B. ein Basic-Schlüsselwort, wie z.B. IF, FOR oder GOTO eingegeben haben und in eine neue Zeile gehen, schreibt Basic diese Schlüsselworte automatisch groß, egal, wie Sie sie eingegeben haben. Dies erhöht natürlich ebenfalls die Lesbarkeit des Programms und hilft Ihnen zugleich, Tippfehler zu entdecken. Wollten Sie nämlich ein Schlüsselwort eingeben und es erscheint nicht in Großbuchstaben, muß sich darin ein Tippfehler befinden. Verwenden Sie aber einen Variablen- oder Subprogramm-Namen und dieser erscheint plötzlich in Großbuchstaben, obwohl Sie ihn klein eingegeben haben, so gibt es ein Schlüsselwort gleichen Namens, woraufhin Sie sich einen anderen Namen ausdenken sollten.

17.3.1.2 Fehlersuche

Tauchen während des Programmablaufs Fehler auf, so stoppt Basic, gibt eine entsprechende Fehlermeldung aus und zeigt Ihnen die Fehlerstelle auch durch einen dünnen Rahmen um die entsprechende Stelle im Programmtext. War das Listing-Fenster geschlossen, wird es hierzu zuvor geöffnet. Sie können den Fehler nun beseitigen und das Programm noch einmal laufen lassen.

Basic selbst findet aber natürlich nur recht grobe Fehler, z.B. Tippfehler. Die meisten **logischen** Fehler müssen Sie auf andere Weise herausfinden, wofür es aber eine ganze Reihe von Hilfen gibt. Diese finden sich alle im Menü **Run**.

Der erste Befehl **Start** startet (naheliegenderweise) das gerade bearbeitete Programm. Lief dieses Programm schon einmal und wurde es durch Sie oder einen Fehler unterbrochen, so beginnt es nun wieder von vorn. **Stop** ist natürlich das Gegenteil von **Start** und hält ein laufendes Programm an. Sie können dann die Programmausgaben am Bildschirm in Ruhe betrachten, sich die Werte der verschiedenen Programmvariablen zeigen lassen und diese bei Bedarf ändern, bevor Sie das Programm mit dem Befehl **Continue** weiterlaufen lassen. **Suspend** wirkt ähnlich wie **Stop**, nur wird hierbei das Programm nur solange unterbrochen, bis auf eine beliebige Taste gedrückt wird. Es können also keine Variablen betrachtet oder andere Experimente ausgeführt werden, während das Programm anhält.

Der Befehl **Trace On** verschafft Ihnen eine genaue Kontrolle über den Ablauf eines Programms. Nachdem **Trace On** ausgewählt wurde, zeigt Amiga-Basic während der Ausführung eines Programms kontinuierlich an, in welcher Zeile sich das Programm gerade befindet. Hierzu wird im Listing-Fenster die gerade ausgeführte Zeile orange umrahmt. Dieser Modus kann mit **Trace Off** wieder abgeschaltet werden. Der Befehl **Step** schließlich gibt Ihnen eine noch feinere Kontrolle über den Programmablauf. Mit ihm können Sie Schritt für Schritt durch ein Programm gehen. Nach der Ausführung jeder Anweisung des Programms hält Amiga-Basic automatisch an, als hätte es den **Stop**-Befehl erhalten.

17.3.2 Unterstützung der Amiga-Fähigkeiten

Amiga-Basic sieht eine ganze Reihe von Funktionen vor, die es erlauben, von einem Basic-Programm aus alle wichtigen Elemente der Amiga-Benutzerschnittstelle zu verwenden. Dies ist natürlich wichtig für jeden, der Programme schreiben möchte, die wie "echte Amiga-Programme aussehen", also sich von ihrer Bedienung her nicht von professionellen Programmen unterscheiden.

Im folgenden werden Sie einen kleinen Überblick über diese Möglichkeiten finden. Er ist wieder recht stichwortartig gehalten und soll Ihnen nur einen Eindruck von den Möglichkeiten verschaffen, die Amiga-Basic bietet. Detaillierte Erläuterungen der einzelnen Befehle finden sich im Kapitel 8 des Amiga-Basic-Handbuchs.

17.3.2.1 Grafik

Der einfachste Grafik-Befehl von Amiga-Basic ist der LINE-Befehl. Mit ihm können Linien zwischen zwei Punkten in einer wählbaren Farbe gezogen werden. Alternativ kann er auch dazu verwendet werden, Rechtecke zu zeichnen, die auf Wunsch gleich mit einer bestimmten Farbe gefüllt werden. Der Befehl CIRCLE kann Kreise, Ellipsen sowie Kreisbögen mit einem bestimmten Radius um einen festgelegten Mittelpunkt herum zeichnen. Auch diese können auf Wunsch gleich gefüllt werden. Komplexere Formen, Polygone (Vielecke), können mit dem AREA-Befehl Punkt für Punkt definiert und mit AREAFILL dann gezeichnet werden.

Noch komplexere Formen als die oben aufgeführten muß man aus diesen "Elementen" zusammensetzen. Sobald eine Fläche der gewünschten Form aber erst einmal einen Rahmen hat, kann sie mit Befehl PAINT jederzeit mit einer beliebigen Farbe gefüllt werden. Dies wirkt genauso wie die Anwendung des Farbeimers in Graphicraft oder Deluxe Paint.

Bei den Farben, die für eine Basic-Grafik verwenden werden können, ist man natürlich nicht auf die vier "Workbench-Farben" beschränkt. Mit dem Befehl PALETTE kann man diese jederzeit ändern. Über PALETTE kann man jedes der Farbbregister im Video-Chip Denise mit neuen Rot-, Grün- und Blau-Werten füllen. Wenn vier Farben nicht genügen, der kann sich mit SCREEN (s.u.) einen eigenen Screen erzeugen, der so viele Farben haben kann, wie er möchte.

17.3.2.2 Fenster und Screens

"Normalerweise" gehen alle Ausgaben eines Programms in das eine Output-Fenster, das auch am Anfang, wenn man Basic startet, geöffnet ist. Auch die oben beschriebenen Grafikbefehle arbeiten zunächst in diesem Fenster. Wenn ein Fenster aber nicht genügt oder das Standard-Output-Fenster nicht gefällt, der kann sich mit dem Befehl WINDOW ein neues erzeugen, dessen Titel, anfängliche Größe usw. völlig frei gewählt werden können. Auch Fenster, die von Programmen erzeugt wurden, können als Output-Fenster verwendet werden (man kann z.B. PRINT-Befehle in sie schicken und in ihnen können natürlich auch Grafiken entstehen). Von einem Programm erzeugte Fenster, aber auch die beiden Standard-Fenster, können jederzeit auch von einem Programm aus mit WINDOW CLOSE wieder geschlossen werden.

Maximal zehn Fenster können von Basic aus verwaltet werden.

Für anspruchsvolle Grafiken genügt es aber kaum, ein spezielles Fenster zu öffnen. In Basic stehen einem ja "normalerweise" nur ein Screen mit vier

Farben und 640 mal 320 Punkten zur Verfügung. Braucht man mehr Farben, eine andere Bildschirmauflösung oder möchte man gar den HAM-Modus des Denise-Chips einmal ausprobieren, kann man mit SCREEN einen neuen virtuellen Bildschirm erzeugen und ihn mit SCREEN CLOSE wieder entfernen, sobald er nicht mehr gebraucht wird.

Maximal vier Screens können von Basic aus verwaltet werden.

17.3.2.3 Menüs

Die Pulldown-Menüs sind eine der wichtigsten Komponenten der Amiga-Benutzerschnittstelle Intuition. Auch von Amiga-Basic aus können deshalb neue Menüs und Menü-Punkte erzeugt werden. Der Basic-Befehl MENU tut genau dies. Amiga-Basic kann allerdings nur reine Text-Menüs erzeugen. Grafische Menüs, wie die Palette oder die Hilfsmittelauswahl in Graphicraft, sind nicht möglich.

Mit einer zweiten Form des MENU-Befehls kann eine Programm jederzeit feststellen, aus welchem Menü der Benutzer zuletzt welchen Befehl ausgewählt hat. Zum Feststellen der Tatsache, ob der Benutzer überhaupt eine Menü-Auswahl getätigt hat, wird ein anderer Mechanismus verwendet, der weiter unten noch erläutert wird.

Ein Basic-Programm kann maximal 10 verschiedene Menüs verwenden. Jedes Menü kann bis zu 19 Punkte enthalten.

17.3.2.4 Die Maus

Einen Teil der Mausbewegungen und Mausclicks überwacht und interpretiert ja bereits die Amiga-Benutzerschnittstelle Intuition (z.B. das Bewegen von Fenstern und Herunterholen von Maus). Mausclicks innerhalb eines Fenster müssen aber normalerweise von dem für dieses Fenster zuständige Programm behandelt werden. Ein Basic-Programm verwendet dafür üblicherweise den Befehl MOUSE. Mit diesem kann man feststellen, ob die linke Maustaste im Moment gedrückt ist oder nicht, über welchem Punkt sie gedrückt und wieder losgelassen wurde und ob es ein Doppelklick war oder nicht.

Die rechte (Menü-) Taste der Maus kann von Basic aus nicht abgeprüft werden. Sie kann nur indirekt über die MENU-Funktion von einem Basic-Programm benutzt werden.

17.3.2.5 BOBs und Sprites

Eine der faszinierendsten Möglichkeiten, die der Amiga grafisch bietet, sind ja bewegte Objekte. Diese gibt es dabei in zwei "Geschmacksrichtungen": die in Hardware erzeugten Sprites, die schnell sind, aber diversen Einschränkungen unterliegen oder die etwas langsameren, aber wesentlich vielseitigeren BOBs (Blitter Objects), die von der Systemsoftware des Amiga und dem Blitter ermöglicht werden. In Amiga-Basic werden beide Typen einfach *Objekte* genannt und immer mit demselben Satz von Befehlen behandelt.

Für die Definition des Aussehens von Objekten befindet sich ein separates Program namens *Objedit* auf der Extras-Diskette. Es ist übrigens in Basic geschrieben und demonstriert so recht eindrucksvoll die Möglichkeit, sinnvolle Amiga-Programme in Basic zu entwickeln.

Die mit diesem Programm entwickelte Form eines Objektes kann dann mit OBJECT.SHAPE für die Erzeugung eines neuen Sprites oder BOBs verwendet werden. Dieses kann man danach mit OBJECT.X und OBJECT.Y an einem bestimmten Punkt am Bildschirm zeigen und ihm mit OBJECT.VX und OBJECT.VY eine Geschwindigkeit (engl. *Velocity*) geben, mit der es sich (ohne weiteres Zutun des Programms!) über den Bildschirm bewegt. Mit OBJECT.AX und OBJECT.AY bekommt es dann zugar noch eine Beschleunigung (engl. *Acceleration*), wodurch die Geschwindigkeit des Objekts sich mit der Zeit automatisch ändert. Treffen zwei Objekte aufeinander oder versuchen sie einen "zulässigen" Bereich zu verlassen, kann sich ein Programm davon informieren lassen (vgl. den nächsten Abschnitt) und mit dem Befehl COLLISION feststellen, welches Objekt auf welches Hindernis gestoßen ist.

Mit OBJECT.PRIORITY kann schließlich auch noch festgelegt werden, welches Objekt "vor" welchem anderen erscheint, wenn zwei oder mehr Objekte an ein und derselben Stelle am Bildschirm stehen. 3-D-Effekte sind so auch für ein Basic-Programm kein Problem.

In einem Basic-Programm können beliebig viele bewegte Objekte verwendet werden, solange der freie Speicherplatz ausreicht (BOBs benötigen teilweise recht viel Speicherplatz).

17.3.2.6 Abfangen von Ereignissen

Auf viele Ereignisse muß ein Programm sehr schnell reagieren können. Wenn zwei Objekte zusammenstoßen oder die linke Maus-Taste niedergedrückt wird, muß am besten sofort eine Reaktion erfolgen. Dies ist aber, wenn es

sehr viele solche Ereignisse gibt, die ja alle regelmäßig und möglichst schnell abgeprüft werden müssen, etwas schwer zu programmieren..

Amiga-Basic bietet deshalb mit dem ON...GOSUB-Befehl (die drei Pünktchen deuten ein anderes Wort an, das hier eingesetzt werden muß) eine Möglichkeit, ein bestimmtes Ereignis mit einer bestimmten Subroutine fest zu verknüpfen. Dies nennt man dann *Abfangen des Ereignisses* (engl. *Event Trapping*). Immer wenn dieses Ereignis, z.B. ein Mausklick, eintritt, ruft Amiga-Basic sofort diese Subroutine auf, egal welcher andere Programmteil gerade aktiv ist. Wenn die Subroutine beendet ist, geht die Kontrolle automatisch wieder an die Stelle zurück, an der diese Unterbrechung auftrat. Eine sehr schnelle Reaktion auf bestimmte Ereignisse kann so stets gewährleistet werden.

Die Ereignisse, die auf diese Weise abgefangen werden können, sind:

- Die Auswahl eines Menü-Punktes (ON MENU)
- Ein Mausklick (ON MOUSE)
- Das Ablaufen einer bestimmten Zeitspanne (ON TIMER)
- Die Kollision zweier Objekte (ON COLLISION)
- Der Versuch des Benutzers, ein Programm zu stoppen (ON BREAK)
- Ein Programmfehler (ON ERROR)

Die Subroutine, die auf ein bestimmtes Ereignis hin von Amiga-Basic aufgerufen wird, kann jederzeit mit einem erneuten Aufruf von ON...GOSUB wieder geändert werden. Je nach Programmsituation können also andere Programmteile ein bestimmtes Ereignis bearbeiten.

17.3.2.7 Musik- und Sprachausgabe

Und last-but-not-least können auch die musikalischen und sprachlichen Fähigkeiten des Amiga von Basic aus genutzt werden. Der Befehl SOUND gibt eine Note einer bestimmten Frequenz mit einer bestimmten Dauer und Lautstärke auf einem der vier Klang-Kanäle aus. Für jeden Klang-Kanal kann zuvor eine Wellenform mit dem Befehl WAVE festgelegt werden.

Die Befehle TRANSLATE und SAY schließlich dienen der Sprach-Synthese. TRANSLATE wandelt einen (englischen) Text in die Phonemschreibweise um und SAY gibt einen Text in Phonemschreibweise aus. Um wirklich gute Ergebnisse (oder deutsche Sprache) zu erzielen, wird man die Phonem-schreibweise eines Wortes im allgemeinen nicht durch TRANSLATE ermitteln lassen, sondern mit einem guten Wörterbuch feststellen.

17.3.3 Amiga-Basic ohne Grenzen

Obwohl Amiga-Basic bereits eine Vielzahl von Befehlen enthält, kann es doch immer wieder vorkommen, daß zusätzliche Fähigkeiten benötigt werden, die in Basic nicht realisierbar sind. Von Basic aus können z.B. keine Requester und Gadgets erzeugt und benutzt werden. Fortgeschrittene Programmierer haben aber dennoch die Möglichkeit, ohne allzu große Mühe auch diese Einschränkungen zu umgehen. Diese Möglichkeiten sind aber "mit Vorsicht zu genießen", da einige Schritte dabei recht fehleranfällig sind.

Funktionen, die diese fehlenden Fähigkeiten realisieren, befinden sich nämlich zum großen Teil bereits in den Bibliotheks-Dateien, aus denen die Systemsoftware des Amiga zum Teil besteht.

Amiga-Basic besitzt eine recht elegante Konstruktion, mit der es möglich ist, solche Funktionen in einem Basic-Programm zu verwenden, als wären sie in Basic geschrieben und würden sich im selben Programm befinden. Mit dem **LIBRARY**-Befehl können eine Reihe solcher Bibliotheks-Dateien angegeben werden, die Amiga-Basic durchsucht, um solche Befehle zu finden. Wird danach in einem Programm ein Befehl verwendet, der nicht im Programm selbst als Subprogramm definiert ist, versucht Amiga-Basic, ihn in einer Bibliothek (engl. *Library*) zu finden.

Solche Bibliotheken müssen aber nicht zur Amiga-Systemsoftware gehören. Auch eigene Funktionspakete, die man z.B. aus Geschwindigkeitsgründen in Maschinensprache geschrieben hat, würde man typischerweise in Bibliotheken ablegen. Für Basic spielt dies keine Rolle. Die einzelnen Funktionen müssen sich nur an gewisse Aufruf-Konventionen halten, die zusammen mit den Funktionsnamen in einer von der eigentlichen Bibliothek getrennten Datei verzeichnet sein müssen.³ Diese Datei trägt immer denselben Namen wie die Bibliotheks-Datei, aber mit einem zusätzlichen ".bmap" dahinter.

Die Beschreibungen der Bibliotheksfunktionen kann man zunächst in recht lesbarer Form erstellen. Das Format dafür ist in einem Anhang des Amiga-Basic-Handbuchs erläutert. Ein besonderes Programm namens **ConvertFd.bas** auf der **Extras**-Diskette wandelt diese Beschreibung dann in das kompakte ".bmap"-Format um, das der **LIBRARY**-Befehl verlangt.

Amiga-Basic ist also nicht nur ein vielseitiger und komfortabler Dialekt der Programmiersprache Basic, sondern auch grenzenlos erweiterbar. Es ist deshalb vielleicht die einzige Sprache, die Sie jemals benötigen werden.

³ Die Aufrufkonventionen entsprechen denen der Sprache C, da ein Großteil der Amiga-Systemsoftware in dieser Sprache geschrieben wurde.

Fairerweise muß man jedoch dazu sagen, daß es viele Programmierprobleme gibt, für die man vielleicht doch besser eine andere Programmiersprache einsetzen sollte — aber eben nicht unbedingt muß.

18

Schnittstellen

Zum Abschluß dieses Buches wollen wir uns nun noch mit einem Thema beschäftigen, das für jeden Computerbesitzer irgendwann einmal wichtig wird: den Verbindungen zur Außenwelt oder *Schnittstellen*, wie der Fachmann sagt. In diesem Kapitel werden Sie erfahren, welche Verbindungsmöglichkeiten vom Amiga zu externen Geräten vorhanden sind und wie die Kabel aussehen müssen, über die man welche Geräte anschließen kann. Wir werden uns dabei aber nur mit den einfacheren und "ungefährlich - eren" dieser Schnittstellen intensiver beschäftigen. Denjenigen unter den Lesern, die "ausgefallene" Geräte an ihren Amiga anschließen wollen, oder gar selbst Geräte entwickeln und diese anschließen wollen, sei das *Hardware Manual* des Amiga empfohlen, in dem sie alle dafür nötigen Informationen finden

18.1 Überblick

Der Amiga besitzt eine reiche Auswahl der verschiedensten *Schnittstellen* oder Anschlußbuchsen, an die man eine Vielzahl von Geräten anschließen kann. Die meisten davon befinden sich auf der Rückseite der Systemeinheit. Einige dieser Schnittstellen sind an Ihrem Amiga bestimmt schon "verkabelt"; einige werden aber auch noch frei sein.

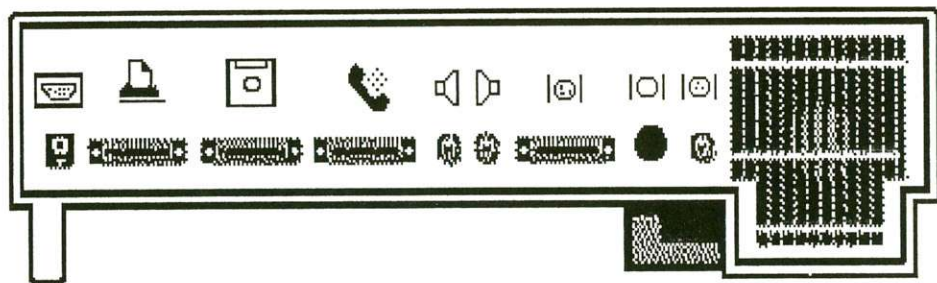


Bild 18 - 1: Die Rückseite des Amiga

Wenn Sie einmal die Rückseite Ihres Amiga betrachten, können Sie die meisten der "interessanten" Schnittstellen sehen. Sie sind alle mit kleinen Bildchen versehen, die schon auf die Geräte hinweisen, die man hier "üblicherweise" anschließt.

Ganz links befindet sich der Anschluß für die Tastatur. Buchse und Stecker entsprechen dem Standard für amerikanische Telefongeräte. Dies hat man aber nur aus praktischen Erwägungen gemacht. Die Signale im Spiralkabel der Tastatur haben in keinsten Weise mit Telefon-Signalen, weder amerikanischen noch deutschen, etwas zu tun. Hier sollten Sie auch niemals andere Geräte anschließen.

Rechts neben dem Tastaturanschluß befindet sich die parallele Schnittstelle, die üblicherweise für den Anschluß eines Druckers an den Amiga verwendet wird. Mehr über dieser Schnittstelle erfahren Sie weiter unten.

Die nächste Schnittstelle dient zum Anschluß eines zusätzlichen Diskettenlaufwerks. Hier kann sowohl ein zweites 3,5-Zoll-Laufwerk als auch ein IBM-kompatibles 5,25-Zoll-Laufwerk angeschlossen werden. Falls Sie sich mit den "Innereien" von Computern nicht **sehr** gut auskennen, sollten sie Basteleien an dieser Schnittstelle unterlassen, da sie mit sehr empfindlichen Teilen im Amiga verbunden ist.

Neben dem Diskettenanschluß befindet sich die serielle Schnittstelle, die mit dem Symbol eines Telefonhörers versehen ist. Dieses Symbol deutet an, daß man hier oft ein sogenanntes MODEM anschließt, das die Kommunikation mit anderen Computern über Telefonleitungen ermöglicht. Auch über diese Schnittstelle erfahren Sie weiter unten noch mehr.

Die zwei kleinen Buchsen neben der seriellen Schnittstelle sind die Lautsprecherausgänge. Sie können diese an Ihre Stereoanlage oder auch den Bildschirm anschließen. Mehr zu den dafür nötigen Kabeln in einem folgenden Abschnitt.

Die restlichen drei Anschlüsse dienen der Verbindung des Amiga mit den verschiedensten Bildschirmen. Sie bedeuten von links nach rechts eine abnehmende Bildqualität und einen ebenso deutlich abnehmenden Preis der zugehörigen Bildschirme.

Damit hätten wir die Anschlüsse an der Rückseite des Amiga schon "durch". Einige weitere Schnittstellen befinden sich aber noch an anderen Stellen der Systemeinheit.

An der rechten Seite des Gehäuses befinden sich z.B. vorne zwei Anschlüsse für Steuerungsgeräte wie Mäuse, Steuerknüppel (Joysticks), Rollkugeln (Trackballs) und Lichtgriffel (Lightpens). Der vorderste davon wird bei Ihnen wahrscheinlich schon von der Maus belegt. Falls Sie jedoch eine andere Art der Steuerung des Amiga vorziehen, können Sie hier das entsprechende Gerät einsetzen.

Hinter den beiden Anschlüssen für Steuerungsgeräte befindet sich unter einer dünnen Plastikabdeckung noch die sogenannte Erweiterungsschnittstelle (engl. *Expansion Interface*). Über diese Schnittstelle wird der gesamte *System-Bus* nach außen geführt. D.h., daß Geräte die hier angeschlossen werden, Zugriff auf alle Bauteile des Systems (z.B. auf die Grafik-Chips oder die CPU selbst) haben. Auch Peripheriegeräte, die auf einen extrem schnellen Datenaustausch (über DMA) angewiesen sind, können hier angeschlossen werden. Eine solche Schnittstelle wird z.B. von Festplatten oder Speichererweiterungen benötigt. Falls Sie sich mit den "Innereien" von Computern nicht **sehr** gut auskennen, sollten sie Basteleien an dieser Schnittstelle unterlassen, da sie mit empfindlichen Teilen im Amiga verbunden ist.

Eine ähnliche Plastikabdeckung verdeckt auf der Vorderseite der Systemeinheit die letzte von außen zugängliche Schnittstelle des Amiga. Ihre einzige Anwendung ist im Moment die Aufnahme der Speichererweiterung, die den "kleinen" Amiga von 256 KByte Speicher auf 512 KByte anwachsen läßt. Möglicherweise wird Commodore oder andere Hersteller in Zukunft andere Speicherweiterungen für diese Schnittstelle anbieten, die noch mehr Speicher oder andere Erweiterungen enthalten. Dies ist im Moment aber nicht der Fall.

18.2 Die parallele Schnittstelle

Wie oben schon erwähnt wurde, ist die parallele Schnittstelle hauptsächlich für den Anschluß von Druckern gedacht. Sie ist weitgehend kompatibel zum weitverbreiteten CENTRONICS-Standard ausgelegt und ermöglicht so einen relativ mühelosen Anschluß einer Vielzahl von Druckern. Die folgende Liste gibt einen Überblick über die Drucker, die im Moment von der Software des Amiga unterstützt werden. Auch andere Drucker können angeschlossen werden, was aber immer mit etwas Schwierigkeiten (oder zusätzlicher Software) verbunden ist.

- Alpha P-101
- Brother 15 XL
- CBM MPS1000
- Diable 630
- Diable ADV D25
- Diablo C-150
- Epson FX/MX
- Epson JX-80
- HP LaserJet
- HP LaserJet Plus
- Okimate 20
- Qume LP 20

Die Anschlußbuchse der parallelen Schnittstelle hat das sogenannte DB-25-Format, enthält also 25 kleine Stifte, die in zwei Reihen zu je 12 und 13 Stiften angeordnet sind. Die folgende Abbildung zeigt das Aussehen der Anschlußbuchse der parallelen Schnittstelle im Amiga. Angedeutet wird auch die Numerierung für die einzelnen Stifte (Pins). Letztere gilt für die Aufsicht auf die Rückseite des Amiga — im Stecker eines Kabels muß man sie natürlich spiegelverkehrt zählen.

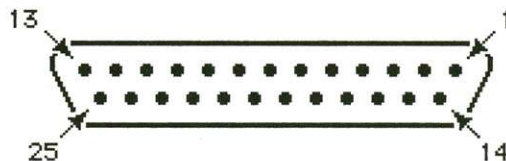


Bild 18 - 2: Die CENTRONICS-Schnittstelle des Amiga

Die Belegung der einzelnen Pins weicht allerdings in einigen Fällen von CENTRONICS-Standard ab, weshalb ein Kabel nicht einfach 1:1 Verbindungen zum Drucker hin durchziehen darf. Die folgende Tabelle gibt

einen Überblick über die Belegung der Pins (in der oben bereits angedeuteten Numerierung). Falls Ihnen die einzelnen Abkürzungen und Fachausdrücke in dieser Tabelle nichts sagen, sollten Sie auf keinen Fall versuchen, ein Kabel selbst zu löten (s.u.), sondern sich vertrauensvoll an Ihren Amiga-Händler wenden.

Pin-Nr.	Bezeichnung	Erläuterung
1	DRDY*	Data Ready
2	D0	Datenbit 0 (niederwertigstes)
3	D1	Datenbit 1
4	D2	Datenbit 2
5	D3	Datenbit 3
6	D4	Datenbit 4
7	D5	Datenbit 5
8	D6	Datenbit 6
9	D7	Datenbit 7 (höchstwertigstes)
10	ACK*	Acknowledge
11	BUSY	Busy
12	POUT	Paper Out
13	SEL	Select
14	GND	Signal Ground (Masse)
15	GND	"
16	GND	"
17	GND	"
18	GND	"
19	GND	"
20	GND	"
21	GND	"
22	GND	"
23	+5V	5 Volt Spannung (100 mA)
24	--	
25	RESET*	Reset

Tabelle 18 - 1: Pinbelegung der CENTRONICS-Schnittstelle ¹

Pin 23 führt eine konstante Spannung, die eventuell für die Stromversorgung eines kleinen Druckers oder anderen hier angeschlossenen Geräts verwendet werden kann. Falls ein Drucker aber nicht für eine solche Stromversorgung vom Computer aus gedacht ist, sollten diese Leitungen auf keinen Fall

¹ Mit einem Stern (*) versehene Pin-Namen bezeichnen Invertierte (Active Low) Signale. Dies gilt auch für alle folgenden Tabellen dieser Form.

verbunden werden. Wenn Sie mit den falschen Anschlüssen auf der Seite des Druckers verbunden werden, kann dadurch entweder der Drucker oder der Amiga schwer beschädigt werden.²

Falls Sie einen CENTRONICS-kompatiblen Drucker besitzen, so können Sie sich mit Hilfe der folgenden Tabelle ein Kabel selber löten (sofern Sie keine Angst haben, sich dabei buchstäblich oder im übertragenen Sinne die Finger zu verbrennen).

Pin (Amiga)	Pin (Drucker)
-------------	---------------

1...13	1...13
14	19
15...22	20...27
25	31

Tabelle 18 - 2: *Druckerkabel für CENTRONICS-kompatible Drucker*

Die hohen Nummer auf der rechten Seite der Tabelle kommen dadurch zustande, weil auf der Druckerseite normalerweise eine andere Steckerform verwendet wird. Lassen Sie sich dadurch nicht verwirren — in jedem Stecker sind bei genauem Hinsehen kleine Zahlen zu finden, die die Numerierung klar stellen.

18.3 Die serielle Schnittstelle

Auch die serielle Schnittstelle des Amiga verwendet das DB-25-Format. Statt Stiften enthält sie aber kleine Buchsen, damit man nicht aus Versehen ein paralleles Kabel hier einsteckt. Ihr Aussehen und die Nummerierung der einzelnen Pins ist also gleich wie bei der parallelen Schnittstelle. Eine zusätzliche Abbildung ersparen wir uns deshalb.

Die serielle Schnittstelle des Amiga entspricht (wieder bis auf einige Ausnahmen) dem RS232-Standard. Falls Sie bereits einen Drucker besitzen, der diesem Standard entspricht, können Sie ihn auch hier anschließen. Falls Sie sich erst einen Drucker zulegen, sollten Sie ihn aber lieber in

² Dies ist keine leere Drohung – genau das ist mir selbst bei den ersten Experimenten mit dem Amiga passiert!

CENTRONICS-kompatibler Form kaufen, da dann die serielle Schnittstelle für andere Zwecke frei bleibt.

Pin-Nr.	Bezeichnung	Erläuterung
1	GND	Frame Ground
2	TXD	Transmit Data
3	RXD	Receive Data
4	RTS	Request To Send
5	CTS	Clear To Send
6	DSR	Data Set Ready
7	GND	System Ground
8	CD	Carrier Detect
9	--	
10	--	
11	--	
12	--	
13	--	
14	- 5V	5 Spannung (50 mA)
15	AUDO	Audio Out
16	AUDI	Audio In
17	EB	Takt
18	INT2*	Interrupt zum Amiga
19	--	
20	DTR	Data Terminal Ready
21	+ 5V	5 Volt Spannung (100 mA)
22	--	
23	+12V	12 Volt Spannung (50mA)
24	C2*	3,58 MHz Takt
25	RESB*	Reset zum Amiga

Tabelle 18 - 3: Pinbelegung der seriellen Schnittstelle

Wie auch schon bei der CENTRONICS-Schnittstelle, dürfen die stromführenden Pins auf keinen Fall mit einem Gerät verbunden werden, sofern dies nicht explizit in der Gerätebeschreibung angegeben wird.

Wie aus den Namen der einzelnen Pins ab etwa Nummer 15 zu sehen ist, dient diese Schnittstelle nicht nur der seriellen Verbindung zu Peripheriegeräten, sondern erlaubt auch den Zugriff auf andere interne Signale. (So kann z.B. ein Interrupt oder ein Reset-Signal zur CPU geschickt werden.) Diese dürften aber nur in den seltensten Fällen benötigt werden. Falls Sie selbst Kabel löten, sollten Sie die entsprechenden Steckeranschlüsse unverbunden lassen.

Ein serielles Kabel benötigt im allgemeinen wesentlich weniger funktionelle (verbundene) Leitungen als ein paralleles. Für den Anschluß eines RS232-kompatiblen Druckers über die serielle Schnittstelle gibt z.B. die folgende Tabelle eine meist völlig ausreichende Verbindung an.

Pin (Amiga):	Pin (Drucker):
--------------	----------------

1	1
2	3
3	2
7	7

Tabelle 18 - 4: *Druckerkabel für RS232-kompatible Drucker*

Denken Sie aber nach dem Anschluß eines solchen Druckers daran, auch mit dem Werkzeug **Preferences** den Drucker auf die serielle Schnittstelle "umzulegen" und die korrekte Baud-Rate einzustellen (üblicherweise 1200 oder 9600).

18.4 Die Stereo-Anschlüsse

Die beiden Ton-Ausgänge am Amiga sind am problemlosesten zu verkabeln. Kabel, mit denen Sie über diese Buchsen den Amiga an Ihre Stereo-Anlage anschließen können, erhalten Sie in jeder besseren Elektrohandlung, die auch Hifi-Geräte führt.

Sie benötigen zwei einfache zweiadrige Kabel mit *Cinch*-Steckern (sprich Zinsch) an beiden Enden, falls Ihre Stereoanlage ebenfalls Cinch-Buchsen besitzt oder ein Kabel "von Doppel-Cinch auf DIN", falls sie nur DIN-Eingänge besitzt. Wenn Sie nur ein wenig Erfahrung mit Hifi-Geräten haben können Sie diese auch sehr leicht selbst löten.

Eine sehr gut klingende Kombination ist auch der Anschluß zweier kleiner Aktiv-Boxen, wie es sie z.B. für den Anschluß an Walkman-Kassettenrecorder gibt. Sie brauchen Ihre Heim-Stereo-Anlage dann nicht für Computer-Musik zu mißbrauchen und die anderen Familienmitglieder vom Musikgenuß abhalten.

Die meisten Monitore für den Amiga besitzen ebenfalls Lautsprecher. Falls Sie nicht unbedingt den vollen Klang einer Stereo-Anlage nutzen wollen, können Sie die Ton-Ausgänge auch an den Lautsprecher anschließen. Bei

einfacheren Video-Monitoren wird der Ton bereits über das Video-Kabel an den Bildschirm übergeben. Verwenden Sie jedoch einen der beiden speziell für den Amiga angebotenen RGB-Monitore, müssen Sie eine zusätzliche Verbindung ziehen. Auch hierfür benötigen Sie wieder Kabel mit Cinch-Steckern an beiden Enden. Der entsprechende Eingang am Monitor ist immer mit dem Wort **Audio In** markiert.

Ein Problem ist dabei allerdings, daß Monitore üblicherweise nur einen Lautsprecher besitzen. Die beiden Lautsprecher-Ausgänge des Amiga müssen deshalb mit nur einem Eingang im Monitor verbunden werden. Hierzu werden jeweils zwei Adern der beiden Kabel, die aus dem Amiga kommen, zusammengelegt (-gelötet) und als eine Ader zum Monitor weitergeführt. Dies ist — im Gegensatz zu dem, was Ihnen vielleicht einige Hifi-Freaks erzählen werden — beim Amiga völlig harmlos und schadet dem Gerät nicht. Entsprechende sogenannte *T-Adapter* können Sie wieder im gut bestückten Audio-Fachhandel erhalten oder sich (sofern Sie die passenden Stecker finden) auch selbst löten.

18.5 Die Video-Anschlüsse

Zum Anschluß eines Bildschirms bietet der Amiga eine Vielzahl von Möglichkeiten. Sie können sowohl einen hochwertigen RGB-Monitor, einen Video-Monitor, wie er z.B. für Video-Kameras oder -Recorder verwendet wird, oder auch einen einfachen Farbfernseher verwenden. Dieser muß im Moment aber NTSC-kompatibel (also für die amerikanische Farbfernseh-Norm ausgelegt) sein; PAL-Versionen des Amiga werden im Sommer 1986 erwartet. Eine genaue Erläuterung dieser letzten Schnittstelle ersparen wir uns.³

³ Hier genügt für eine Verbindung zum Monitor auch ein einfaches zweiadriges Kabel mit Cinch-Steckern, weshalb sowieso nicht viel zu erläutern ist.

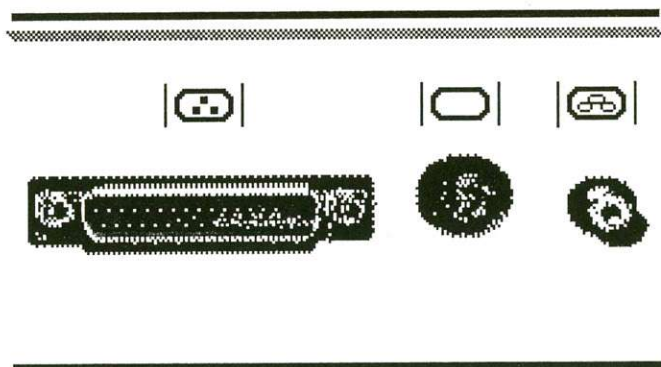


Bild 18 - 3: Die Video-Schnittstellen des Amiga

Die besten Ergebnisse (und die wenigsten Kopfschmerzen) können sie zweifellos mit einem RGB-Monitor erreichen, der über den flachen Stecker links angeschlossen wird. Es handelt sich um einen DB-23-Stecker und damit um ein sehr exotisches Format, das fast nur im Elektronik-Versandhandel zu bekommen ist.

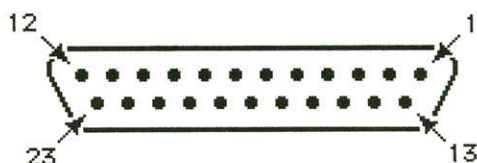


Bild 18 - 4: Ein DB-23-Stecker

An der RGB-Schnittstelle, die über diesen Stecker hinausgeführt wird, finden Sie alle Signale, die üblicherweise für den Betrieb eines Monitors benötigt werden. Sowohl digitale RGB-Monitore, als auch analoge RGB-Monitore und Composit-Farb-Monitore können über diese Schnittstelle versorgt werden. Die brilliantesten Farben erhalten Sie bei Verwendung eines analogen RGB-Monitors.

Die RGB-Schnittstelle führt sowohl die analogen RGB-Signale als auch die digitalen (wie sie im Kapitel *Grafik* erläutert wurden) nach außen. Über die digitalen Signale können auch Monitore angeschlossen werden, die zur Farbgrafik-Karte des IBM-PC kompatibel sind. Von diesen gibt es eine reichhaltige Auswahl zu recht günstigen Preisen. Mit einem solchen digitalen RGB-Monitor können jedoch nur maximal 15 verschiedene Farben der 4096, die der Amiga beherrscht, sichtbar gemacht werden.

Pin-Nr.	Bezeichnung	Erläuterung
1	XCLK*	External Clock
2	XCLKEN*	External Clock Enable
3	RED	Analoger Rot-Wert
4	GREEN	Analoger Grün-Wert
5	BLUE	Analoger Blau-Wert
6	DI	Digitale Intensität
7	DB	Digital Blau An
8	DG	Digital Grün An
9	DR	Digital Rot An
10	CSYNC*	Composite-Sync-Signal
11	HSYNC*	Horizontales Sync-Signal
12	VSNC*	Vertikales Sync-Signal
13	GNDRTN	Masse für XCLKEN*
14	ZD*	Zero Detect
15	C1*	Clock Out
16	GND	Masse
17	GND	Masse
18	GND	Masse
19	GND	Masse
20	GND	Masse
21	- 5V	5 Volt Spannung (50 mA)
22	+12V	12 Volt Spannung (175 mA)
23	+ 5V	5 Volt Spannung (300 mA)

Tabelle 18 - 5: Pinbelegung des RGB-Anschluß

Bei Verwendung eines der beiden Amiga-Monitore von Commodore werden Sie keinerlei Schwierigkeiten mit dem Anschluß haben — passende Kabel liegen bei. Bei anderen Monitoren kann Ihnen die obige Tabelle vielleicht helfen, sich ein Kabel selbst zu löten oder von einem Fachmann löten zu lassen. Falls Sie mit dem Gedanken an einen anderen Monitor spielen, sollte dieser aber gewisse Mindestanforderungen erfüllen. Diese sind:

- RGB-Analog-Signal 0,7 Volt; 75 Ohm Abschluß
- Video-Bandweite größer als 15 MHz
- Horizontale (Zeilen-) Frequenz 15,7 kHz
- Punktgröße 0,43 oder kleiner

Im Zweifelsfall ist eine höhere Bandbreite und eine kleinere Punktgröße immer besser!

Die folgende Tabelle listet z.B. die Verbindungen für ein Kabel auf, das mit Erfolg bei einem hochwertigen Video-Monitor einer anderen Marke, dem Sony KV 1311CR, verwendet wurde.

Pin (Amiga):	Pin (Monitor):
3	25
4	26
5	27
10	30
16...19	8...11
23	29 & 33 (über einen 75 Ohm Widerstand)

Tabelle 18 - 6: Videokabel für Sony-Monitor

Die zweite Video-Schnittstelle liefert ein sogenanntes Composite-Video-Signal, wie es üblicherweise für die Verbindung zwischen Video-Geräten (Recordern, Kameras und Monitoren) verwendet wird. Es handelt sich um eine 8-polige DIN-Buchse, deren Belegung Sie in der folgenden Tabelle aufgelistet finden.

Pin-Nr.:	Bezeichnung	Erläuterung
1	--	
2	GND	Masse
3	AUDIO LEFT	Linker Tonkanal
4	COMP VIDEO	Composite Video
5	GND	Masse
6	--	
7	+ 12V	12 Volt Spannung (60 mA)
8	AUDIO RIGHT	Rechter Tonkanal

Tabelle 18 - 7: Pinbelegung des Composite-Video-Anschluß

Sich ein Kabel für diese Schnittstelle selbst zu löten, ist allerdings nur in den wenigsten Fällen nötig. Sie sind recht preiswert im Video-Fachhandel erhältlich.

18.6 Die Steuer-Schnittstellen

Der im Titel dieses Abschnitts verwendete Name für die beiden kleinen Buchsen an der rechten Seite der Amiga-Systemeinheit bemäntelt etwas ihre wahre Funktion. Die Amiga-Dokumentation ist hier etwas deutlicher. Darin heißen die beiden Schnittstellen *Game-Ports*. Hier werden üblicherweise außer der Maus nur Steuerknüppel oder Rollkugeln für Video-Spiele angeschlossen. Das Format der beiden Steckdosen heißt DB-9. Entsprechende Stecker sind problemlos in vielen Elektronik-Bastler-Geschäften zu bekommen.

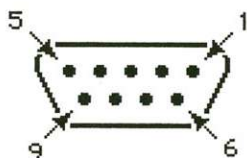


Bild 18 - 5: Eine Steuer-Schnittstelle

Beide Game-Ports sind im wesentlichen gleich belegt. Die Namen der Pins und Ihre Funktionen wechseln jedoch, je nachdem, welche Steuergeräte daran angeschlossen sind. Die folgenden Tabellen geben einen Überblick über die verschiedenen Belegungen. Dazu muß aber noch bemerkt werden, daß nur der erste (vorderste) Game-Port einen Lichtgriffel akzeptiert. Mäuse, Steuerknüppel und Rollkugeln können in beide Buchsen gesteckt werden.

Pin-Nr.:	Bezeichnung	Erläuterung
1	MOUSE V	Vertikaler Zähler
2	MOUSE H	Horizontaler Zähler
3	MOUSE VQ	Vertikale Quadratur
4	MOUSE HQ	Horizontale Quadratur
5	MOUSE BUTTON 2	Rechte Maustaste
6	MOUSE BUTTON 1	Linke Maustaste
7	+ 5V	5 Volt Spannung (125 mA)
8	GND	Masse
9	MOUSE BUTTON 3	Dritte Maustaste (nicht vorhanden)

Tabelle 18 - 8: Pinbelegung für den Maus-Anschluß ⁴

Wie die Tabelle zeigt, akzeptiert der Amiga auch Mäuse mit drei Tasten. Es gibt im Moment jedoch kein Programm, das einen Druck auf diese dritte Taste erkennt und in irgendeiner Form darauf reagiert.

Pin-Nr.:	Bezeichnung	Erläuterung
1	FORWARD*	Schalter Vorwärts
2	BACK*	Schalter Zurück
3	LEFT*	Schalter nach links
4	RIGHT*	Schalter nach rechts
5	POT X	Horizontales Potentiometer
6	FIRE*	Feuer-Taste
7	+5V	5 Volt Spannung (125 mA)
8	GND	Masse
9	POT Y	Vertikales Potentiometer

Tabelle 18 - 9: Pinbelegung für Steuerknüppel und Rollkugeln

⁴ Die Zählsignale werden **zusammen** mit den Quadratur-Signalen verwendet, um nicht nur die von der Maus zurückgelegte Strecke, sondern auch die Richtung festzustellen.

Pin-Nr.:	Bezeichnung	Erläuterung
1	--	
2	--	
3	--	
4	--	
5	LIGHT PEN PRESS	Stift berührt Bildschirm
6	LIGHT PEN*	Strahl gefunden
7	+5V	5 Volt Spannung (125 mA)
8	GND	Masse
9	--	

Tabelle 18 - 10: *Pinbelegung für Lichtgriffel*

Welches Gerät an welche Schnittstelle angeschlossen ist, bemerkt zum Teil die Systemsoftware. Im wesentlichen ist dafür aber das gerade laufende Programm zuständig. Deshalb dürften andere Steuergeräte als die Maus nur in wenigen dafür ausgelegten Programmen funktionieren. In der Workbench kann z.B. ein Joystick oder ein Lichtgriffel nicht eine Maus am vordersten Anschluß ersetzen.

Gerade durch die zusätzliche Stromversorgung bieten sich die beiden Game-Ports allerdings auch für den Anschluß (exotischer) anderer Geräte an (die man vielleicht sogar selbst gebaut hat). Wie die von diesen gelieferten Werte von einem Programm aus ermittelt werden können, ist im *Hardware Manual* des Amiga erläutert.

Stichwortverzeichnis

; 260
< 261
> 261

Abbruch-Level ändern 277
Absatz 201
Abspeichern 101
Adreß-Generator 369
Agnus 369
Aktions-Knopf 48
Alarm Off 96
Alarm On 96
Alarmzeit einstellen 97
Alerts 146
Alphanumerisches Terminal 37
Amiga-Basic 421
— Umgang 427
Amiga-Fähigkeiten 429
Amiga-Grafik, Anwendungen 391
Amiga-Software 407
Amiga-Tasten 22, 127
AmigaDOS 359
AmigaDraw 220
Analoguhr 96
Ändern von Text 326
AnimObjects 32, 388
Anweisungs-Blocks 425
Anwendungsprogramm 93, 151
Aspect 119
Assembler 416
Aufforderung, Diskette einlegen 42
Aufgaben 107
Auflösung 30
Aufräumen einer Diskette 73
Ausdruck
— Negativ 119
— Positiv 119
Ausschneiden 157, 198
Ausschnitt vergrößern 159
Auswählen mit der Maus 51
Auswählen von Disketten 68
Auto-Size 103
Automatische Größe 103
Automatisierung 333

Back-Gadget 58, 134, 147
BACKSPACE-Taste 74
Basic 417, 422
— Moderne Features 425
Basispaket 20
Baud Rate 114
Bedienungselement 138
Bedingte Anweisungen 340
Bedingte Ausführung 280
Bedingung EXISTS 280, 342
Bedingungen 280
Befehl 129
— Analog 96
— Bend 187
— Change Palette 161
— Clean Up 54, 73, 85
— Close 59, 70, 79
— Copy 157, 198
— Custom Brush 165
— Cut 157, 198
— Cycle Colors 166
— Cycle Draw 167
— Digital 96
— Discard 87, 90
— Duplicate 62, 83
— Empty Trash 87
— Erase 158
— Erase Screen 158
— Find and Replace Text 206
— Frame 157
— Info 80, 90, 97
— Initialize 77
— Keyboard 210, 214
— Last Error 54
— Magnify 158
— Mirror 160
— New 101
— Open 55, 69, 79, 90, 94, 102, 105
— Open Document 199
— Paste 157, 198
— Print 102, 156
— Print Document 199
— Redraw 54
— Rename 73, 80, 100, 143
— Save 101
— Save As 102
— Save As Brushes 156
— Save As Painting 156
— Save Document 199
— Set 96
— Set Ruler Margins 201

- Snapshot 68, 85, 90, 144
- Staff 210
- Symmetry Center 185
- Synthesizer 210
- Text 164
- Undo 157
- Version 54
- Befehl eingeben 227
- Befehl mit der Maus 51
- Befehl umleiten 261
- Befehle der Workbench 91
- Befehle überspringen 300
- Befehlsliste 153
- Befehlszeilen verketten 233
- Benutzerschnittstelle 121
- Bewegen von Fenstern 56
- Bewegte Bilder 368
- Bewegung der Maus 126
- Bewegung des Cursors 317
- Bibliotheken 410
- Bildausschnitt verschieben 137
- Bildschirm
 - -Attribute 110
 - Bild zentrieren 114
 - -größe 114
 - Physikalischer 111
 - Virtueller 111
- Bildteile löschen 177
- Bitmapped 30
- Bitmapped Grafik 375
- Bleistift 197
- Blitter 369
- Blitter Objects 32, 387
- BOB 32, 387, 432
- BOBs 387
- Bold 104
- Boot-Diskette 28
- Booten 45
- Bouncing ball 29
- Brush handle 190
- Bytes per Block 76

- C 417
- CAD 368
- Calculator 99
- Cancel 63, 84, 112
- CENTRONICS-Schnittstelle 440
- Change current directory 241
- Change Printer 112, 117
- Chips 359
 - Funktionsweise 361
- CLI 37, 114, 223

- Einschalten 225
- Kommandos 255
- Sichtbarmachen 226
- Verlassen 229
- CLI-Befehl
 - Assign 262
 - Cd 241, 263
 - Copy 264
 - Date 266
 - Delete 267
 - Dir 268
 - Diskcopy 270
 - Echo 272
 - Ed 273
 - Edit 274
 - Else 340
 - Endcli 275
 - Endif 341
 - Execute 276, 333
 - Failat 277
 - Filenote 278
 - Format 279
 - If 280, 340
 - Info 281
 - Install 282
 - Join 283
 - Lab 284
 - List 285
 - Loadwb 287
 - Makedir 288
 - Newcli 232, 289
 - Prompt 290
 - Protect 291
 - Quit 292
 - Relabel 293
 - Rename 294
 - Run 233, 295
 - Say 297
 - Search 298
 - Skip 300
 - Sort 301
 - Stack 302
 - Type 303
 - Wait 304
 - Why 305
- CLI-Befehle eingeben 230
- CLI-Disketten 354
- CLI-Fenster 227
- CLI-Kommandos, Format 256
- CLI-Task beenden 275
- CLI-Task erzeugen 289
- CLI-Tasks 232

Clock 89, 95
 Color 118
 Color Control 185
 Color Palette 173
 Command Line Interface 37
 Command Line Interpreter 223
 COMMENT 81
 Compiler 414
 Composite-Video-Anschluß 448
 CON: 248
 Continue 63
 Coordinates 190
 Copper 371
 copy of 65, 82, 90
 CPU 25, 360
 Cursor 74, 315
 Cursor-Tasten 317
 Custom-VLSI-Chips 30

 Dateien 36, 78, 235, 355
 Dateien anlegen 273
 Dateien kopieren 264
 Dateien löschen 267
 Dateien umbenennen 294
 Dateien verbinden 283
 Dateienbaum 243
 Dateinamensmuster 306
 Dateiverwaltung 426
 Dateiverzeichnis
 — ändern 263
 — anlegen 288
 — anzeigen 268
 — Aktuelles 241
 — Wurzel 243
 — RAM: 247
 Dateiverzeichnisse 78, 235
 Datentypen 425
 Datum ändern 266
 Datum einstellen 114
 DEFAULT TOOL 106
 Default Tool 76
 DELETEABLE 81
 Deluxe Music Construction Set 220
 Deluxe Paint 168
 Deluxe Video Construction Set 220
 Demos 109
 Denise 366
 Deselektieren 51
 DESTINATION Disk 63
 DEVS: 252
 Digitaluhr 96
 Direct Memory Access 362

Discard 84
 Diskette
 — 3,5-Zoll 61
 — Aufräumen 73
 — Beschriften 62
 — Informationen 75
 — Initialisieren 77
 — Löschen 77
 — Umbenennen 73
 — Formatieren 279
 Disketten 67
 — 3,5-Zoll- 43
 — Als Schubladen 86
 — Auswählen 68
 — Füllanzeige 70
 — Kopieren 270
 — Verschieben 68
 — Umbenennen 293
 Disketten-Icon 68, 86
 Disketten-Icons 44
 Diskettenanschluß 27
 Diskettenfenster 69
 Diskettenlampe 43, 63
 Diskettenlampe 63
 Diskettenlaufwerk 24, 247
 DMA 362, 439
 DMA-Kanäle 369
 Doppelklick 69, 115, 125
 Doppelklick-Zeitspanne 116
 DPaint 169
 — Grundzüge 170
 — Menüs 184
 — Starten 169
 — Tastenbefehle 192
 Draft 103
 Drucken 102, 186
 Drucker 248, 352, 440
 Druckeranpassung 117
 Dual-Playfield-Modus 382

 Ed 313
 — Aufrufen 314
 — Tricks 328
 — Verlassen 314
 Ed-Befehle
 — Direkte 316
 — Escape 323
 — Liste 330
 Ed-Fenster 315
 Edit Pointer 112, 117, 124
 Editor 273

Eigenschaften von Amiga-Programmen 122

Ein-/Ausgabe-Umleitung 253

Einfügen 198

Einfügen von Text 327

Einlesen 184

Einschaltvorgang 41

Einsetzen 157

Ellipsen zeichnen 179

Enable 220

Entwurf 103

Ereignisse abfangen 432

Ersetzen 206

Erweiterungsschnittstelle 27, 439

Escape-Befehle 323

— Globale 324

Exec 408

Externes Laufwerk 64

Extras-Diskette 28

Farb-Kontrolle 185

Farbdrucker 118, 156

Farbe 367

Farbe mischen 162

Farbeimer 179

Farben 172, 376

Farben einstellen 113

Farben schattieren 189

Farben verschmieren 189

Farbenrotation 165

Farbkomponente 377

Farbmischung 113

Farbmonitor

Farbpaletten-Animation 165, 190

Farbregister 377

Farbrolle 203

Farbton 174

Farbtopf 162

Farbwechsel 166

Fast Feedback 190

Fehler 277

Fehlersuche 428

Fenster 56, 133, 137, 392, 430

— Aktiv 137

— Auffrischen 392

— Bewegen 56

— Disk Copy 63

— Einer Schublade 80

— Eines Werkzeugs 95

— Inaktiv 137

— Nach hinten legen 59

— Nach vorne legen 59

— Philosophie 133

— Titel-Leiste 134

— Überlappende 58

Fenster-Gadgets 58

Fenster-Titel 134

Festplattenlaufwerk 247

Financial Cookbook 220

Finden 206

Flächen 179

Flächen füllen 164

FONTS: 252

Footprints 165

Formatieren des Textes 201

FrameGrabber 391

Front-Gadget 58, 134, 147

Füllanzeige 70

Fußzeile 204

Gadget 58, 138

— Kombinierte 142

— Sensoren- 139

— Tasten- 139

— Text- 141

Game-Ports 449

Gemälde drucken 156

GenLock-Interface 390

Geräte 238, 410

Geräte-Treiber 252

Gerätename 238

Gerätenamen, Gebrauch 246

Geschwindigkeit der Tasten-Wiederholung
115

Gleichzeitigkeit 107

Grafik 367, 375, 430

Grafik-Anpassung 118

Grafik-Bibliothek 411

Grafik-Fähigkeiten 30

Grafik-Hardware 34, 390

Grafik-Modi 32

Grafikauflösung 367

Grafikfähiger Drucker 103

Graphic 103

Graphic Select 118

Graphicraft 151

— Grundzüge 152

— Menüs 154

Graustufung 118

Gray Scale 118

Griff des Reglers 140

Größe des Ausdrucks 103

Größe eines Zeichensatzes 104

Größen-Gadget 70, 146

- Gummikugel 48
- Häkchen 131
- HAM 379
- HAM-Modus 30
- Hardware-Sprites 384
- Haupt-Menüs 130
- Haupteinheit 21
- Heißer Punkt 117
- Helligkeit der Farbe 174
- Herunterzieh-Menü 127
- Hervorhebung 200
- Hervorhebung eines Menü-Punktes 129
- Hilfestellung 195
- Hintergrund-Befehle 233
- Hintergrundfarbe 163, 172
- Hierarchisches Dateisystem 237
- Historie 38
- Hold and Modify 30
- Hold-And-Modify-Modus 379
- Hüllkurve 33, 216, 397
- Hüllkurve eines Tons 400
- Icon 44, 52
- Icon editieren 357
- IconEd 356
- Icons 67, 88, 223, 355
- Image 119
- Info release 1.1 75
- Info-Dateien 355
- Info-Fenster 76
 - Einer Schublade 81
 - Eines Werkzeugs 97
 - Eines Projekts 106
- Informationen über eine Datei 285
- Informationen über eine Diskette 75
- Initialisieren einer Diskette 77
- Instrument, Charakteristika 218
- Interaktive Auflistung 268
- Interlace-Modus 380
- Internes Laufwerk 64
- Interpreter 414
- Interrupts 364
- Intuition 121, 412
- Italic 104
- Kamera 198
- Key Repeat Delay 115
- Key Repeat Speed 115
- Kickstart 26
- Kickstart-Diskette 28, 42
- Klang digitalisieren 398
- Klangerzeugung 365, 397
- Klangerzeugung mit Paula 401
- Klebstoff-Tube 198
- Klicken 51
- Kombinierte Gadgets 142
- Kommando-Zeile 323
- Kommandofolge 333
- Kommandofolge beenden 292
- Kommandofolgen 252
- Kommandofolgen mit Parametern 336
- Kommandos 354
- Kommentar 260
- Kommentar für Datei 278
- Kommunikationsprogramm 221
- Komponenten 21
- Kompositionshilfe 209
- Kompositionsteil 211
- Konsole 248
- Kontroll-Leiste 170
- Koordinaten 190
- Kopfzeile 204
- Kopien 60
- Kopieren 157, 198
 - Mit einem Diskettenlaufwerk 62
 - Mit zwei Diskettenlaufwerken 64
 - Von Dateien 264
- Kopiervorgang, Abschluß 64
- Koprozessoren 361
- Kreise zeichnen 179
- Kursiv 104
- Kurven 177
- Large 103
- Leertaste 22
- Lichtgriffel 451
- Lineal 163
- Linien 177
- Linke Maustaste 69
- Lisp 418
- Listen 408
- LoadWB 348
- Logische Geräte 239, 250
- Logische Geräte ändern 262
- Logo 418
- Löschen 158
 - Einer Diskette 77
 - Schubladen 83
 - Von Dateien 267
 - Von Text 321, 327
- Lupe 182
- M68000 25, 360

- Malprogramm 151, 169
- Marke setzen 284
- Markierte Menü-Punkte 131
- Maschinencode 416
- Maus 21, 23, 46, 122
 - Funktionsprinzip 123
 - Funktionsweise 47
 - Kabelanschluß 50, 123
 - Knöpfe 50
 - Schlafende 125
 - Unterseite 48
- Maus-Anschluß 450
- Maus-Spiele 49
- Mausbewegungen 68, 431
 - Übersetzung 115
- Mausklick 51, 125, 431
 - Mit der Tastatur 127
- Maustaste 68
- Maustasten 48, 125
- Mauszeiger 49, 53, 123, 138
 - Ändern 117
 - Heißer Punkt 117, 124
- Medium 103
- Mehrere Aufgaben 107
- Menü
 - Alarm 96
 - Brush 155, 165, 186
 - Color 155, 161
 - Custom 155
 - Disk 77, 87, 91
 - Edit 154, 157
 - Font 104, 181, 189
 - Format 104
 - Mode 96
 - Modes 188
 - Picture 177, 184
 - Prefs 189
 - Project 101, 154
 - Screens 210
 - Shape 155, 163
 - Special 54, 68, 85, 91, 154, 158
 - Style 104
 - Styles 200
 - Tastatur-Äquivalente 132
 - Type 96
 - Workbench 73, 84, 91
- Menü-Auswahl 129, 130
- Menü-Befehl erteilen 54
- Menü-Knopf 48, 126
- Menü-Leiste eines Werkzeugs 96
- Menü-Punkt 128
 - Abgehakt 132
 - Hervorgehoben 129
 - Markiert 131
- Menü-Taste 52, 126
- Menü-Titel 96
- Menüs 52, 53, 127, 431
- Menüs aufrufen 53
- Menus-Übersetzung 116
- Menütitel 52
- Microsoft Basic 424
- MIDI 405
- MIDI-Gerät 33
- MIDI-Interface 405
- MODEM 27
- Modula 419
- Monitor 24
- Mülleimer 84, 87
- Mülleimer-Icon 87
- Menü-Punkt 128
- Multitasking 37, 107
 - Probleme 109
 - Vorteile 108
- Musicraft 209
 - Aufbau 210
- Musikausgabe 433
- Muster verwenden 306
- Nachrichten 409
- Name des Werkzeugs 106
- Namenskonventionen 245
- Neues Projekt 101
- Neustart 45
 - Tastenkombination 46
- NIL: 247
- NOT DELETEABLE 81
- Notenblatt 210
- Notepad 100
 - Möglichkeiten 104
- Notizblock 99
- NTSC-Norm 24
- Number Free 76
- Number of Blocks 76
- Number Used 76
- Objekte 99
- Objekte der Workbench 67
- Öffne Diskette 55
- Öffnen eines Projekts 105
- Öffnen eines Werkzeugs 94
- Open 55
- Orientierung des Ausdrucks 119
- Original-Diskette 61

- PAL-Amiga 389
- PAL-Norm 24
- Palette ändern 161
- Paper-Color 104
- Papier-Farbe 104
- PAR: 248
- Parallele Schnittstelle 248, 440
- Parameter 257, 336
- Pascal 419
- Paula 364
- Klangerzeugung 401
- Pen-Color 104
- Pfeil 49
- Phoneme 404
- Physikalische Geräte 238, 246
- Physikalischer Bildschirm 111
- Pinselformen 156, 165, 175
- Bearbeiten 186
- Drehen 187
- Selbstdefinierte 186
- Verzerren 187
- Pinselformen 156, 165, 175
- Selbstdefinierte 176
- Vordefinierte 175
- Pinselformen 156, 165, 175
- Pinselformen 156, 165, 175
- Pinselformen 156, 165, 175
- Pixel 376
- Plain 104
- Platzhalter 306, 336
- Playfield-Scrolling 368
- Polygone zeichnen 179
- Polyscope-Diskette 29
- Preferences 89, 112, 140
- Hauptfenster 112
- Print As 103
- Prioritäten 364
- Programm
- Boxes 109
- Dots 109
- Lines 109
- Programm-Chaining 426
- Programm-Marken 425
- Programme 36, 89
- Programme eingeben 427
- Programmiersprachen 413
- Programmierung 407
- Grundlagen 413
- Projekt öffnen 105
- Projekte 89, 93, 99
- Prompt 227
- Prozess 37
- Prozessor 360
- PRT: 248
- PullDown-Menü 127
- Punkt setzen 124
- RAM-Disk 349
- Probleme 351
- RAM-Speicher 25, 109
- RAM: 247
- Raster-Schrittweite 181
- RAW: 248
- Read Only 76
- Read/Write 76
- Rebooten 45
- Rechteck 163
- Rechteck zeichnen 179
- Requester 143
- Besonderheiten 145
- Reset Colors 113
- Retry 69, 84
- RETURN-Taste 74
- RGB-Anschluß 447
- RGB-Monitor 25
- RGB-Regler 174
- Robocity-Demo 32
- Rollbalken 71, 136
- Rollkasten 72, 137
- Rollkugel 450
- Rollpfeil 72, 137
- Sattheit der Farbe 174
- Schale 36
- Schalter 131
- Schattierung 118
- Schere 198
- Schieberegler 115, 140
- Schieß-Gadget 59, 70, 134
- Schlüsselwort
- ALL 264, 267
- KEYS 285
- NODATES 285
- OPT 268
- QUICK 285
- QUIET 264, 267
- SINCE 285
- UPTO 285
- Schlüsselworte 257
- Schnappschüsse 85
- Schnelle Rückkopplung 190
- Schnittstellen 27, 437
- Schreibmarke 74, 101, 141, 315
- Schreibschutz 61

- Schreibschutz-Schieber 61
- Schublade
 - Demos 58, 109
 - Utilities 99, 106, 355
- Schubladen 36, 78
 - Aufgabe 78
 - Erzeugen neuer 83
 - Löschen 83
 - Umgang 79
- Schubladenfenster 80
- Schwarz-Weiß-Grenze 119
- ScoreWriter 221
- Screens 110, 146, 394, 430
 - Bedienung 147
 - Mehrere 147
- Seitenformat 203
- Seitengröße 204
- Seitenzahl 204
- Selektieren mit der Maus 51
- Selektions-Knopf 48
- Selektions-Taste 130
- Sensoren 142
- Sensoren-Gadget 139
- SER: 248
- Serielle Schnittstelle 27, 114, 248, 442
- Set Point 117, 124
- Shade 118
- Shell 36, 224
- Sicherungs-Schalter ändern 291
- Sicherungskopien 60, 270
- Sichtbarer Ausschnitt 136
- Simple Refresh 392
- Small 103
- Smart Refresh 392
- Software 35, 407
- Software-Sprites 387
- Sonderzeichen 197
- Sound-Fähigkeiten 33
- Sound-Hardware 34
- SOURCE Disk 63
- Speicherbedarf 109
- Speicherverwaltung 409
- Spezialeffekte 403
- Spiegel 160
- Spiegelachse 160
- Spiegelachsen 182
- Spiele 221
- Sprach-Fähigkeiten 34
- Sprachausgabe 433
- Sprachsynthese 403
- Spraydose 179
- Sprite-Kanal 386
- Sprites 367, 383, 432
- Sprunganweisung 300
- Sprünge 340
- Stack-Größe ändern 302
- StartDisk 251
- Startdiskette 106, 239
- Startdiskette maßschneidern 352
- Startfähige Diskette 282
- Startup-Sequence 347
- Status 76
- STATUS 81
- Stereo-Anschlüsse 444
- Steuer-Schnittstellen 449
- Steuerknüppel 450
- Stift-Farbe 104
- Stil 104
- Sub-Menü 102
- Sub-Menüs 130
- Subprogramme 426
- Suchen von Text 326
- Super Bitmap 392
- Synthesizer 209, 215
- SYS: 106, 239
- System-Alert 143
- Systemprogramm 93
- Systemsoftware 121, 408
- Takt 213
- Taktfrequenz 363
- Taschenrechner 99
- Task 37
- Task-Verwaltung 409
- Tasks 409
- Tastatur 21, 214
- Tastatur-Äquivalente 132
- Tastatur-Maus 126
- Tastaturabkürzungen 191, 207
- Tasten-Gadget 139
- Tasten-Wiederholung 115
- Telecraft 221
- Text
 - ändern 197
 - ausgeben 272, 297
 - ausgeben 297
 - drucken 199
 - einfügen 320
 - eingeben 196
 - formatieren 201
 - in ein Bild einfügen 180
 - löschen 321
 - suchen 298
- Text-Blocks 325

- Text-Editor ed 313
- Text-Formulare 205
- Text-Gadget 141
- Text-Stil ändern 200
- Textcraft 193
 - Funktion des Zahlenfeldes 209
 - Projekte 199
 - Überblick 194
- Textdatei ausgeben 303
- Textdatei sortieren 301
- Texteditor 274
- Texteingabe 74
- Textpassagen finden 206
- Textverarbeitung 100
- Textverarbeitungssystem 193
- Titel-Leiste 57, 111, 126
 - einblenden 156
 - eines Fensters 134
- Titelzeile 44, 53
- Tonart 213
- Töne 397
- Tonkanäle 397
- Tool Types 76
- Threshold 119
- Tutorial 29, 196
- TYPE Disk 76

- Übergröße Grafik 382
- Uhrzeit einstellen 114
- Umbenennen einer Diskette 73
- Underline 104
- Unter-Menüs 130
- Unterstrichen 104
- Use 112
- Utilities 99, 106

- Vergößerung 183
- Vergrößerungsglas 158
- Verschieben des Text-Fensters 322
- Verschieben von Disketten 68
- Versionsnummer 55
- Verzögern eines Befehls 304
- Vibrato 216
- Video-Anschlüsse 445
- Virtueller Bildschirm 111, 146
- Vordergrundfarbe 172

- Warnungen 277
- Wellenformen 366
- Werkbank 28, 41
- Werkzeug
 - Öffnen 94
 - Preferences 112
- Werkzeuge 89, 93 137, 163
- WOM 26
- Workbench 28, 36, 41
 - Befehle 91
 - Icons 88
 - Objekte 67
- Workbench starten 287
- Workbench-Diskette 28, 42
- Workbench-Farben 113
- Workbench-Menü 53
- Write Only Memory 26

- Zeichensatz 104
- Zeichensatz auswählen 181, 189
- Zeichensätze 353
- Zeichensätze laden 181
- Zeichenstil 104
- Zeilennummern optional 425
- Zeit ändern 266
- Zentraleinheit 360
- Ziehen 51
- Ziehen der Maus 125
- Zoom 182
- Zwischenablage 198
- Zyklisches Zeichnen 167

Weitere Fachbücher aus unserem Verlagsprogramm

COMMODORE

P. Rosenbeck

Das Commodore 128-Handbuch 1985, 383 Seiten

In diesem Buch finden Sie einen Querschnitt durch alle wichtigen Funktions- und Anwendungsbereiche des Commodore 128. Sie werden mit dem C64/C128-Modus und der Benutzung von CP/M-3.0 vertraut gemacht, erfahren alles über die Grafik- und Soundmöglichkeiten des C 128, lernen die Techniken der Speicherverwaltung und das Banking kennen und werden in die Programmierung mit Assemblersprache sowie die Grafikprogrammierung des 80-Zeichen-Bildschirms eingeführt. Ein umfassendes Handbuch, das Sie immer griffbereit haben sollten!
Best.-Nr. MT 90195, ISBN 3-89090-195-9
(sFr. 47,80/6S 405,60)

DM 52,-

J. Hückstädt

BASIC 7.0 auf dem Commodore 128 1985, 239 Seiten

Ganz gleich, ob Sie bereits über Programmierkenntnisse verfügen oder nicht, dieses Buch wird Ihnen helfen, den größtmöglichen Nutzen aus dem leistungsstarken BASIC 7.0 des Commodore 128 PC zu ziehen. Sie eignen sich bei der Durcharbeitung dieses Buches alle notwendigen Kenntnisse an, um immer anspruchsvollere Aufgabenstellungen zu bewältigen: Listenverarbeitung, indexsequentielle Dateiverwaltung, Grafikdarstellungen und Soundzeugung. Ein unentbehrliches Lehrbuch, das sich auch für den geübten Anwender als Nachschlagewerk eignet.
Best.-Nr. MT 90149, ISBN 3-89090-170-0
(sFr. 47,80/6S 405,60)

DM 52,-

G. Jürgensmeier

WordStar 3.0 mit MailMerge für den Commodore 128 PC 1985, 435 Seiten

WordStar ist ein umfangreiches und leistungsfähiges Textverarbeitungsprogramm und damit sicherlich zu Recht das meistverkaufte Programm seiner Art. Doch bedeutet dies nicht unbedingt, daß es auch einfach zu bedienen ist. Hier setzt dieses Buch an: Es macht in vorbildlicher Weise mit allen Möglichkeiten von WordStar und MailMerge vertraut und ist damit eine ideale Ergänzung zum Handbuch. Es sammelt alle wichtigen Informationen für den effektiven Einsatz dieser Programme auf dem Commodore 128 PC.
Best.-Nr. MT 780, ISBN 3-89090-181-6
(sFr. 45,10/6S 382,20)

DM 49,-

Dr. P. Albrecht

dBASE II für den Commodore 128 PC 1985, 280 Seiten

Das vorliegende Buch gibt nach einer kurzen Einführung in den Komplex »Datenbanken« eine Anleitung für den praktischen Umgang mit dBASE II. Schon nach Beherrschung weniger Befehle ist der Anwender in der Lage, Dateien zu erstellen, mit Informationen zu laden und auszuwerten. Dabei hilft ihm ein integrierter Reportgenerator, der im Dialog mit dem Benutzer Berichte gestaltet und in Tabellenform ausdrückt.
Best.-Nr. MT 838, ISBN 3-89090-189-1
(sFr. 45,10/6S 382,20)

DM 49,-

Dr. P. Albrecht

Multiplan für den Commodore 128 PC 1985, 226 Seiten

MULTIPLAN wurde ursprünglich für das 16-Bit-Betriebssystem MS-DOS entwickelt. Inzwischen ist aber auch die in diesem Buch beschriebene CP/M-Version für den Commodore 128 PC auf dem Markt, die den vollen Leistungsumfang der 16-Bit-Version enthält.
Das vorliegende Buch soll eine praktische Einführung in den Umgang mit MULTIPLAN auf dem Commodore 128 PC geben. Anhand von praxisnahen Beispielen werden alle Befehle und Funktionen in der Reihenfolge beschrieben, die der Arbeit in der Praxis entspricht. Bereits nach Abschluß des ersten Kapitels werden Sie in der Lage sein, eigene kleine MULTIPLAN-Anwendungen zu realisieren.
Best.-Nr. MT 836, ISBN 3-89090-187-5
(sFr. 45,10/6S 382,20)

DM 49,-

K. Schramm

Die Floppy 1570/1571 2. Quartal 1986, ca. 400 Seiten

Dieses Buch soll es sowohl dem Einsteiger als auch dem fortgeschrittenen Programmierer ermöglichen, die vielfältigen Möglichkeiten dieses neuen Gerätes voll auszuschöpfen. Sämtliche Betriebsarten und Diskettenformate werden ausführlich erläutert. Anhand vieler Beispiele werden Sie in die Dateiverwaltung mit dieser Floppy eingeführt. Der Benutzer lernt die zahlreichen Systembefehle kennen und erfährt zugleich wichtige Grundlagen für das Arbeiten mit dem Betriebssystem CP/M.
Best.-Nr. MT 90185, ISBN 3-89090-185-9
(sFr. 47,80/6S 405,60)

DM 52,-

S. Baloui

C64-Fischertechnik Messen, Steuern, Regeln Februar 1986, 174 Seiten

Ziel dieses Buches ist es, jedem Besitzer eines Commodore 64/VC20 eine neue Welt zu erschließen: Die Welt der Roboter, der computergesteuerten Fertigungsstraßen. Alles, was Sie benötigen, ist einer der beiden genannten Computer und der Fischertechnik-Computing-Baukasten mit dazugehörigem Interface.
Best.-Nr. MT 90194, ISBN 3-89090-194-8
(sFr. 27,60/6S 233,20)

DM 29,90

H. Haberl

Mini-CAD mit Hi-Eddi-Plus 1985, 234 Seiten inkl. Diskette

Neben den »Standardbefehlen« zum Setzen und Löschen von Punkten, dem Zeichnen von Linien, Kreisen und Rechtecken sowie dem Ausfüllen unregelmäßiger Flächen und dem Verschieben und Duplizieren von Bildschirmbereichen bietet Hi-Eddi eine Reihe von Besonderheiten, die dieses Programm von anderen Grafikprogrammen abhebt: Bis zu sieben Grafikbildschirme stehen gleichzeitig zur Verfügung; es besteht die Möglichkeit, Text in die Grafik einzufügen, die Bildschirme zu verknüpfen oder in schneller Folge durchzuschalten.
Best.-Nr. MT 736, ISBN 3-89090-136-0
(sFr. 44,20/6S 374,40)

DM 48,-

Die angegebenen Preise sind Ladenpreise

Sie erhalten Markt & Technik-Bücher bei Ihrem Buchhändler

Markt & Technik Verlag AG Unternehmensbereich Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München

Weitere Fachbücher aus unserem Verlagsprogramm

D. A. Lien

BASIC-Programmierung PC-10/PC-20

1985, ca. 500 Seiten

Ein amerikanisch-lockerer BASIC-Kurs von dem kalifornischen Professor Lien. Durch seine Systematik ideal als Kursunterlage für PC-10/20 und Kompatibles. Mit Einführung in das PC-10-System und Tastendarstellung im Text.

Best.-Nr. PW 559, ISBN 3-921803-66-7
(sFr. 54,30/öS 460,20)

DM 59,-

H. Ponnath

C64 - Wunderland der Grafik

1985, 236 Seiten inklusive Beispieldiskette

Dieses Buch zeigt eine Vielzahl sehr interessanter Lösungen, um die grafischen Möglichkeiten des Commodore 64 optimal zu nutzen. Als Krönung enthält es ein zuschaltbares Assemblerprogramm, das umfangreiche grafische und einige neue BASIC-Befehle anbietet. Im zweiten Teil des Buches wird eine Möglichkeit gezeigt, wie man bis zu 70 verschiedene Farben erzeugen kann. Viele Beispielprogramme begleiten die Reise durch das Wunderland der Grafik.

Best.-Nr. MT 90363, ISBN 3-89090-363-0
(sFr. 45,10/öS 382,20)

DM 49,-

H. L. Schneider/W. Eberl

Das C64-Profilhandbuch

1985, 410 Seiten

Ein Buch, das alle wichtigen Informationen für professionelle Anwendungen mit dem C64 enthält. Mit allgemeinen Algorithmen, die auch auf andere Rechner übertragbar sind, und vielen Utilities, getrennt nach BASIC- und Maschinenprogrammen. Besonders nützlich: erweiterte PEEK- und POKE-Funktionen.

Best.-Nr. MT 749, ISBN 3-89090-110-7
(sFr. 47,80/öS 405,60)

DM 52,-

W.-J. Becker/M. Folprecht

Programmieren unter CP/M mit dem C64

1985, 290 Seiten

Wenn Sie wissen wollen, wie das Betriebssystem CP/M-2.2 auf dem C64 implementiert ist, außerdem einiges über Turbo-Pascal, Nevada-Fortran, MBASIC-80 erfahren wollen, dann ist dieses Buch genau richtig für Sie! Mit Schaltplänen zur eigenen Fertigung des CP/M-Moduls. Für eingeweihte C64-Profis.

Best.-Nr. MT 751, ISBN 3-89090-091-7
(sFr. 47,80/öS 405,60)

DM 52,-

W. Kassera/F. Kassera

C64 - Programmieren in Maschinensprache

1985, 327 Seiten inklusive Beispieldiskette

In diesem Buch finden Sie über 100 Beispiele zur Assembler-Programmierung mit viel Kommentar und Hintergrundinformationen: Das Schreiben von Maschinenprogrammen - Rechnen und Texten mit vorhandenen Routinen - Bedienung von Drucker und Floppy - Wie man BASIC- und Maschinenprogramme verknüpft - Erstellen von eigenen Befehlen in Modulform. Für Profis!

Best.-Nr. MT 830, ISBN 3-89090-168-9
(sFr. 47,80/öS 405,60)

DM 52,-

W. B. Sanders

Einführungskurs: Commodore 64

1984, 276 Seiten

Die Programmiersprache BASIC - Einsatzgebiete des Commodore-64-BASIC: Grafik, Musik, Dateiverwaltung - mit vielen Beispielprogrammen, häufig benötigten Tabellen und nützlichen Tips - für Einsteiger und Fortgeschrittene.

Best.-Nr. MT 685, ISBN 3-89090-017-8
(sFr. 35,-/öS 296,40)

DM 38,-

J. W. Willis/D. Willis

Commodore 64 - leicht verständlich

1984, 154 Seiten

Informationen für den Computer-Neuling - Installation und Inbetriebnahme - Programmieren in BASIC - Grafik und Töne - Auswahl von Hardware und Zubehör - Software für Ihren Computer - die ideale Einführung in das Arbeiten mit Ihrem Commodore 64.

Best.-Nr. MT 700, ISBN 3-89090-022-4
(sFr. 27,50/öS 232,40)

DM 29,80

M. J. Winter

Das Commodore-64-LOGO-Arbeitsbuch

1984, 225 Seiten

Kinder lernen auf dem Commodore 64 mit der Schildkröte als Lehrer: Bilder malen - Grafikeffekte erzeugen - Wörter verarbeiten - Prozeduren und Variablen - Umgang mit Begriffen wie: Längenmaß, Winkel, Dreieck, Quadrat.

Best.-Nr. MT 720, ISBN 3-89090-063-1
(sFr. 31,30/öS 265,20)

DM 34,-

D. Highmore/L. Page

Der sensible Commodore 64

1985, 130 Seiten

Eine Softwareammlung zu den technologischen Neuererscheinungen im Commodore 64 - für Erstbenutzer wie für Experten ein Buch zur optimalen Softwarenutzung.

Best.-Nr. PW 727, ISBN 3-921803-45-4
(sFr. 27,50/öS 232,40)

DM 29,80

J. Mihalik

35 ausgesuchte Spiele für Ihren Commodore 64

1984, 141 Seiten

Programmieren Sie selbst 35 faszinierende Spiele - geschrieben in Commodore-64-BASIC - mit Farbe, Grafiken und Ton - Vorschläge zur Programmabwandlung - für kreative Computerfans, die ihre Programmierkenntnisse vertiefen wollen!

Best.-Nr. MT 774, ISBN 3-89090-064-X
(sFr. 23,-/öS 193,40)

DM 24,80

E. H. Carlson

BASIC mit dem Commodore 64

1984, 320 Seiten

Ein BASIC-Lehrbuch für den jugendlichen Anfänger - übersichtlich gegliederte Lernprogramme - Alles über INPUT-GOTO - Let-Befehle - Editorfunktionen - POKE-Befehle für die Grafik - geeignet auch als Leitfaden für Lehrer und Eltern.

Best.-Nr. MT 657, ISBN 3-922120-91-1
(sFr. 44,20/öS 374,40)

DM 48,-

Die angegebenen Preise sind Ladenpreise

Sie erhalten Markt & Technik-Bücher bei Ihrem Buchhändler

Markt & Technik Verlag AG Unternehmensbereich Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München

Weitere Fachbücher aus unserem Verlagsprogramm

F. Ende

Das große Spielebuch - Commodore 64

1984, 141 Seiten

46 Spielprogramme · Wissenswertes über Programmier-
technik · praxisnahe Hinweise zur Grafikerstellung · alles
über Joystick- und Paddlesteuerung · das Spielebuch mit
Lerneffekt.

Best.-Nr. MT 603, ISBN 3-922120-63-6

(sFr. 27,50/6S 232,40)

Best.-Nr. MT 604 (Beispiele auf Diskette)

(sFr. 38,-/6S 342,-)

DM 29,80

DM 38,-*

* inkl. MwSt. Unverbindliche Preisempfehlung.

P. W. Dennis/G. Minter

Spiele für den Commodore 64

1984, 196 Seiten

Bewährte alte und raffinierte neue Spiele für Ihren Com-
modore 64 · klar und übersichtlich gegliederte Programme
im Commodore-BASIC · Sie lernen: wie man Unterpro-
gramme einsetzt · eine Tabelle aufbauen und verarbeiten ·
Programme testen · mit vielen Programmiertricks · für
Anfänger.

Best.-Nr. MT 90074, ISBN 3-89090-074-7

(sFr. 23,-/6S 193,40)

Best.-Nr. MT 795 (Beispiele auf Diskette)

(sFr. 38,-/6S 342,-)

DM 24,80

DM 38,-*

* inkl. MwSt. Unverbindliche Preisempfehlung.

S. Krute

Grafik & Musik auf dem Commodore 64

1984, 336 Seiten

68 gut strukturierte und kommentierte Beispielprogramme
zur Erzeugung von Sprites und Klangeffekten · Sprite-Tricks
· Zeichengrafik · hochauflösende Grafik · Musik nach
Noten · spezielle Klangeffekte · Ton und Grafik · für fortge-
schrittene Anfänger, die alle Möglichkeiten des C64 ausnut-
zen wollen.

Best.-Nr. MT 743, ISBN 3-89090-033-X

(sFr. 35,-/6S 296,40)

DM 38,-

H. L. Schneider

Commodore-64-Listings - Band 1: Spiele

1984, 199 Seiten

Mit ausführlicher Dokumentation · Spielanleitung · Varia-
blen für die Änderung der Spiele · vollständige Listings für:
Bürger Joe · Nibbler · Zingel Zangel · Universe · Würfelpo-
ker · Maze-Mission · der magische Kreis · Todeskommando
Atlantik · Enterprise.

Best.-Nr. MT 748, ISBN 3-89090-068-2

(sFr. 23,-/6S 193,40)

Best.-Nr. MT 804 (Beispiele auf Diskette)

(sFr. 38,-/6S 342,-)

DM 24,80

DM 38,-*

* inkl. MwSt. Unverbindliche Preisempfehlung.

H. L. Schneider

Commodore 64-Listings

Band 2: Dateiverwaltung · Schule · Hobby

1984, 179 Seiten

Ein Buch mit Programmen für die ganze Familie · DATAVE -
Eine Dateiverwaltung · mathematische Funktionen · Konju-
gation und Deklination in Latein · Regressionsanalyse · Bun-
desligatabelle.

Best.-Nr. MT 766, ISBN 3-89090-071-2

(sFr. 23,-/6S 193,40)

DM 24,80

Dr. P. Albrecht

Commodore 64 - Multiplan

1984, 230 Seiten

Multiplan jetzt auch für den Commodore 64 · der volle Lei-
stungsumfang der 16-Bit-Version · Einführung in die
Arbeitsweise von Tabellenkalkulationsprogrammen · praxis-
nahe Beispiele · Beschreibung aller Befehle und Funktio-
nen. Ein Buch nicht nur für Anfänger.

Best.-Nr. MT 655, ISBN 3-922120-89-X

(sFr. 44,20/6S 374,40)

DM 48,-

K. Schramm

Die Floppy 1541

1985, 434 Seiten

Für alle Programmierer, die mehr über ihre VC-1541-Floppy-
station erfahren wollen. Der Vorgang des Formatierens · das
Schreiben von Files auf Diskette · die Funktionsweise von
schnellen Kopier- und Ladeprogrammen · viele fertige Pro-
gramme · Lesen und Beschreiben von defekten Disketten ·
Für Einsteiger und für fortgeschrittene Maschinensprache-
Programmierer.

Best.-Nr. MT 90098, ISBN 3-89090-098-4

(sFr. 45,10/6S 382,20)

Best.-Nr. MT 710 (Beispiele auf Diskette)

(sFr. 38,-/6S 342,-)

DM 49,-

DM 38,-*

* inkl. MwSt. Unverbindliche Preisempfehlung.

ATARI

I. Lücke/P. Lücke

Der ATARI 520 ST

2. überarbeitete und erweiterte Auflage 1986, 198 Seiten

Dieses Buch enthält alle Informationen, die für Interessierte
und für alle stolzen Besitzer eines gerade erworbenen ATARI
520/260 ST wichtig sind. Die jetzt vorliegende überarbeitete
und erweiterte Auflage trägt den neuesten Entwicklungen
bei ATARI Rechnung. Unter anderem wurden das inzwi-
schen deutschsprachige Betriebssystem und einige geän-
derte Systemausstattungsmerkmale berücksichtigt. Das
Buch ist somit nicht nur eine Rechnerbeschreibung mit ho-
hem Informationswert, es leistet auch als Nachschlagewerk
wertvolle Dienste.

Best.-Nr. MT 90229, ISBN 3-89090-229-4

(sFr. 45,10/6S 382,20)

DM 49,-

Die angegebenen Preise sind Ladenpreise

Sie erhalten Markt & Technik-Bücher bei Ihrem Buchhändler

Markt & Technik Verlag AG Unternehmensbereich Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München



MARKUS BREUER

Jahrgang 1960, ist zur Zeit noch Student der Informatik an der Universität Dortmund. Studienschwerpunkte: Software-Ergonomie, Datenbanken und Objekt-orientierte Systeme. Thema der Diplomarbeit: Entwicklung einer neuartigen, Objekt-orientierten Benutzeroberfläche für Datenbanken und Information-Retrieval-Systeme.

Neben dem Studium langjährige Tätigkeit in der Entwicklung und Realisierung verschiedener Programmsysteme auf Mikrocomputern, in jüngster Zeit vor allem auf Lisa und Macintosh.

Freier Autor mit den Spezialgebieten Programmiersprachen, Software-Entwicklungen, Macintosh und andere hochentwickelte Benutzerschnittstellen.

Das AMIGA- Handbuch

Der Commodore Amiga stellt einen neuen Schritt in der Entwicklung der Personal Computer dar. Er setzt die neuesten Entwicklungen der Chip-Technologie ein, um dem Endanwender eine extrem leistungsfähige Maschine zu einem vergleichsweise günstigen Preis auf den Schreibtisch stellen zu können.

Er besitzt geradezu phantastische Farbgrafik-Fähigkeiten (4096 Farben), die auch für die Gestaltung der Benutzeroberfläche konsequent eingesetzt werden. Die Bedienung des Amiga ist deshalb sehr leicht zu erlernen. Sie benötigen keine schwer zu merkenden Befehle mit unleserlichen Abkürzungen. Statt dessen werden farbige grafische Symbole und Menüs, aus denen Sie die gewünschten Befehle einfach auswählen, eingesetzt.

Trotz des sehr hohen Bedienungskomforts, den er bietet, ist dieser neue Computer extrem schnell. Er ist sogar so schnell, daß er scheinbar gleichzeitig mehrere Programme ausführen kann (Multitasking) und erspart Ihnen so langwierige (und langweilige!) Wartezeiten.

Aus dem Inhalt:

- Beschreibung der System-Architektur
- Auf der Werkbank des Amiga (Workbench)
- Intuition: Grundlagen der Amiga-Bedienung
- Die Grafikprogramme Graficraft und Deluxe Paint
- Amiga für Fortgeschrittene:
 - die graphische Benutzerschnittstelle »CLI«
- Automatisierung des Amiga (Kommandofolgen)
- Die Spezialchips des Amiga: DENISE, PAULA und AGNUS
- Grundlagen von Sound und Grafik
- Programmierung des Amiga (Amiga-BASIC von Microsoft, LATTICE C)

Das vorliegende Buch enthält zum einen eine ausführliche Beschreibung des Amiga-Systems, die Ihnen bei einer anstehenden Kaufentscheidung hilft, diesen neuen Computer einzuschätzen. Zum anderen ist es aber auch als Handbuch geschrieben, mit vielen Bildschirmfotos und Übersichtstabellen, das Ihnen im täglichen Einsatz hilft, schnell und reibungslos mit dem Amiga zu arbeiten.

Pflichtlektüre für jeden, der sich für den neuen Supercomputer Amiga interessiert.

ISB N 3-89090-228-6

Markt & Technik



4 001057 902282

DM 49,-

sFr. 45,10
öS 382,20