

AUSGABE



8

AMIGA

Markt & Technik

DM 16,-

ÖS 120,- / Sfr 16,-

Lit 16000 / hfl 21,- / dkr 75,-

SONDERHEFT

STARTHILFE

Know-how für den Einstieg

Kurse:

CLI und Workbench

Problemlösung:

Amiga optimal konfiguriert

Rat & Tat

Kaufberatung:

Hardware, Software, Public Domain

Pannenhilfe:

Wenn Sie nicht mehr weiter wissen...

Lexikon:

So verstehen Sie Computer-Chinesisch

**ALLE
PROGRAMME
AUCH AUF
DISKETTE
ERHÄLTICH**



STEFAN OSSOWSKI'S SCHATZTRUHE

- ① **Haushaltsbuch** bis zu 25 Konten, flexibel, leicht bedienbar, mit umfangreicher **deutschsprachiger** Dokumentation DM 8,-
- ③ **MountainCAD** professionelles CAD-Programm, **deutsche Anleitung** DM 8,-
- ④ **Spiele I, II, III** 10 erstklassige PD-Spiele aus allen Bereichen wie Action, Geschicklichkeit, Strategie (3 Disketten) DM 24,-
- ⑤ **Anti-Virus 8** Programme gegen alle Viren DM 8,-
- ⑥ **Text** hochwertige **deutsche** Textverarbeitung DM 8,-
- ⑦ **Utility-Disk** 25 nützliche Utilities aus allen Bereichen DM 8,-
- ⑨ **Sonix-Paket** Original-Sonix-Player + 4 weitere Disketten mit phantastischer Sonix-Musik. **Top-Hit!** DM 40,-
- ⑩ **Business 3** Disketten: Tabellenkalkulation, Vers. engl. DM 24,-, relat. Datenbank, sehr gute Textverarb. Vers. deutsch DM 70,-
- ⑬ **Paranoid** sensationelles Breakout-Spiel DM 8,-
- ⑭ **Buchhaltung** erstes **deutsches** PD-Buchhaltungsprogr. DM 8,-
- ⑯ **AMIGA-Paint** sehr gutes **deutsches** Malprogramm DM 8,-
- ⑰ **Videodatei** bringt Ordnung in Ihre Videodatei, **deutsch** DM 8,-
- ⑱ **Fußballmanager** bei diesem Spiel können Sie Ihre Fähigkeiten als Manager eines Fußballclubs testen, **deutsch** DM 8,-
- ⑲ **Platten/Cassetten/CD-Liste**, **deutsch** DM 8,-
- ⑳ **Giroman** komfortables **deutsches** Programm, mit dem Sie Ihr Girokonto einfach verwalten können DM 8,-
- ㉒ **Kampf um Eriador, V 2.0** taktisches Strategiespiel für 2 Personen mit sehr guter Grafik und Sound, **deutsch** DM 8,-
- ㉔ **Label 2.0** Etikettendruckprogramm, **deutsch** DM 8,-
- ㉖ **Risiko** die Amiga-Umsetz. d. bek. Brettspiels, **deutsch** DM 8,-
- ㉗ **Ray-Tracing-Construction-Set, V 2.0** phantastisches Programm zur Berechnung von Licht und Schatten - siehe Test Amiga 1/88 - komplett auf 3 Disketten mit **deutscher Anleitung** DM 24,-
- ㉘ **Wizard of Sound** ein phantastisches Musikprogramm zur Erstellung eigener Lieder, mit **deutscher** Anleitung (2 Disks) DM 10,-
- ㉙ **Broker** ein sehr gutes **deutsches** Börsenspiel DM 8,-
- ㉚ **Quickmenü** erst. Sie sich Ihre eig. Workbench i. **deutsch** DM 8,-
- ㉛ **Blizzard** phantastisches Ballerspiel m. sehr guter Animat. DM 8,-
- ㉜ **DSort** **deutsches** Diskettenkatalogisierungsprogramm DM 8,-
- ㉝ **Pascal** ein komplettes Pascal-Paket (3 Disketten) mit Compiler, **deutscher Anleitung** u. einem s. gut. deutschen Editor DM 24,-
- ㉞ **Diskey** Diskettenmonitor mit **deutscher Anleitung** DM 8,-
- ㉟ **Peters Quest** Geschicklichkeitsspiel mit lustiger Handlung und **deutscher Anleitung** DM 8,-
- ㊱ **Spiele** auf dieser Diskette sind 3 Tetris-ähnli. Spiele enth. DM 8,-
- ㊲ **MRBackup** Festplattensicherungsprogramm mit **deutscher Anleitung** DM 8,-
- ㊳ **Universal-Datei** **deutsches** Datei-Verwaltungsprogr. DM 8,-
- ㊴ **Assembler** ein komplettes Entwicklungssystem für Maschinensprache in **deutsch!** DM 8,-
- ㊵ **Bibel-Quiz** lehrreich und unterhaltsam DM 8,-
- ㊶ **Faktura** Fakturierungsprogramm inkl. Mahnungen und Adressen - mindestens 1 MB Speicher DM 8,-
- ㊷ **BootMaster** Mit diesem Programm können Sie individuelle Bootblöcke mit Lauftext und Sternenhintergrund erstellen. DM 10,-
- ㊸ **Banner II** Dieses Programm ermöglicht es Ihnen, komplette Banner mit Ihrem Drucker zu erstellen. Leicht bedienbar! DM 8,-
- ㊹ **Boulder V1.3** Boulder ist ein **sehr schnelles Geschicklichkeitsspiel**, das an den C 64-Klassiker Boulder-Dash angelehnt ist. Ohne Sound! DM 8,-

* TAIFUN * TAIFUN * TAIFUN * TAIFUN * TAIFUN * TAIFUN * TAIFUN * TAIFUN

Die deutsche Public-Domain-Serie aus dem Hause Ossowski! Wußten Sie schon, daß beim PD-Versand Stefan Ossowski alle drei Monate 10 neue TAIFUN-Disketten mit den interessantesten Neuerscheinungen des PD-Marktes vorgestellt werden? Bevorzugt präsentieren wir Ihnen die neuesten deutschen Programme. Am 10. November erschienen die neuen TAIFUN-Disketten Nr. 111 bis 120.

Schnupperpreis: DM 53,- V-Scheck, DM 57,- Nachnahme

Versandkosten Inland: DM 3,- V-Scheck DM 7,- Nachn.
(Porto/Verpackung): Ausland: DM 6,- V-Scheck DM 15,- Nachn.

ABO-SERVICE

Bei uns erhalten Sie fast jede PD-Serie auch im günstigen Abonnement! Auf unsere Staffelpreise gewähren wir außerdem einen **10%igen ABO-Rabatt!** Die Fish-Serie ist z. B. schon bis Nr. 270 lieferbar! Rufen Sie uns doch einfach an oder schreiben Sie uns, wenn Sie an weiteren Informationen zu unserem ABO-Service interessiert sind!

Zuverlässigkeit
+ Schnelligkeit
+ Service
= PD-Vertrieb Stefan Ossowski
Testen Sie uns!

Professionelle Anwendungssoftware für den anspruchsvollen User (ab Nr. 101):

- ⑩① **RIM-5 = Relationale Datenbank**
Außerst leistungsfähig, sowohl für den privaten als auch für den geschäftlichen Bereich geeignet. **Mit deutscher Anleitung und ausführlichem Einführungskurs.** DM 30,-
- ⑩② **AnalytiCalc = Tabellenkalkulation**
Leistungsstark mit deutscher Anleitung. Ein unverzichtbares Hilfsmittel für Kalkulationsaufgaben jeder Art (**Bericht Amiga 12/89**). DM 30,-
- ⑩③ **DEA Arithmetica = Die Göttin Arithmetica**
Besonders für Schüler, Studenten und Lehrer geeignet. Ableitungen, **Kurvendiskussion** und Skizzieren von Funktionsgraphen problemlos möglich. **Deutsch!** DM 30,-
- ⑩④ **Haushaltsbuch Version 2.1**
Komplettlösung zur Verwaltung Ihrer privaten Finanzen! Sie erhalten einen Überblick über Ihre Finanzen, können Ihr Geld effektiver nutzen und Schwachstellen in der Budgetierung erkennen. Leicht bedienbar, auch für Computer-Neulinge geeignet. Unterstützung durch Demo-Dateien, frei von buchhalterischen Fachausdrücken. Funktionsüberblick: verschiedene Dateien, mehrere Kontenlisten, bis zu 10 Bilanzen, 40 freidefinierbare Konten, Suchroutinen, doppelte Buchführung, Filterfunktionen, Jahresübertrag, Mausunterstützung ... **Deutsche Dokumentation im Ringbuch!** 1 MB Speicher erforderlich! DM 98,-
- ⑩⑤ **Xytronic II**
Intergalaktische Handelssimulation mit guter Grafik und gutem Sound sowie sehr hoher Motivation. **Natürlich in Deutsch!** DM 29,-
- ⑩⑥ **Data-Manager**
Eine sehr leicht zu bedienende Dateiverwaltung, mit der Sie Adressen, Mitgliederlisten etc. perfekt verwalten können. **Deutsches Programm mit deutscher Anleitung!** DM 15,-
- ⑩⑦ **EGOS = European Game of Strategy**
Strategiespiel für 2 Personen indem Sie um die Vorherrschaft in Europa kämpfen! Taktik und Strategie sind gefordert! **Mit deutscher Anleitung!** DM 29,-
- ⑩⑧ **Bundesliga 2000**
Programm zur Bundesligaergebnisverwaltung und Tabellenberechnung. Verwaltung beliebig vieler Ligen möglich. Historie seit 1985. **Deutsch!** DM 15,-
- ⑩⑨ **Money Player Deluxe - Geldspielgerät**
Sie fühlen sich wie in der Spielhalle! **Start- und Risikoautomatik, Sonder-spiele viele Extras und toller Spielspaß. Palaufösung und Maussteuerung!** DM 39,-
- ⑩⑩ **Wizard of Sound 2.0 - Musikprogramm**
WoS 2.0 ist ein sehr gutes Musikprogramm mit 61 Instrumenten, kompletter Notation, Player-Programm, Instant-Replay-Modus, einfachste Noteneingabe, ... Auch für Musik-Laien geeignet! **Auf 2 Disketten mit deutscher Dokumentation!** DM 35,-
- ⑩⑪ **Broker 1.2 - Das Börsenprogramm**
Sehr **realistische Börsensimulation**, mit bis zu 4 Spielern spielbar. Broker 1.2 ist unterhaltsam, spannend und lehrreich. **Deutsch!** 1 MB Speicher! DM 29,-
- ⑩⑫ **Vokabel-Trainer-Englisch 1.0**
Leicht zu bedienendes, leistungsfähiges Lernprogramm. Der Wortschatz ist individuell erweiterbar. Damit macht das Lernen Spaß! **Deutsch!** DM 15,-
- ⑩⑬ **Pente**
Ein **"5-Gewinnt-Spiel"** erweitert um einige Regeln. Sie spielen gegen den Computer. Für jung und alt mit hoher Motivation, Super! DM 19,-
- ⑩⑭ **BootMenü**
Mit BootMenü von Oliver Wagner können Sie im Disketten-Bootblock ein Menü einrichten, über das Sie beim Bootvorgang per Maus 10 verschiedene Programme aufrufen können! **Mit deutscher Anleitung!** DM 19,-
- ⑩⑮ **CBB - CopyBootBlock Installierungsprogramm**
Mit CBB können Sie in Ihrem Bootblock ein Kopierprogramm installieren, das sie bei **jedem Bootvorgang per Mausclick starten können.** 2 Laufwerke oder 1 MB. DM 19,-
- ⑩⑯ **Danger Castle**
Ein Hüpf- und Rennspiel für 1 oder 2 Personen mit **Wahnsinnsgrafik, Stereo-sound** und 20 verschiedenen Levels! **Deutsche Anleitung!** DM 39,-
- ⑩⑰ **SpeedRunner**
Eine Loadrunner-Variante mit Level-Editor, 100 Levels, **Super-Grafik und Klasse-Sound.** In Assembler geschrieben und **Palaufösung!** **Deutsche Anleitung!** DM 39,-
- ⑩⑱ **Transit - Imagine**
Mit diesem Programmpaket können Sie Grafiken in eigene Programme einbinden, IFF-Grafiken in C-Sourcecode konvertieren, in eigenen C-Programmen verwenden. Auch für Modula II und Assembler! Mit Imagine können Sie Menüpunkte und Gadgets entwerfen und als C-Source abspeichern! **Mit deutscher Anleitung!** DM 89,-

Wir führen alle bekannten PD-Serien wie Fish, Panorama, Faug Amicus, Auge, Taifun, Chiron, RPD, Kickstart, Sli-deshows, TBAG, Franz ... Viele Programme deutschsprachig.

5,-DM

kosten unsere aktuellen 2 Katalogdis-ketten. Mit deutschem Inhaltsverzeichnis unseres gesamten PD-Angebots von weit über 1500 Disketten. Ab DM 4,50. Gegen Verrechnungsscheck oder in Briefmarken anfordern.

Stefan Ossowski - Ihr PD-Spezialist - Veronikastr. 33, D-4300 Essen 1, Tel./Btx: 0201/788778

● Forderungen sind noch lange nicht berechtigt, nur weil sie ständig wiederholt werden. Oft werden sie aber deswegen erfüllt. »Ceterum censeo Carthaginem esse delendam.« Mit diesem Satz pflegte vor mehr als 2000

● Die Zahl derer, die genau wissen, was in einem Computer vorgeht, blieb trotz allem eine verschwindend geringe Minderheit. Stehen Computer also nur zur Dekoration auf dem Schreibtisch? Oder haben sich die Zeiten gewandelt?

● Man könnte es sich einfach machen und – wie viele meiner geschätzten Kollegen – behaupten: Endlich steht der Mensch im Mittelpunkt.

Ceterum censeo

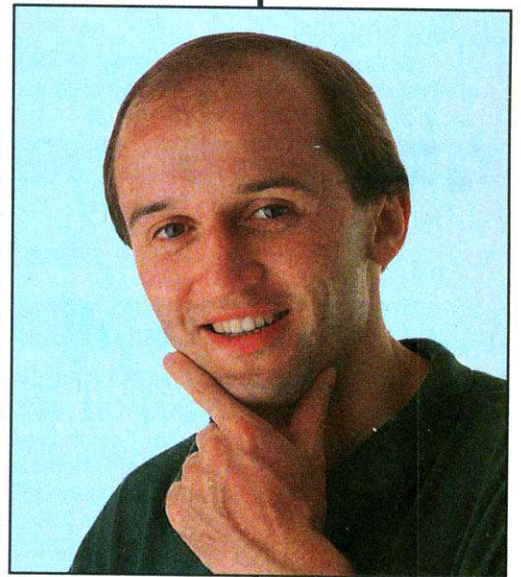
Jahren der Politiker Marcus Porcius Cato seinen römischen Senatskollegen auf die Nerven zu gehen.

● Anfang der achtziger Jahre unseres Jahrhunderts wurde immer wieder dieselbe Litanei heruntergebetet: Ohne fundierte Computerkenntnisse habe der Arbeitnehmer der Zukunft bereits in wenigen Jahren keine Chance mehr, sich auf dem Arbeitsmarkt zu behaupten. Selbst Schüler, hieß es, bräuchten dann intensive »EDV-Kenntnisse«, um halbwegs brauchbare Zensuren zu erreichen. Die ganze Welt sah man schon verdrahtet und verkabelt und professionelle Schwarzseher warnten vor der maschinellen Verdummung ganzer Landstriche – der Mensch als integriertes Bedienelement einer elektronischen Konsole.

● Was ist aus dieser schönen neuen Welt geworden? War Anfang der siebziger Jahre »Computer« noch ein Begriff, von dem nur wenige Insider wußten, was er bedeutet, so steht heute auf fast jedem Schreibtisch ein Rechner, die Zahl der Informatik-Studenten stieg drastisch an und jeder Schüler wird heute mit der Lehre von Bits und Bytes traktiert.

Leider ist dies noch immer Wunschdenken. Zwar sind zaghafte Ansätze sichtbar, Computer begreifbarer und ihre Bedienung intuitiver zu machen, aber reicht das? Bevor wirklich der Mensch im Mittelpunkt steht, muß sich vieles ändern.

● Computer sind dazu da, Probleme zu lösen, nicht zu schaffen. Bis sich diese Ansicht durchgesetzt hat, wird einige Zeit vergehen. Bis dahin – ceterum censeo...



Ihr
Klaus Sonnenleiter
(Redakteur)

K. Sonnenleiter

Kaufberatung

6 Fragen über Fragen

... und ebenso viele Antworten. Das Angebot an Soft- und Hardware für den Amiga wird immer größer. Wir helfen Ihnen, den Überblick zu behalten.

8 Stück für Stück

Der »nackte« Amiga 500 ohne Erweiterung wird schnell zum Alptraum. Hier erfahren Sie, welche Erweiterung Sie für welche Anwendung benötigen.

13 Führer im Software-Dschungel

Amiga-Software zu überblicken ist die reinste Sisyphusarbeit. Martin Jobst hat sich für Sie umgesehen und stellt die wichtigsten Programme vor.

50 Fast umsonst

Public Domain ist ein unerschöpfliches Software-Reservoir mit interessanten Alternativen und Ergänzungen zur Standardsoftware.

Hintergrund

24 Der gläserne Amiga

Was steckt im Gehäuse dieses Wunderrechners? Hier finden Sie die Erklärung.

Grundlagen

28 Von Fenstern, Mäusen und Konsolen

Der Amiga ist wesentlich einfacher zu bedienen als die meisten andern Computer. Dennoch hält er so manchen Stolperstein bereit. Hier finden Sie eine leicht verständliche Einführung, die Ihnen über die ersten Hürden hilft.

46 Die Schaltzentrale

Mit dem Preferences-Programm stellen Sie alle Eigenschaften Ihres Amiga wunschgemäß ein. Wie, das verrät Ihnen Andreas Regul.

Erste Hilfe

90 Pannenhilfe

Gerade zu Beginn Ihrer Amiga-»Karriere« werden Sie zwangsläufig auf einige Hindernisse stoßen. Hier erfahren Sie, wie diese beseitigt werden.

Einstieg

52 Kein Platz an der Sonne

Rückenschmerzen, Nackenkrämpfe, spiegelnde Monitore? Der richtige Aufbau Ihres Arbeitsplatzes kann Ihnen viel Frust und Ärger ersparen.

54 Fast wie Weihnachten

Durch das Chaos beim Auspacken Ihres Amiga kann leicht etwas schiefgehen. Brigitte Bobenstetter zeigt Ihnen, wie Sie es richtig machen.

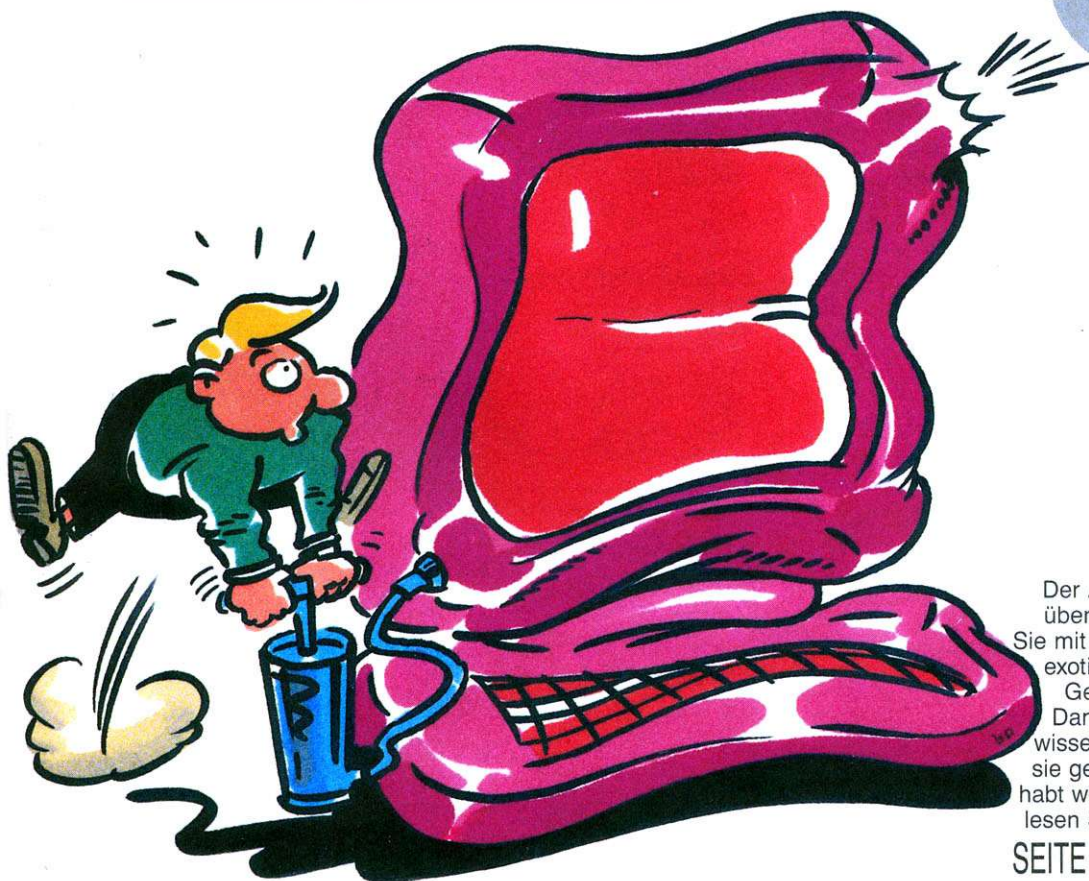
55 Countdown

Nicht jeder Start ist ein Erfolg. Damit Ihr Start auf dem Amiga keine Bruchlandung wird, lesen Sie diesen Artikel.

Wenn Sie gezielt Hilfe für Ihren Amiga benötigen, ist unsere Pannenhilfe immer für Sie da.

SEITE 90

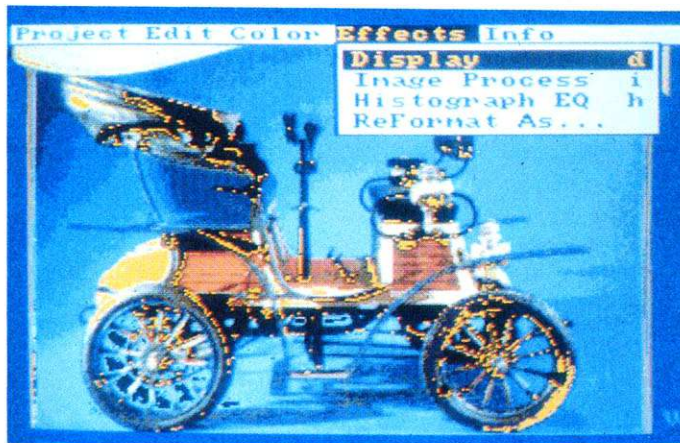
Pfff...



Der Amiga überrascht Sie mit vielen exotischen Geräten. Damit Sie wissen, wie sie gehandhabt werden, lesen Sie ab

SEITE 28

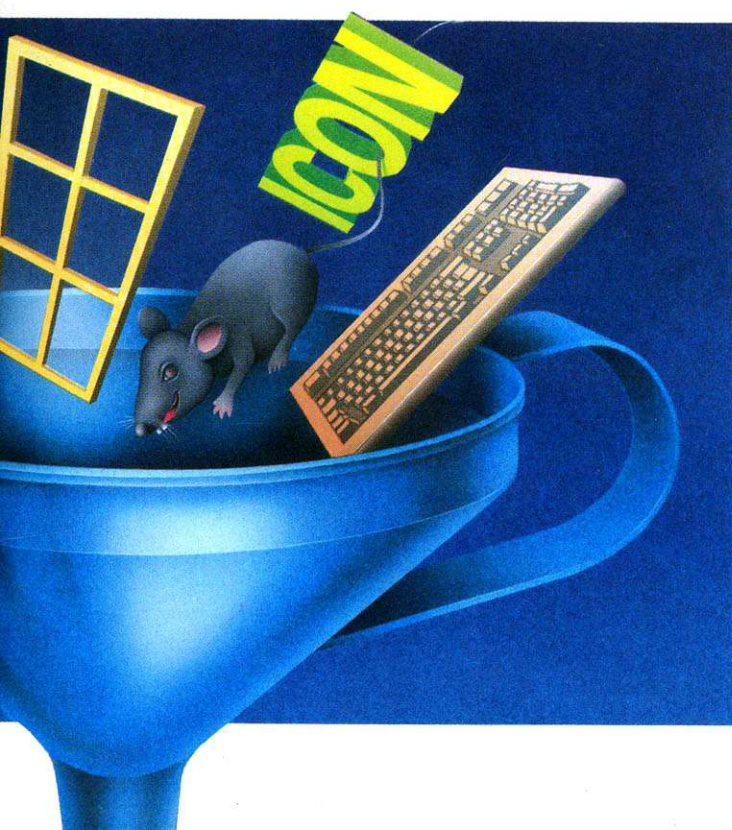




Es gibt eine Vielzahl von guten und hilfreichen Programmen auf dem Amiga. Aber welche sind die richtigen für Sie? Wir stellen Ihnen die interessantesten vor. **SEITE 13**



Wollten Sie nicht schon immer wissen, was in Ihrem Computer steckt? Der gläserne Amiga lüftet das Geheimnis und zeigt, wo Sie RAM, ROM und CPU finden. **SEITE 24**



Lexikon

59 Amiga von A bis Z

In der Computer-Fachliteratur haben sich viele Begriffe durchgesetzt, die nicht im »Duden« stehen. Wir machen dieses »Computerchinesisch« verständlich.

Bücher

88 Aus unserem Bücherregal

Bücher zum Amiga gibt es wie Sand am Meer. Aber nur wenige davon sind wirklich brauchbar. Wir stellen alle vor, welche unbedingt zu jeder Grundausstattung gehören sollten.

Listings

94 Vollbremsung im Amiga

Die unentbehrliche Hilfe für Highscore-Jäger. Ein Tastendruck genügt, und schon ist das Spiel gestoppt.

95 Der Bilderklau

Grafiken aus kommerziellen Programmen sind häufig traumhaft schön. Bauen Sie solche Bilder mit diesem raffinierten Hilfsprogramm aus.

98 Öl, Geld und Bomben

... brauchen Sie, wenn Sie Ihre Gegner bei »Öl-Magnat« bezwingen wollen. Alles drei erhalten Sie durch geschicktes Wirtschaften und etwas Glück im Spiel.

106 Sperrgebiet

Irrgartenspiele sind eine faszinierende Sache. Lassen Sie sich von »Maze« begeistern. Aber sind Sie vorsichtig, Sie bewegen sich auf wackligen Pfaden.

Programmieren

117 Programmieren, was ist das?

Programmieren ist keine Magie. Sie brauchen nur etwas Mut, um den ersten Schritt zu wagen, und Sie sollten wissen, was Sie tun.

121 Turmbau zu Babel

Die Vielfalt an modernen Programmiersprachen erinnert schon fast an den legendären Turmbau zu Babel. Hans Waldhäusel schafft den richtigen Überblick.

126 Big Brother is checking you

Abtippen ist eine fehlerträchtige Angelegenheit. Schonen Sie Ihre Nerven und benutzen Sie unseren Checksummer.

Sonstiges

3 Editorial

57 Mitmachkarte

124 Impressum

Alle Programme aus Artikeln mit einem -Symbol finden Sie auch auf der Programm-service-Diskette zu diesem Sonderheft

Kaufberatung **Fragen ü**

**Was
bringt mir eine
Festplatte?**

**Was muß
ich beim
Kauf beachten?**

**Speicher-
erweiterung
oder ein zweites
Laufwerk?**

**Hf-Emulator
oder Monitor?**

er Fragen

**Der Amiga ist in der Grundausstattung
für die meisten Anwendungen nicht ausreichend.**

**Aber bevor Sie jetzt losstürzen, um Ihre Ausrüstung zu
vervollständigen, lesen Sie erst unsere
Kaufberatung – und holen Sie sich dann,
was Sie wirklich brauchen.**

Um Ihren Amiga richtig zu nutzen, benötigen Sie leistungsfähige Programme. Oft allerdings arbeiten diese Programme nicht mit der Minimalausstattung eines Amiga 500 (512 KByte Speicher und ein Disketten-Laufwerk) zusammen.

Bevor Sie jetzt aber Ihren frisch erstellten 500er gegen eine leistungsfähigere Variante eintauschen, warten Sie's ab. Denn mit dem Amiga verfolgt Commodore ein weitgehend offenes Konzept. Sie können ihn auch nach dem Kauf noch mit Erweiterungen versehen. So sind im Handel verschiedene Arten von Speichererweiterungen, zusätzliche Disketten-Laufwerke, schnelle Festplatten und vieles mehr erhältlich.

Oft wird den Einsteiger jedoch angesichts des riesigen Angebots eine gewaltige Unsicherheit überkommen. Denn woher soll er wissen, mit was er seinen Computer am vernünftigsten ausstatten soll? Und welche Erweiterung sollte für die von ihm bevorzugte Anwendung Vorrang haben.

Den Überblick über jede verfügbare Erweiterung können wir Ihnen natürlich auch nicht verschaffen. Aber wir wollen Ihnen einen »Weg durch den Dschungel« bahnen, um Ihnen die Entscheidung zu erleichtern, welche Erweiterung für Sie Vorrang haben sollte. Auf diese Weise stellen Sie Ihre Computeranlage »Stück für Stück« (siehe Seite 8) zusammen und bauen sie stets so weit aus wie Sie es gerade für nötig halten.

Aber nicht nur die Hardware muß Ihren Erfordernissen angepaßt werden, auch beim Softwarekauf sollten Sie sich Gedanken machen, welches Programm Ihren Bedürfnissen gerecht wird. Natürlich können Sie nicht jedes Programm aufwendig testen, bevor Sie es kaufen. Auch hätte dieses Vorgehen keinen großen Sinn – Sie wären hinterher womöglich ebenso schlaue wie vorher.

Statt dessen haben wir für Sie einen »Führer durch den Software-Dschungel« engagiert. Martin Jobst hat die gängige Amiga-Software unter die Lupe genommen und sortiert: Zunächst nach Anwendungsgebieten und dann nach Leistungsumfang und Bedienbarkeit.

Lassen Sie sich überraschen auf Seite 13. Aber Standardsoftware muß nicht immer die Lösung sein. Für viele Aufgaben gibt es Lösungen im Public-Domain-Pool. Dabei handelt es sich meist um Programme von Autoren, die – aus welchen Gründen auch immer – kein Interesse an einer kommerziellen Verwertung ihrer Programme haben. Um diese Programme dennoch einer breiten Öffentlichkeit zugänglich zu machen, werden sie über eine der zahlreichen Public-Domain-Serien verbreitet (beispielsweise »Fred Fish«, »RPD«, »Taifun«, und viele andere). Auf Seite 50 finden Sie jede Menge Software »Fast umsonst«.

**Ist PD
für mich
sinnvoll?**

**Welcher
Drucker?**

**Welche
Programme
brauche ich?**

Der Amiga 500 besticht durch den günstigen Preis und trotzdem volle Kompatibilität zu seinem großen Bruder. Der niedrige Preis schlägt sich jedoch in der mangelnden Ausstattung nieder. Während der Einsteiger mit der Grundkonfiguration anfangs durchaus zufriedengestellt ist, kann das Arbeiten mit dieser schnell sehr mühsam werden. Wir zeigen Ihnen, welche Hardware-Ausstattung zum vernünftigen Arbeiten mit dem Amiga 500 notwendig ist und welche Geräte Sie am besten wählen.

Der Speicher- riesen?

Umsteiger vom C64 oder ähnlichen Computern kennen das Problem. Beim Händler hört man etwas von 512 KByte RAM-Speicher – »gigantisch«. Zu Hause findet man schon bei den ersten Versuchen, die Multitaskingfähigkeiten zu nutzen, eine störende Meldung auf dem Bildschirm. Ein Blick ins Handbuch erhärtet den Verdacht: »Fehler 103« bedeutet mangelnder Speicherplatz.

Ihr Amiga 500 besitzt in der Grundausstattung einen Speicher von 512 KByte, was der achtfachen Kapazität des Vorgängers entspricht. In der Tat eine ganze Menge. Aufgrund verschiedener Eigenschaften ist der Amiga jedoch ein wahrer Speicherfresser. So setzen seine Grafikfähigkeiten und das Multitasking – also das Arbeiten mit mehreren Programmen gleichzeitig – dem »Überfluß« an Speicher bald ein jähes Ende. Mit Malprogrammen wie DPaint III oder Digipaint ist ohne Speicherausbaue kaum vernünftiges Arbeiten möglich. Vor allem wenn Sie Bilder mit hoher Auflösung – also vielen Bildpunkten – und mehreren Farben malen wollen, stoßen Sie schnell an die Grenzen des Speichers. Wenn Sie Vielschreiber sind und mit dem Computer lange Texte bearbeiten wollen, ist das Maß (oder der Speicher) schneller voll als Sie glauben. Wollen Sie gar eine Zeitung mit einem Desktop-Publishing-Programm gestalten, werden Sie kaum über ein, zwei Seiten hinauskommen, bevor Ihrem Amiga die Luft (bzw. der Speicher) ausgeht. Anwendungen im Video- oder Animationsbereich sind eben-

solche Speicherfresser wie musikalische Experimente auf dem Amiga. Selbst manchen Spielen sind die 512 KByte manchmal zu wenig.

Sie sehen, wofür Sie Ihren Amiga auch einsetzen, ein Ausbau des Arbeitsspeichers ist fast unumgänglich. Sicher wird der eine oder andere sich mit dem eingebauten Speicher zufriedengeben, oft wird jedoch sehr schnell der Ruf nach mehr laut. Egal für welche Anwendung Sie sich interessieren, um die Erweiterung des RAM-Speichers kommen Sie nicht herum.

Für den Speicherausbaue sind keine umständlichen Boxen notwendig, die dem Chaos auf dem Computertisch weiter Auftrieb geben, auch kommen keine störenden Kabel hinzu. Commodore macht es Ihnen leicht und hat bereits im Gehäuse des Amiga 500 Platz für eine Speichererweiterung vorgesehen. Zum Einbau einer Speicherkarte müssen Sie nicht einmal das Gehäuse des Computers öffnen. Ein kleiner Deckel an der Unterseite eröffnet Ihnen das Tor zu neuen Speicherdimensionen (Bild 1): Deckel auf, Karte einstecken, Deckel zu: fertig. Daß vor dem Einstecken der Karte der Amiga ausgeschaltet werden muß, sollten Sie nicht vergessen. Wieviel Speicher Ihnen der Amiga nach dem neuerlichen Einschalten zur Verfügung stellt, hängt von der Erweiterung ab, die Sie Ihrem Amiga zur Seite gestellt haben.

Grundsätzlich läßt sich die Speicherkapazität intern – also mittels des Einschubfaches – um bis zu 1,8 MByte erweitern, womit Ihnen am Ende 2,3 MByte an freiem Arbeitsspeicher zur Verfügung stehen. Soviel ist aber für die meisten Zwecke gar nicht nötig. Wieviel Speicher Sie letztendlich brauchen, hängt von individuellen Ansprüchen und Faktoren ab. Wir wollen Ihnen aber einige Tips geben, welcher Speicheraus-

Kaufhilfe Hardware

Stück für Stück

Erweiterungen für den Amiga gibt es wie Sand am Meer. Von Haupt- über Massenspeicher bis zu Grafikfähigkeiten können Sie alles Mögliche erweitern. Wir sagen Ihnen, was Sie wirklich brauchen.

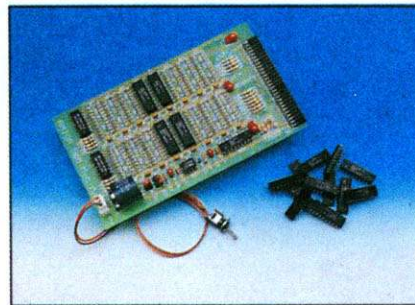


Bild 1. An der Unterseite des Amiga 500 befindet sich eine Öffnung für diese Speichererweiterungskarte

bau für welche Anwendungen sinnvoll und zweckmäßig ist.

Sind Sie vor allem an Textverarbeitung interessiert, dann sollten Sie den Speicher Ihres Computers um 512 KByte auf 1 MByte erweitern, um genügend Platz für Ihre Dokumente zu haben, das gleiche gilt für Malprogramme. Wollen Sie Animationen erstellen oder im Videobereich arbeiten, ist unter 1,5 MByte nicht viel zu wollen. Gleiches gilt für Anwendungen im DTP-Bereich. Musiker sollten ebenfalls den Speicher auf die-

ses Ausmaß erweitern, hier sind vor allem digitalisierte Klänge die Bittfresser. Setzen Sie Ihren Amiga für kaufmännische Anwendungen ein, wird im Normalfall 1 MByte Speicher ausreichend sein. Sie finden unsere Empfehlungen in Tabelle 1 noch einmal zusammengefaßt, sollten aber nicht vergessen, daß es sich dabei nur um gutgemeinte Vor-

schläge handelt, die nicht allgemeingültig sein können.

Haben Sie sich einmal für eine Konfiguration entschieden, so bleibt die Qual der Wahl. Unzählige Hersteller bieten Speicherkarten für den Amiga 500 an. Preislich reicht die Spanne etwa von rund 300 Mark für 512 KByte bis etwa 900 Mark für 1,8 MByte. Qualitativ bestehen zwischen den Karten kaum Unterschiede, lediglich in der Ausstattung ergeben sich Differenzen. Hier sollten Sie auf einige Merkmale achten, die aber fast alle Karten bieten. Erstens sollte die Erweiterung hardwaremäßig – mit einem Schalter – abschaltbar sein. Einige Programme verweigern nämlich die Zusammenarbeit mit dem Zusatzspeicher und bringen im Extremfall den Computer zum Absturz. Deshalb ist es notwendig, die Karte mit einem Schalter außer Betrieb setzen zu können. Die Speichererweiterung sollte außerdem mit einer akkugedufferten Echtzeituhr ausgestattet sein: Wie Sie vielleicht

Speicherausbaue

Anwendung	Mindestspeicher
Textverarbeitung	1 MByte
Malprogramme	1 MByte
Kaufm. Anwendungen	1 MByte
Animationen	1,5 MByte
DTP	1,5 MByte
Musik (Samples)	1,5 MByte
Video-Anwendungen	1,5 MByte

Tabelle 1. Sinnvolle Speicherkonfigurationen

festgestellt haben, verfügt der Amiga über eine eingebaute Uhr mit Kalenderfunktion. Leider läuft die Uhr nach Ausschalten des Computers nicht mehr weiter und muß beim nächsten Einschalten erneut gestellt werden. Verfügt die Speichererweiterung jedoch über eine batteriegepufferte Uhr, läuft diese auch nach dem Ausschalten weiter und braucht so nur einmal gestellt zu werden.

Sollten die intern maximal verfügbaren 2,3 MByte für Ihre Zwecke nicht ausreichen, müssen Sie sich mit einer externen Erweiterungsbox zufriedengeben. Diese hat den Vorteil, daß der Arbeitsspeicher auf bis zu 8 MByte erweiterbar bleibt.

Benötigen Sie diesen Speicherausbau (umfangreiche Animationen oder DTP-Anwendungen mit seitenreichen Dokumenten sind denkbare Kan-

didaten), dann sollten Sie dieser Form den Vorrang geben. Bedenken Sie ein weiteres Argument: Falls Sie sich die Option des späteren Aufrüstens offenhalten wollen, benötigen Sie eine Speicherkarte (auch intern), die erweiterbar ist.

Stapelweise Platten?

Sind Sie Diskjockey? Wahrscheinlich nicht. Bei der Arbeit mit Ihrem Amiga haben Sie sich aber bestimmt schon oft als solcher gefühlt. Mit unglaublicher Penetranz fordert der Amiga den Anwender auf, Diskette x einzulegen, um im nächsten Moment lieber Diskette y haben zu wollen, sich daraufhin aber doch wieder für x entscheidet. Nur äußerst geduldige Anwender werden auf Dauer mit die-

sen Situationen fertig, der Rest sucht sehr schnell nach Wegen, dieses Übel beiseite zu schaffen. Der Hauptgrund dafür liegt darin, daß der Amiga viele vom System benötigte Daten zu bestimmten Zeitpunkten von der Startdiskette nachlädt. Befindet sich aber gerade dann Ihre Datendiskette im Laufwerk, so haben Sie das Spiel wieder verloren und müssen die Diskette wechseln.

Der einfachste Weg, dem Diskjockey-Syndrom zu entkommen, ist die Anschaffung eines zweiten Disketten-Laufwerkes (Bild 2). Dieses bietet Ihnen die Möglichkeit, die Startdiskette im internen Laufwerk zu belassen und das neue Laufwerk mit Daten- oder anderen Disketten zu füttern. Auf diese Weise kann der Amiga jederzeit auf die von der Startdiskette benötigten Daten zurückgreifen. Um so manchen ungewollten Diskettenwechsel werden Sie trotzdem nicht herumkommen, diese reduzieren sich jedoch auf ein erträgliches Maß.

Wie bei den Speichererweiterungen gibt es auch bei den Disketten-Laufwerken eine große Auswahl. Auch hier bestehen qualitativ kaum Unterschiede, da die meisten auf demselben Basislaufwerk von der Firma NEC aufbauen. Differenzen gibt es neben Form und Farbe lediglich bei der Ausstattung. Achten Sie darauf, daß das Disketten-Laufwerk Ihrer Wahl auf jeden Fall abschaltbar ist, einige Programme erfordern dies. Ebenfalls wichtig ist, daß der Bus durchgeführt ist. Das heißt, daß sich am Laufwerk ein Stecker befindet, an dem spä-

ter ein weiteres Disketten-Laufwerk angeschlossen werden kann.

Schnörkel, wie ein sogenanntes Track-Diskplay, haben mehr optischen Effekt als praktischen Nutzen. Sie zeigen an, auf welcher Spur der Schreib-/Lesekopf des Laufwerkes sich gerade befindet. Im Normalfall ist diese Information aber völlig nebensächlich.

Haben Sie den Speicher auf das entsprechende Maß erweitert und sind Sie stolzer Besitzer eines zweiten Disketten-Laufwerkes, verfügen Sie über eine Konfiguration, mit der recht vernünftig gearbeitet werden kann. Ohne diese Grundausstattung wird die Arbeit mit dem Amiga aber schnell zum Ärgernis und Lust zu Frust. Alle weiteren besprochenen Zusatzgeräte sind bei der Arbeit mit dem Amiga nützlich, aber nicht immer notwendig oder nur für bestimmte Anwendungen sinnvoll.

Speicher-millionäre

So verhält es sich etwa mit Festplatten. Diese alternativen Speichermedien, die mit großen Banketten nur den Namen gemeinsam haben, sind vor allem beim Arbeiten mit großen Datenmengen interessant. Sie sind im Grunde nichts anderes als Disketten-Laufwerke mit hohem Speichervermögen und bieten üblicherweise (je nach Größe) die Kapazität von 30 bis 150 Disketten in einer Einheit. Ein weiterer Unterschied besteht darin, daß Festplatten im allgemeinen fest eingebaut sind und daher nicht gewechselt werden. Dabei sind die Zugriffszeiten (die Lade- und Speicherdauer) im Vergleich zu herkömmlichen Laufwerken wesentlich kürzer. Da sich die Frage nach einer Festplatte für die meisten Anwender aus Preisgründen (eine gute Festplatte ist im Moment kaum unter 1000 Mark zu bekommen) erst später stellt, wollen wir an dieser Stelle nicht näher darauf eingehen.

Fester Bestandteil eines jeden Computersystems ist aber sicherlich ein Drucker. Schließlich will man seine Ergebnisse nicht nur auf dem Schirm betrachten, sondern auch »hand-feste« Resultate erhalten. Anwendungen wie Textverarbeitung »schreien« geradezu nach einem Drucker. Grundsätzlich läßt sich jeder Drucker, der einen parallelen Anschluß - ge-

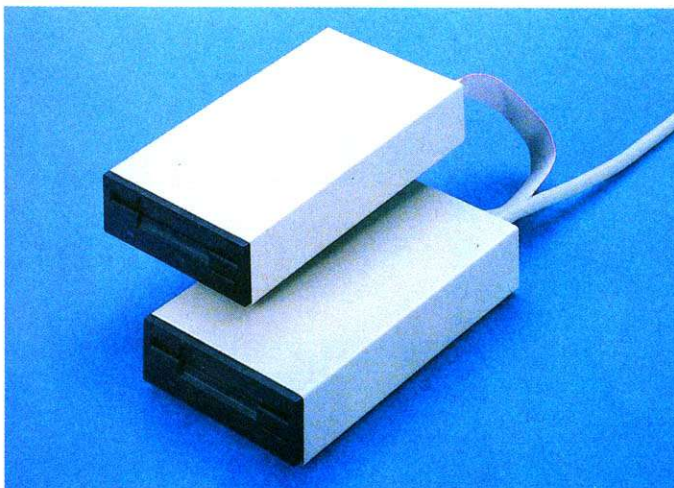


Bild 2. Ein zweites Disketten-Laufwerk macht die Arbeit mit dem Amiga wesentlich einfacher

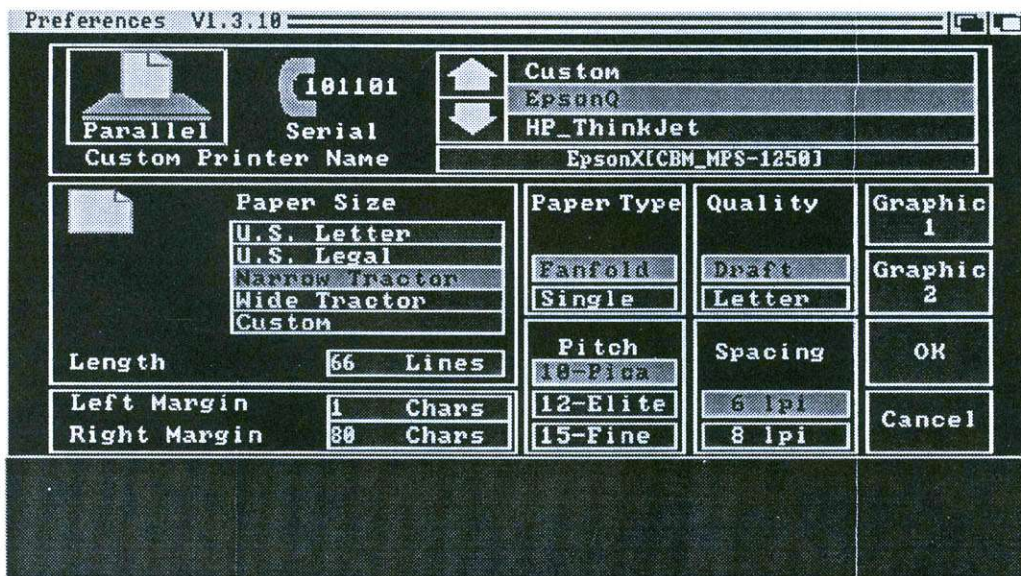


Bild 3. In den »Preferences« stellen Sie Ihren Druckertyp ein

Deluxe Paint III

Als die Bilder laufen lernten



DELUXE PAINT

DPaint II war bisher das leistungsfähigste Malprogramm auf dem Amiga. Diese Zeiten sind glücklicherweise vorbei, denn nun gibt es Deluxe Paint III! Ob Sie Zeichentrick-Pionier im eigenen Heimkino sind oder fernsehgerechte Vorspanne für Ihre Videofilme erzeugen wollen, ob Sie Grafik beruflich benötigen oder als Hobbymaler auf den Spuren der großen Meister wandeln: Mit Deluxe Paint III eröffnen sich Ihnen Möglichkeiten, die Sie bisher für unmöglich gehalten haben. Was das Programm so alles kann?


Markt & Technik
Zeitschriften · Bücher
Software · Schulung

- Alle Funktionen von Deluxe Paint II
- Extra Halfbrite-Modus für 64 Farben
- Beliebige Definition und Handhabung von Pinseln
- Spezielle Maleffekte wie Schattieren, Verschmieren, Weichzeichnen, Farbverlauf, Sprühdose
- Neue Füllfunktionen
- Exaktes Zeichnen geometrischer Formen
- Schnelle Perspektive-Funktion für 3-D-Grafiken
- Alle Amiga-Zeichensätze sowie spezielle Color-Fonts nutzbar
- Overscan-Unterstützung
- Filme aus Einzelbildern erstellen
- Pinselanimation: Jeder Teil eines Filmes kann als Pinsel ausgeschnitten werden und ist selbst wieder ein kleiner Film
- Bewegen-Option: Jeder Pinsel kann in jede beliebige Richtung bewegt und dreidimensional gedreht werden, ja sogar bei der Bewegung Spuren hinterlassen. DPaint III macht daraus automatisch einen Film.

- 1 Mbyte RAM reicht für kreative Animationen
- ANIM-Speicherformat – Kompatibilität zu anderen Animationsprogrammen

Hardware-Anforderungen:

Amiga mit mindestens 1 Mbyte RAM.

Deluxe Paint III deutsch

Bestell-Nr.: 54138

DM 249,- (sFr 225,-*/öS 2490,-*)

Update von Deluxe Paint II auf Deluxe Paint III

(gegen Einsendung der Originaldiskette und Verrechnungsscheck)

Bestell-Nr. 54138U

DM 99,-* (sFr 89,-*/öS 890,-*)

Deluxe Paint III Demo

Demo-Versionen erhalten Sie gegen Vorkasse direkt beim Verlag

Bestell-Nr. W718

DM 15,-* (sFr 15,-*/öS 100,-)

Deluxe Print II

Bestell-Nr. 52582

DM 199,-* (sFr 179,-*/öS 1990,-*)



ELECTRONIC ARTS®

Deluxe Photolab deutsch

Bestell-Nr. 54112
DM 249,-* (sFr 224,-*/öS 2190,-*)

Update Photolab englisch auf deutsch

Bestell-Nr. 54112 U
DM 49,-* (sFr 49,-*/öS 490,-*)

Deluxe Video 1.2 deutsch

Bestell-Nr. 52583
DM 249,-* (sFr 225,-*/öS 2190,-*)

Zusatzpakete zu Video:

Seasons & Holidays

Bestell-Nr. 52580

Deluxe Art Parts

Bestell-Nr. 52581
je DM 29,-* (sFr 26,-*/öS 290,-*)

Deluxe Music deutsch

Bestell-Nr. 52579
DM 199,-* (sFr 179,-*/öS 1990,-*)

*Unverbindliche Preisempfehlung

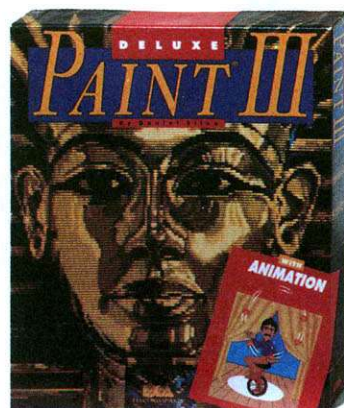
Zusatzpakete zu Musik:

It's only Rock'n'Roll

Bestell-Nr. 54115

Hot & Cool Jazz Bestell-Nr. 54116

je DM 29,-* (sFr 26,-*/öS 290,-*)



Markt & Technik-Bücher und -Software erhalten Sie in den Fachabteilungen der Warenhäuser, im Versandhandel, in Computer-Fachgeschäften oder bei Ihrem Buchhändler.

INFO-COUPON

Bitte senden Sie mir weitere Informationen zu Deluxe Paint III

Name _____

Straße _____

PLZ/Ort _____

Bitte ausschneiden und einsenden an:
 Markt & Technik Verlag AG, Buch- und Software-Verlag,
 Hans-Pinsel-Straße 2, 8013 Haar bei München
 Ami SH 8

nauer gesagt eine sogenannte Centronics-Schnittstelle – besitzt, an den Amiga anschließen. Gleiches gilt für solche mit seriellem Anschluß, genauer der sogenannten RS232-Schnittstelle. Den jeweiligen Drucker-typ wählen Sie einfach mittels der Preferences-Einstellungen (Bild 3) von der Workbench (auf Seite 46 finden Sie einen Artikel, der sich mit diesem Thema beschäftigt).

Am weitesten verbreitet sind mit gutem Grund noch immer die sogenannten Matrixdrucker (auch Nadeldrucker genannt). Diese sind preislich erschwinglich und liefern durchaus brauchbare Ergebnisse. Dabei geht die Tendenz wegen der eindeutig besseren Druckqualität in der letzten Zeit immer mehr zum 24-Nadel-Drucker im Gegensatz zum älteren System, das mit neun Nadeln druckte. Wie der Name schon sagt, arbeiten diese mit 24 Nadeln, um den Ausdruck punktweise – Text oder Grafik – zu erstellen. Sie sind dadurch nicht

tern »geparkt« werden und braucht nicht immer neu eingefädelt zu werden) und vielfältige Bedientasten machen ihn zum Favoriten seiner Preisklasse. Apropos Preis: mit seinem Listenpreis von rund 900 Mark schlägt er schon seine Konkurrenten, daß er von verschiedenen Händlern schon unter 700 Mark angeboten wird, macht ihn noch attraktiver.

Der Anschluß des LC 24-10 an den Amiga erfolgt – genauso wie der anderer Druckermodelle – einfach mittels eines Standard-Parallelkabels. Neben Text kann der LC 24-10 auch Grafiken ausdrucken, allerdings nur in Schwarzweiß, Farben werden in entsprechende Grauwerte umgesetzt. In der Praxis zeigt sich, daß man farbige Ausdrücke nur sehr selten benötigt. Wer's trotzdem gern farbig haben möchte, benötigt einen Drucker mit entsprechenden Qualitäten. Als Standard unter den Farbdruckern hat sich der CP6 von NEC durchgesetzt. Zu einem Preis von rund

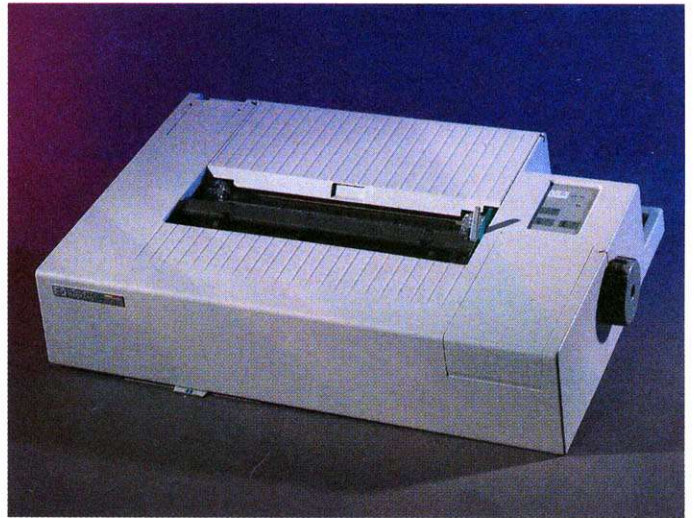


Bild 6. Tintenstrahldrucker bringen die Farben wesentlich kräftiger zu Papier

1400 Mark bietet er Ausdrücke in allen Farben des Regenbogens. Günstiger – und auch in Farbe – druckt der LC 10-C von Star. Zu einem Preis von etwa 700 Mark versucht er, den Bildschirm des Amiga in Farbe zu kopieren – allerdings in der Qualität eines 9-Nadel-Druckers (Bild 5).

Leider hängt die Qualität eines Farbausdrucks auf einem Matrixdrucker sehr stark vom Zustand des Farbbaus ab. Da sich die verschiedenen Farben, je nach Verwendung, unterschiedlich schnell verbrauchen, läßt der Ausdruck schon sehr bald zu wünschen übrig. Ganz abgesehen davon, schlägt ein mehrfarbiges Farbband mit bis zu 60 Mark zu Buche.

Wesentlich hochwertiger sind die Ausdrücke eines farbigen Tintenstrahldruckers, wie des »Paintjet+« von Hewlett-Packard. Durch die Verwendung von feinen Düsen zum Farbauftrag anstatt Nadeln, werden die Farben ungleich kräftiger und

gleichmäßiger. Bild 6 zeigt zum Vergleich einen Beispielausdruck des HP Paintjet+. Diese Qualität muß aber auch bezahlt werden: ein farbfähiger Tintenstrahldrucker ist nicht unter 3000 Mark zu haben und daher für den Durchschnittsanwender häufig von vorneherein uninteressant.

Ebenso verhält es sich mit Laserdruckern. Diese sind derzeit kaum unter 4000 Mark zu haben. Bedenkt man, daß dies etwa der fünffache Preis des Amiga ist, so wird diese qualitativ hochwertige Form des Druckens schnell wieder uninteressant.

Klänge aus der Retorte

Eine nützliche und günstige Hardware-Erweiterung für Musikfreunde ist ein sogenannter Sound-Digitizer (Bild 7). Damit können beliebige Klänge in den Amiga übertragen und ausgegeben werden. Der Amiga ist mit einem leistungsfähigen Chip zur Klangerzeugung versehen. Dieser begnügt sich nicht damit, Töne rein synthetisch – per Software – zu erzeugen, sondern arbeitet mit der Sampling-Methode. Das bedeutet, daß analoge, natürliche Klänge digitalisiert und wiedergegeben werden können. Digitalisiert heißt, die analogen Signalwerte werden in entsprechende (digitale) Zahlenwerte umgesetzt. Bei der Wiedergabe werden aus den Zahlen wieder analoge, akustisch wahrnehmbare Signale.

Sound-Digitizer werden von etwa 80 bis 200 Mark angeboten. Bastler können wesentlich



Bild 4. Der Star LC 24-10 macht Ihrem Amiga Druck

nur schneller als ihre Kollegen mit neun Nadeln, sondern liefern auch bessere Ergebnisse. Waren 24-Nadler bis vor rund zwei Jahren für den Heim-anwender unerschwinglich, sinken die Anschaffungskosten in letzter Zeit rapide.

Ein Drucker sticht dabei mit seinem besonderen Preis-Leistungs-Verhältnis aus dem vielfältigen Angebot hervor: der LC 24-10 von Star Micronics (Bild 4). Der attraktive 24-Nadler beherrscht vier verschiedene Schönschriften und druckt mit hoher Geschwindigkeit. Features wie Papierparkfunktion (Endlospapier kann bei zeitweiser Verwendung von Einzelblät-

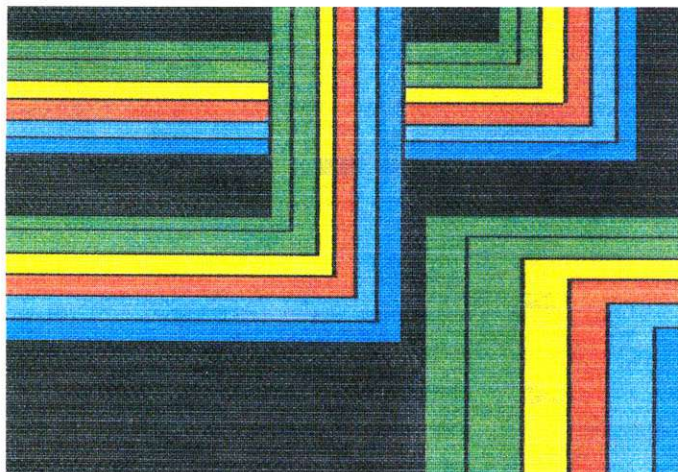


Bild 5. Ein Farbausdruck mit einem Matrixdrucker

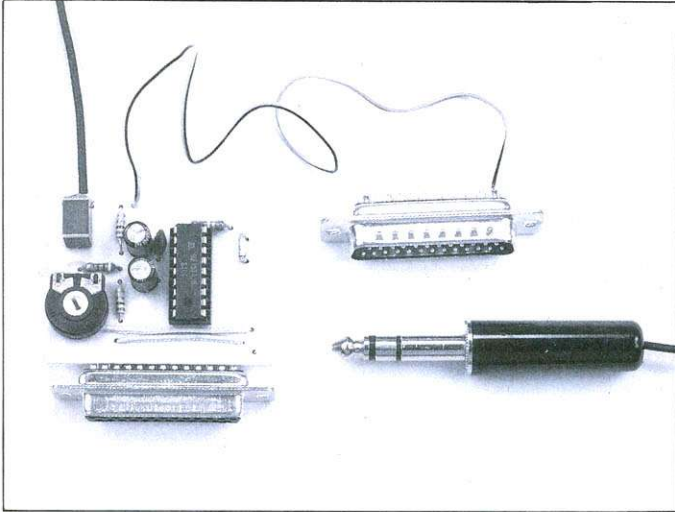


Bild 7. Ein Sound-Digitizer holt den Klang in den Computer

preisgünstiger den Digitizer aus unserem Sonderheft 1 nachbauen. Die teureren Modelle bieten meist die Möglichkeit, in Stereo zu digitalisieren und arbeiten mit einer höheren Sampling-Rate, was Klangverbesserungen mit sich bringt. Für den Hausgebrauch reicht aber ein einfaches Modell ohne weiteres aus. Zur Digitalisierung wird der Digitizer mit einer Klangquelle (z. B. Stereoanlage) verbunden. Mit entsprechender Software (siehe auch Überblick Software, Seite 18), die bei manchen Digitizern bereits zum Lieferumfang gehört, können Klänge nun »gesampelt« und bearbeitet werden.

Nebenbei bemerkt: Um in den vollen Genuß der Klänge

Ihres Amiga zu kommen, sollten Sie diesen mit einer Stereoanlage verbinden. Dies geschieht mit einem einfachen Stereo-Kabel mit Chinchsteckern – erhältlich im Elektrofachhandel. Verbinden Sie mit Hilfe dieses Kabels die beiden Buchsen neben dem Mausanschluß mit dem Eingang Ihres Verstärkers. Besitzt dieser keine Chinch-Buchsen, hilft Ihr Fachhändler mit entsprechenden Adaptern sicher gerne aus.

SO

Martin Jobst studiert Publizistik und Kommunikationswissenschaft in Salzburg und arbeitet seit drei Jahren als freier Mitarbeiter für Zeitschriften des Markt & Technik Verlages. Sie können ihn erreichen über unsere Verlagsanschrift (Markt & Technik Verlag AG, Redaktion Sonderhefte, z. Hd. Martin Jobst, Hans-Pinsel-Straße 2, 8013 Haar bei München.)

Kaufhilfe Software

Führer im Software- Dschungel

Als Neuling am Amiga werden Sie vom Software-Angebot geradezu überwältigt. Wir helfen bei der Auswahl und sagen Ihnen, welche Programme sich im praktischen Einsatz bewährt haben: Orientierung im Software-Dschungel.

von Martin Jobst

Wie in so vielen Bereichen finden sich auch unter Software-Produkten jede Menge schwarzer Schafe. Viele Programme sind gänzlich unbrauchbar oder nur sehr eingeschränkt einsetzbar. Was die Werbung verspricht, erweist sich oft als glatte Lüge.

Wir wollen Ihnen den Tritt ins Fettnäpfchen ersparen und stellen Ihnen im ersten Teil dieses Beitrags Programme aus verschiedenen Bereichen vor, die Sie getrost kaufen können. Dabei ist die Auswahl speziell auf den Einsteiger zugeschnitten. Natürlich werden wir auch auf eventuelle Nachteile der Programme eingehen. Die Auswahl kann als Grundausrüstung für den Amiga-Anwender verstanden werden.

Im Anschluß daran finden Sie eine Übersicht über rund 20 gängige Amiga-Programme samt Kurzbeschreibung. Das Auswahlkriterium war auch hier die Praxistauglichkeit der Programme. Da es sich bei der Kürze nur um Überblicksdaten handelt, erhalten Sie neben der Preisangabe auch den Verweis auf den jeweiligen Testbericht im Amiga-Magazin samt Testnote. Der Überblick soll Ihnen zur Orientierung dienen und über den Funktionsumfang der einzelnen Programme Aufschluß geben.

Doch vorerst einige grundsätzliche Worte zum Software-Kauf. Überzeugen Sie sich vor dem Kauf eines Programms von dessen Qualitäten durch einen ausgiebigen Test bei Ihrem Händler. Pochen Sie auf Ihr Recht, das Produkt gründlich prüfen zu können und kaufen Sie niemals die Katze im Sack. Üblicherweise kauft man nicht einmal Schallplatten ohne vorherige Hörprobe. Bei Computer-Programmen ist ein Check aber wesentlich wichtiger: Erstens wollen Sie mit dem Programm ja vernünftig arbeiten und zweitens ist dieses ja auch meistens nicht gerade billig. Gerade im Zeitalter von Software-Discountern nimmt das Katze-im-Sack-Syndrom immer mehr überhand. Auch wenn Sie das Programm von einem Testbericht her zu kennen glauben: Erst ein persönlicher Blick darauf zeigt, ob es wirklich auch für Sie das ideale Instrument darstellt.

Stürzen wir uns gleich auf ein heikles Thema: Amiga und Textverarbeitung. Damit scheinen

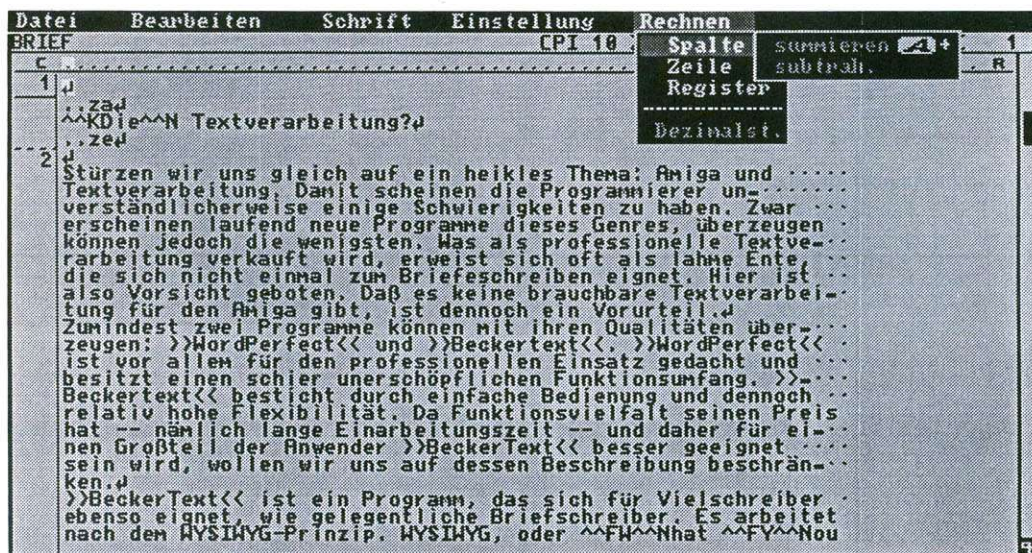


Bild 1. »Beckertext« ist einfach zu bedienen und bietet dennoch vielfältige Möglichkeiten

die Programmierer unverständlicherweise einige Schwierigkeiten zu haben. Zwar erscheinen laufend neue Programme dieses Genres, überzeugen können jedoch die wenigsten. Was als professionelle Textverarbeitung verkauft wird, erweist sich oft genug als lahme Ente, die sich nicht einmal zum Briefeschreiben eignet. Hier ist also Vorsicht geboten. Daß es keine brauchbare Textverarbeitung für den Amiga gibt, ist dennoch ein Vorurteil.

verwaltung, automatisches Stichwort- und Inhaltsverzeichnis oder einen Thesaurus (Synonymlexikon) nicht verzichten können, sind mit Wordperfect bestens bedient.

»Beckertext« ist ein Programm, das sich für Vielschreiber ebenso eignet, wie für gelegentliche Briefschreiber. Es arbeitet nach dem WYSIWYG-Prinzip. WYSIWYG, oder What You See Is What You Get, heißt: Der Text erscheint auf dem Bildschirm so, wie später im

»Beckertext« beherrscht alle Grundfunktionen einer Textverarbeitung: Kopieren, Löschen und Verschieben von Textteilen, Such- und Ersetz-Funktionen, Verwendung verschiedener Schriftarten (kursiv, unterstrichen, fett etc.), variabler Zeilenabstand, Blocksatz usw. Daneben besitzt es noch einige zusätzliche Features. So können mehrere Texte in verschiedenen Fenstern gleichzeitig bearbeitet werden – eine Option, die für viele Anwendungen nützlich ist. Ebenfalls brauchbar ist die eingebaute Trennautomatik, die Wörter am Ende der Zeile trennt. Nur sollten Sie dieser nicht uneingeschränkt vertrauen, da sie nicht ganz fehlerfrei arbeitet. Dies liegt daran, daß die Funktion nur mit Regeln und ohne Ausnahmexikon arbeitet. Da die deutsche Sprache bei der Silbentrennung oft sehr eigen ist, kommt es manchmal zu falschen Trennungen.

Weitere interessante Elemente von »Beckertext« sind die automatische Erstellung ei-

Eine große Hilfe bei der Korrektur von Texten ist die eingebaute Lexikon-Funktion. Damit können Sie Ihre Texte auf Fehler überprüfen lassen. Das Lexikon ist lernfähig und kann durch neue Wörter ergänzt werden. Leider liefert Data Becker das Lexikon mit einem sehr beschränkten Wortschatz aus, so daß es erst nach Erweiterung sinnvoll einsetzbar ist.

Wort perfekt mit Beckertext?

Die Benutzerführung von »Beckertext« ist komplett in Deutsch (Bild 1), was die Arbeit wesentlich vereinfacht. Die Arbeitsgeschwindigkeit ist durchaus akzeptabel, der Bildaufbau relativ flott. Die Bedienung wird durch die Num-Lock-Funktion noch beschleunigt: Nach dem Vorbild von Programmen aus dem MS-DOS-Bereich, ist der abgesetzte Ziffernblock mit verschiedenen Editier-Funktionen belegt. So kann neben einfachen Cursor-Bewegungen auch seitenweise geblättert werden oder der Cursor per Tastendruck an den Textanfang gebracht werden etc.

»Beckertext« wird zusammen mit einem ausführlichen, sehr gut verständlichen, deutschen Handbuch ausgeliefert und kostet 198 Mark. Trotz einiger Unzulänglichkeiten wie einer fehlenden Fußnotenverwaltung oder den umständlichen Lade-/Speicher-Funktionen ist »Beckertext« die Standard-Textverarbeitung für den Amiga. Standard heißt: für viele Zwecke einsetzbar und unkompliziert in der Bedienung.

Bei den Malprogrammen hat sich seit den Urzeiten des Amiga ein Programm an der Spitze halten können: »Deluxe Paint« oder kurz »DPaint«. Trug die erste Version einst zur Popularität des neuen Supercomputers Amiga bei, haben auch die Nachfolger gute Karten. »DPaint II« entwickelte sich zur Referenz bei den Malprogrammen und das kürzlich erschienene »DPaint III« (Bild 2) ist auf dem besten Weg, in die Fußstapfen seiner Vorgänger zu treten.

Mit »DPaint« wird Ihr Computer zum universellen Malwerkzeug. Es eröffnet Möglichkeiten, die mit herkömmlichen Medien nicht realisierbar sind. »DPaint« zeigt, daß Grafikprogramme mehr können, als Linien und Kreise ziehen. Wir wollen Ihnen einige der Möglichkeiten der »DPaint«-Serie vor-



Bild 2. »DPaint III« ist das Malprogramm für den Amiga

Zumindest zwei Programme können mit ihren Qualitäten überzeugen: »Wordperfect« und »Beckertext«. »Wordperfect« ist vor allem für den professionellen Einsatz gedacht und besitzt einen schier unerschöpflichen Funktionsumfang. »Beckertext« besticht durch einfache Bedienung und dennoch relativ hohe Flexibilität. Da Funktionsvielfalt seinen Preis hat – nämlich lange Einarbeitungszeit – und daher für einen Großteil der Anwender »Beckertext« besser geeignet sein wird, wollen wir uns auf dessen Beschreibung beschränken. Dennoch möchten wir an dieser Stelle betonen, daß Wordperfect zweifellos das Programm für den professionellen Einsatz ist. Studenten, Autoren und andere Leute, die auf Funktionen wie Fußnoten-

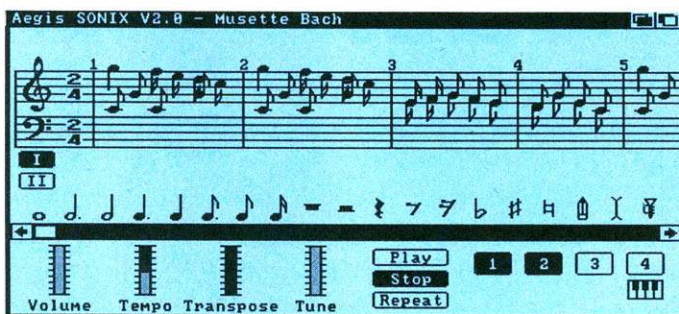


Bild 4. »Sonix« macht den Amiga zur Musikmaschine

Ausdruck. Kursiv oder fett geschriebene Buchstaben sieht man bereits am Bildschirm wie die Formatierung (Rand etc.) des Textes. Das WYSIWYG-Prinzip wird von »Beckertext« konsequent eingehalten, nur Seitenzahl und Kopf-/Fuß-Zeilen (siehe unten) erscheinen nicht auf dem Schirm.

nes Inhaltsverzeichnisses oder einer Indexliste. Dazu müssen die Wörter im Text markiert werden, den Rest macht das Programm. Ebenso wie die laufende Seitennummer können am Anfang und/oder am Ende der Seiten Kopf- bzw. Fußzeilen stehen, gleichbleibende Texte wie etwa der Titel einer Arbeit etc.

Eine Rechenoption ermöglicht das Eingliedern von Rechenergebnissen in Texte. Auch für die Einbindung von Bildern ist »Beckertext« geeignet. Dabei hilft das mitgelieferte Programm »BtSnap«, mit dem Bildteile aus anderen Programmen/Bildern ausgeschnitten und anschließend in »Beckertext« verwendet werden können. Die Bilder können an eine beliebige Stelle im Text gesetzt werden, lassen sich allerdings nur horizontal verschieben. Ebenso kann nur ober- und unterhalb der Bilder Text sein, nicht aber daneben.

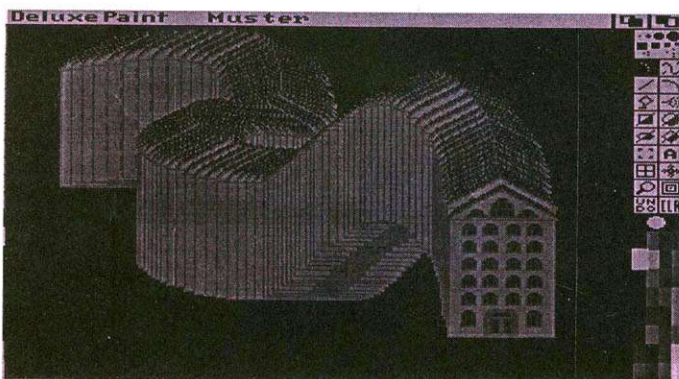


Bild 3. Mit selbstdefinierten »Brushes« lassen sich interessante Effekte erzielen

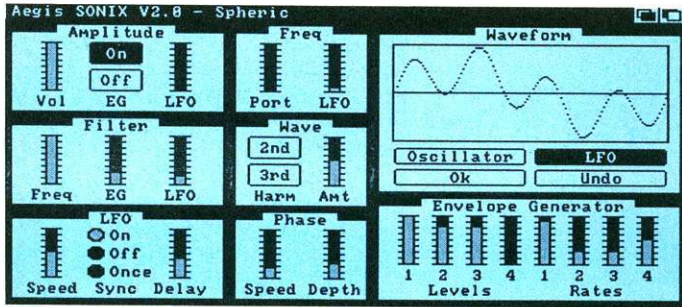


Bild 5. Der Synthesizer von Sonix macht den Klang

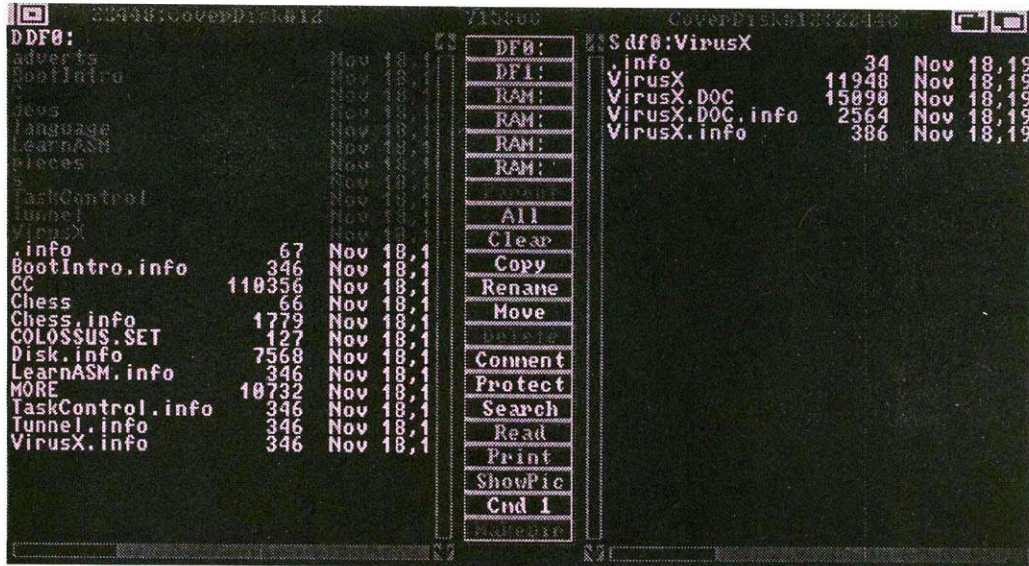


Bild 6. Die komfortable Oberfläche von »Diskmaster« erspart Ihnen den Ärger mit dem CLI

stellen, die aufgrund der Funktionsvielfalt nur eine kleine Auswahl darstellen.

Eine der interessantesten Fähigkeiten ist das Malen mit Brushes (engl.: Pinsel). Diese Pinsel können nicht nur herkömmlicher Art sein, Sie können auch Ausschnitte einer Grafik als Pinsel verwenden. Damit lassen sich interessante Effekte erzielen (Bild 3). Der Pinsel kann in seiner Größe beliebig verändert, gedreht oder seine Perspektive manipuliert werden. Mit Weichzeichnern oder Schmiereffekten läßt sich der Ausdruck eines Bildes verändern.

Gemalt wird mit der Maus. Neben den oben erwähnten Brushes, können auch »normale« Pinsel in verschiedenen Stärken, eckig oder rund, verwendet werden. Kreise, Linien oder Polygone lassen sich leicht mit Hilfe entsprechender Funktionen realisieren. Eine Sprühfunktion ermöglicht den sog. Airbrush-Effekt. Wie mit einer Farbspritz-Pistole können die Punkte in unterschiedlichen Dichten aufgetragen werden. Eine Textfunktion zur Schrifteinbindung gehört ebenso zur Ausstattung, wie die Lupe zum exakten Nachbearbeiten von Bildteilen. Mittels einer Sym-

metrie-Funktion können kaleidoskopähnliche Effekte erzeugt werden. Mit der Füll-Option können eingeschlossene Flächen sowohl einfarbig als auch mit einem Farbverlauf ausgemalt werden. Farbanimationen schließlich ermöglichen den zyklischen Wechsel der Farben, wodurch interessante Bewegungseffekte erzielt werden.

»DPaint II« arbeitet mit maximal 32 Farben in allen gängigen Auflösungen. »DPaint III« eröffnet zusätzlich die Möglichkeit im Extra-Halbrite-Modus

mit 64 verschiedenen Farben zu arbeiten. Außerdem können Sie mit der neuesten Version von »DPaint« auf einfache Weise Animationen erstellen. »DPaint« arbeitet dabei mit dem Konzept des Daumenkinos. Beim Abspielen des Films werden also viele Seiten schnell »durchgeblättert«. Bei der Erstellung der Animation steht Ihnen »DPaint« mit einer Fülle Hilfsmittel zur Seite. Es hilft Ih-

Von den Grafikkünsten zu einer weiteren Spezialität des Amiga: der Klangerzeugung. Aus der breiten Masse der mittlerweile angebotenen – teilweise sehr speziellen – Software aus diesem Bereich, sticht ein Programm mit seinen universellen Fähigkeiten hervor. Es trägt den Namen »Sonix« und kommt aus dem Hause Aegis. Wie »DPaint« ist auch »Sonix« bereits einige Jahre am Markt vertreten. Aufgrund seiner Stärken ist es auf seinem Gebiet nach wie vor konkurrenzlos.

Sonix vereint die fantastischen Möglichkeiten eines Synthesizers mit denen eines Kompositionsprogramms. Es verwöhnt den Musiker mit seiner einfachen Bedienung und den vielfältigen Möglichkeiten.

Für das Arrangieren von Musikstücken stehen Ihnen acht Spuren zur Verfügung, wobei vier mit den internen Tonkanälen des Amiga belegt werden. Die gewünschten Tonhöhen und -längen setzen Sie einfach mit der Maus auf die Notenzeilen (Bild 4). Sonix verfügt über ein großes Repertoire an Notensymbolen, wenn auch nicht groß genug, um den Ansprüchen klassischer Stücke zu genügen. Zum Editieren der Stücke stehen Ihnen viele Funktionen zur Verfügung, ebenso können Sie das Abspielen Ihres Meisterwerks beeinflussen.

Als Instrumente können Sie beliebige digitalisierte Instrumente verwenden. Sie können aber auch mit dem eingebauten Synthesizer eigene Klänge erzeugen. Auf Basis eines analogen Synthesizers bietet Ihnen Sonix hier vielfältige Möglichkeiten zur Klangerzeugung. Neben dem Lautstärkenverlauf können Sie die Grundschwingung des Klanges ebenso beeinflussen wie etwa Obertöne oder Filterwerte. Die einzelnen Werte können Sie mit der Maus über die vielen Regler und Schalter manipulieren (Bild 5). Die Grundschwingung kann mit der Maus sogar gezeichnet werden. (Die Möglichkeit, die Form der Grundschwingung frei zu bestimmen, bieten normalerweise nur Synthesizer der oberen Preisklasse.)

Sonix verfügt über eine MIDI-Option, die den Datenaustausch mit einem Synthesizer erlaubt. Die Verbreitung von Sonix hat dazu geführt, daß Musikstücke für dieses Programm auf Public Domain-Disketten verfügbar sind. Werfen Sie einmal ein Ohr auf Diskette 47 der Serie Chiron Conceptions!

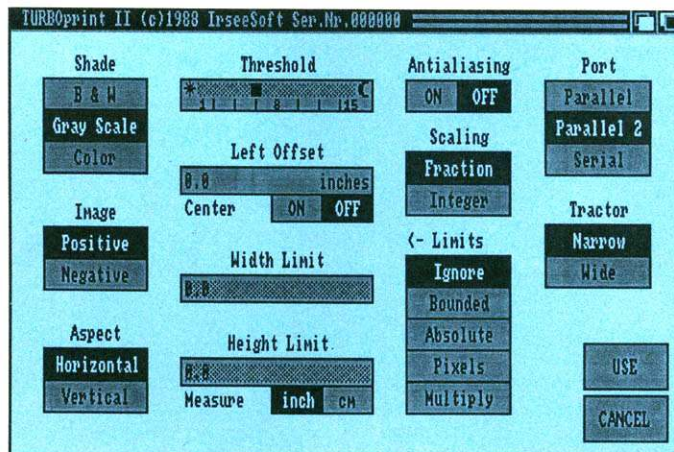


Bild 7. Die Preferences von »TurboPrint II« bieten viele Einstellmöglichkeiten

K AUFBERATUNG

Sonix wird zusammen mit einem ausführlichen Handbuch und einigen Musikbeispielen ausgeliefert und kostet rund 100 Mark. Es ist nicht nur der Klassiker unter den Musikprogrammen, sondern ist für jeden, der sich für Musik und Amiga – wenn auch nur im entferntesten – interessiert, das Programm. Lassen Sie die Soundfähigkeiten Ihres Computers nicht brachliegen.

»Move« können Dateien verschoben werden: Befindet sich beispielsweise ein File nicht im korrekten Verzeichnis, können Sie es mit diesem Befehl in ein anderes transferieren, ohne es mühsam im »alten« entfernen zu müssen.

Eine weitere Besonderheit an »Diskmaster« ist die Möglichkeit, den Inhalt von Verzeichnissen auszudrucken. Ebenso ungewöhnlich ist die

schirmfarben Ihren Bedürfnissen anpassen. Auf Wunsch (und wenn Sie das Public-Domain-Programm »Arc« haben) können Files auch gepackt werden. Gepackte Files verbrauchen auf Diskette wesentlich weniger Speicherplatz. Allerdings müssen diese vor der Verwendung wieder entpackt werden, was »Diskmaster« ebenfalls beherrscht.

»Diskmaster« kostet 120 Mark und ist in seiner Sparte das derzeit beste Programm. Es sei jedem Anwender zur täglichen Arbeit empfohlen.

Wenn Sie den Wunsch haben, Ihre Lieblingsgrafik nicht nur am Bildschirm zu betrachten, sondern auf Papier zu bannen, benötigen Sie ein entsprechendes Druckprogramm. Zwar ist es von der Workbench und auch von vielen Programmen aus möglich, Bilder zu

werden müssen. »TurboPrint II« geht einen Schritt weiter. Da das Programm – einmal gestartet – im Hintergrund weiterläuft können aus jedem laufenden Programm Bilder mittels einer Tastenkombination ausgedruckt werden. Ein weiteres Tastenkürzel ermöglicht das Ablegen des momentan angezeigten Bildes auf Diskette oder RAM-Disk. Auf Wunsch können diese Operationen auch nur mit einem bestimmten Ausschnitt des Bildschirms geschehen.

Doch damit nicht genug: »TurboPrint II« verbessert auch die Druckfunktionen anderer Programme. Dies wird dadurch ermöglicht, daß es im Hintergrund arbeitet, und so die Drucktreiber der Workbench ersetzt. Die Qualität der Ausdrucke wird wesentlich verbessert und auch die Geschwindigkeit merklich erhöht. Vor allem dadurch, daß Sie das Programm optimal an den eigenen Drucker anpassen können. Dazu dient ein eigenes Preferences-Programm zu »TurboPrint II«.

Mit diesen Preferences (Bild 8) können Sie die Auflösung Ihres Druckers optimal ausnutzen. Neben der Wahl verschiedener Drucktreiber für die Auflösung kann auch das Muster bestimmt werden, mit dem die Punkte der Grafiken auf Papier gebracht werden. Genauso können Sie das Verhältnis Höhe/Breite verändern sowie Helligkeit und Kontrast des Ausdruckes variieren.

Ein zweites Fenster (Bild 7) bietet weitere Einstellmöglichkeiten. Hier können Sie die Maße des Bildes verändern oder bestimmen, ob der Ausdruck invertiert erfolgen soll.

»TurboPrint II« überzeugt durch seine Flexibilität und gehört in jede Softwaresammlung. Gerade die Möglichkeit, die Druckqualität anderer Programme zu verbessern und die Workbench-Treiber zu ersetzen, macht es interessant. So manchem Programm, das aufgrund seiner schlechten Druckqualität unbrauchbar ist, verhilft es so zu neuen Ehren.

Auf den folgenden Seiten finden Sie eine Auswahl weiterer Programme, die von der Qualität her überzeugen können. Sie soll Ihnen Orientierung im Software-Dschungel verschaffen. Noch ein Wort zu den Testnoten: Die maximale bei AMIGA-Tests vergebene Punktzahl ist 12. Die Testnoten am Ende des Programms geben die erreichte Punktzahl an. Viel Spaß bei Ihrem Einkaufsbummel. so

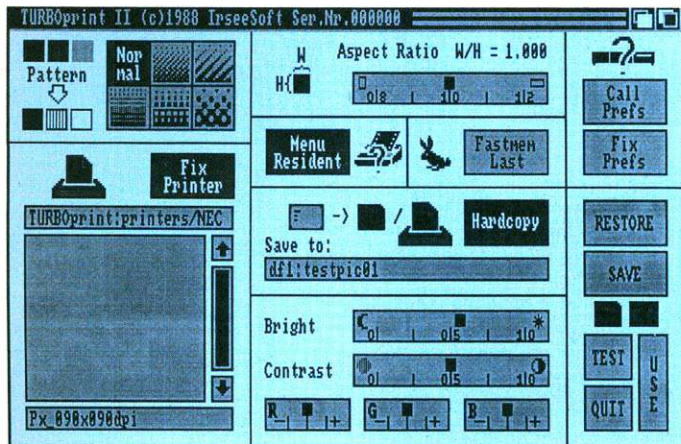


Bild 8. »TurboPrint II« holt das Letzte aus Ihrem Drucker heraus

Jeder Amiga-Anwender ärgert sich früher oder später über das CLI oder Command Line Interpreter. Dennoch ist es bei der Arbeit mit dem Amiga unerlässlich: Dateien müssen kopiert, Files umbenannt werden usw. Durch den fehlenden – von der Workbench jedoch gewohnten – Komfort wird der Umgang mit dem CLI meist zur nervenden Tipperei. Daß es auch anders geht, zeigen Benutzeroberflächen für das CLI, die den Umgang mit Dateioperationen wesentlich vereinfachen. Unter den zur Zeit angebotenen Programmen hebt sich »Diskmaster« angenehm von der Konkurrenz ab.

»Diskmaster« bietet Ihnen eine übersichtlich gestaltete Oberfläche (Bild 6). Während in der Mitte alle wichtigen Befehle für Diskettenoperationen anklickbar bereitstehen, werden links und rechts davon Quell- bzw. Ziel-Verzeichnis angezeigt. Neben Standard-CLI-Befehlen finden Sie auch einige spezielle Befehle. So etwa können Sie mit »Showpic« beliebige IFF-Bilder anzeigen lassen. Mit »Read« können Text-Files gelesen werden, wobei der Text angenehm weich über den Bildschirm rollt und Sie auch vor- und zurückblättern können. »Print« bringt den Text zu Papier. Mit dem Spezial-Befehl

Möglichkeit nicht nur einzelne Files, sondern auf Wunsch auch vollständige Disketten zu kopieren. Eine Anzeige am oberen Bildschirmrand informiert Sie über den momentan verfügbaren Speicherplatz auf den angesprochenen Laufwerken und zusätzlich über den freien Arbeitsspeicher.

Die Druckmaschine

Interessant an »Diskmaster« sind auch die vielfältigen Einstellungsmöglichkeiten. Abgesehen davon, daß es die deutsche PAL-Auflösung unterstützt, kann etwa die Bildschirmauflösung verändert werden. Genauso können Sie die Bild-

Bild 9. Ein Probeausdruck, der die fantastischen Fähigkeiten von »TurboPrint II« zeigt



Dieses Bild wurde auf einen Epson LQ-2550 in rund 75 Sekunden gedruckt

drucken. Die Möglichkeiten und Ergebnisse lassen aber sehr zu wünschen übrig. Mit einem entsprechenden Hilfsprogramm wird der Amiga zur flexiblen Druckmaschine.

Auch hier hat sich ein Programm etablieren können: »TurboPrint II« von IrseeSoft. Dabei ist »TurboPrint II« wesentlich mehr als ein einfaches Druck-Utility. Zum Preis von 99 Mark holt es das Letzte aus Ihrem Drucker heraus. Ganz gleich, ob Sie einen Farbdrucker oder einen einfachen Matrixdrucker besitzen: Die Ergebnisse werden Sie überzeugen (siehe auch Testausdruck Bild 9).

Die meisten Druckprogramme beschränken sich auf den Ausdruck von IFF-Grafiken, die vorher in den Speicher geladen



TOP-COMPUTER AUF DEM PRÜFSTAND

Das neue Magazin für Computer und Musik

Im neuen MIDI Special finden Sie alle Informationen zum Thema "Musik & Computer": Wie steige ich in den MIDI-Bereich ein? - Vier Computertypen und Ihre Voraussetzungen: Apple Macintosh, Atari ST, Commodore Amiga und IBM-kompatible PCs.

Außerdem im MIDI Special: Die dazugehörige Software (Sequencer-Software) mit großem Praxisteil sowie Tips&Tricks. ■ Synthesizer zwischen 400 und 4000 DM im Miditest. ■ Eine umfassende Marktübersicht über Synthesizer. ■ Informationen für Einsteiger, Fortgeschrittene und Profis durch aktuelle Nachrichten und Buchvorstellungen. ■ Sagen Sie uns wie Ihnen das neue MIDI-Special gefällt und gewinnen Sie einen Synthesizer!

Das ST Magazin Sonderheft "MIDI" gibt es bei Ihrem Zeitschriftenhändler!

TOP-COMPUTER AUF DEM PRÜFSTAND

SOFTWARE: GRUNDLAGEN: MIDI-PROGRAMME, AKTUELLE SEQUENZER FÜR AMIGA - ATARI - MAC - PC, DAS MIDI-STUDIO IM COMPUTER. HARDWARE: BRANDNEU: YAMAHA SY 77, EINSTIEG UNTER 400 MARK: DER CASIO MT 540, 60 SYNTHIES IM ÜBERBLICK. PRAXIS: ERSTE SCHRITTE ZUM SONG, MIDI-TECHNIK, LEICHT VERSTÄNDLICH, DIE BESTEN TIPS & TRICKS.

TOP-COMPUTER AUF DEM PRÜFSTAND

SOFTWARE: GRUNDLAGEN: MIDI-PROGRAMME, AKTUELLE SEQUENZER FÜR AMIGA - ATARI - MAC - PC, DAS MIDI-STUDIO IM COMPUTER. HARDWARE: BRANDNEU: YAMAHA SY 77, EINSTIEG UNTER 400 MARK: DER CASIO MT 540, 60 SYNTHIES IM ÜBERBLICK. PRAXIS: ERSTE SCHRITTE ZUM SONG, MIDI-TECHNIK, LEICHT VERSTÄNDLICH, DIE BESTEN TIPS & TRICKS.

TOP-COMPUTER AUF DEM PRÜFSTAND

SOFTWARE: GRUNDLAGEN: MIDI-PROGRAMME, AKTUELLE SEQUENZER FÜR AMIGA - ATARI - MAC - PC, DAS MIDI-STUDIO IM COMPUTER. HARDWARE: BRANDNEU: YAMAHA SY 77, EINSTIEG UNTER 400 MARK: DER CASIO MT 540, 60 SYNTHIES IM ÜBERBLICK. PRAXIS: ERSTE SCHRITTE ZUM SONG, MIDI-TECHNIK, LEICHT VERSTÄNDLICH, DIE BESTEN TIPS & TRICKS.

Grafik: DIGIPAIN 3



Anwendungsgebiete: Malprogramm für den HAM-Modus (4096 Farben), geeignet zum Nachbearbeiten von digitalisierten Bildern

Kurzbeschreibung: Neben herkömmlichen Werkzeugen zum Malen – Freihand, Linien, Kreise etc. – bietet Digipaint 3 eine Menge weiterer Funktionen, die die Fähigkeiten des HAM-Modus ausnutzen.

So lassen sich feine Schattierungen ebenso erstellen, wie Farbflächen nach Anwendungsgebiete der Lasurmalerei auf den Bildschirm »gehaucht« werden können. Eine UNDO-Funktion macht Pannen rückgängig, mit »AGAIN« lassen sich Operationen erneut anwenden.

Mittels den Textur-Funktionen können digitalisierte oder gemalte Bilder auf verschiedenste Formen aufgespannt werden. Spannt man etwa ein rechtwinkeliges Gitter auf eine Kugel, bilden sich Längen- und Breitengrade in perspektivisch richtiger Darstellung.

Auch Texte lassen sich in Bilder einbauen. Dazu können Standardschriften verwendet werden. Die Buchstaben können mit einer Krümmungsfunktion auch verzerrt werden und lassen sich auf unregelmäßige Flächen spannen.

Besonderheiten: Bildschirmgröße bis 1024 x 1024 Punkte, Konvertierprogramm wird mitgeliefert

++ Positiv: leichte Bedienbarkeit durch gut strukturierten Menüaufbau; Lernhilfen im Handbuch; Bildschirmgröße stufenlos einstellbar

– Negativ: wenig Werkzeuge zur Auswahl

Fazit: Digipaint 3 ist ein schnelles Malprogramm für den HAM-Modus des Amiga mit vielseitigen Hilfsroutinen.

Preis: rund 200 Mark

Hersteller: NewTek

Anbieter: Casablanca, Nehringskamp 9, 4630 Bochum, Tel. 0234/41 19 94

Test in: AMIGA-Magazin 10/89, S. 172

Testnote: 10,2

Grafik: DELUXE PAINT III



Anwendungsgebiete: Komplexes Malprogramm mit Animationsmodul

Kurzbeschreibung: DPaint III unterstützt alle Bildschirmauflösungen des Amiga, einschließlich PAL und Overscan. Bis auf den HAM-Modus (4096 Farben) stehen auch alle Farbmodi zur Verfügung.

Zum Malen steht ein vielfältiges Repertoire an Werkzeugen zur Verfügung, darunter Standardfunktionen wie das Zeichnen von Kreisen, Linien, Polygonen etc. Interessant ist neben der Verwendung von verschiedenen Pinselstärken die Definition von eigenen Brushes (Pinseln): Diese können Ausschnitte aus einem beliebigen Bild sein und produzieren faszinierende Effekte.

Mit »Smear« lassen sich Farben verwischen, »Smooth« läßt die Konturen eines Bildes weicher werden. Mit einer Perspektive-Funktion können räumliche Eindrücke erzeugt werden. Daneben stehen noch viele weitere Instrumente zur Bildbearbeitung zur Verfügung.

Der eingebaute Animationsteil erlaubt es, auf einfache Weise Bewegungsabläufe zu erstellen.

Besonderheiten: Möglichkeit zur Erstellung von Animationen

++ Positiv: vielfältige Funktionen, ausgezeichnetes deutsches Handbuch

– Negativ: keine Unterstützung des HAM-Modus

Fazit: DPaint III ist das Malprogramm für den Amiga. Nach der Tradition seiner Vorgänger bietet es Grafikwerkzeuge vom Feinsten und besticht nun auch durch einen eingebauten Animations-Teil.

Preis: 249 Mark

Hersteller: Electronic Arts

Anbieter: Markt & Technik, Hans-Pinsel-Str. 2, 8013 Haar, Tel. 089/4613-0

Test in: AMIGA-Magazin 6/89, S. 150

Testnote: 10,7

Grafik: REFLECTIONS



Anwendungsgebiete: Raytracing-Programm zur Erzeugung realistischer, dreidimensionaler Bilder

Kurzbeschreibung: Mit »Reflections« können auf Basis von Drahtgittermodellen dreidimensionale Bilder berechnet werden. Raytracing heißt auf deutsch Strahlenverfolgung und bedeutet, daß um das Objekt Lichtquellen gesetzt werden können. Die Reflexion oder Brechung der Lichtstrahlen an den Objekten wird bei der Berechnung berücksichtigt.

Die Objekte können verschiedene Oberflächenstrukturen besitzen, etwa Holz, durchsichtiges oder spiegelndes Glas. Mit der Funktion Textur können auch Bilder um Objekte – etwa Kugeln – gelegt werden.

»Reflections« besteht aus mehreren Modulen. Mit dem ersten, »Construct«, werden Objekte und Szenen erstellt. Diese werden mit dem Modul »Beams« mit rund 16,8 Millionen Farben berechnet. Da diese mit dem Amiga nicht dargestellt werden können, wird ein zusätzliches Modul, »Show«, zur Darstellung benötigt.

Mit der Textur-Funktion können Bilder um Säulen- oder Kugelformen gelegt werden.

Besonderheiten: Berechnet professionelle Bilder mit bis zu 16,8 Millionen Farben.

++ Positiv: umfangreiches deutsches Handbuch, niedriger Preis

– Negativ: keine Unterstützung von Co-processoren

Fazit: Reflections ist ein Ray-Tracing-Programm mit einem konkurrenzlosen Preis-Leistungs-Verhältnis

Preis: rund 100 Mark

Hersteller: Markt & Technik

Anbieter: Markt & Technik, Hans-Pinsel-Str. 2, 8013 Haar, Tel. 089/4613-0

Test in: AMIGA-Magazin 7/89, S. 140

Testnote: 9,7

Grafik: BUTCHER 2.0



Anwendungsgebiete: dient zur Manipulation von IFF-Bildern

Kurzbeschreibung: Mit Butcher lassen sich Bilder auf vielfältige Weise verändern. Mit der Mosaik-Funktion werden Grafiken in beliebige Raster zerlegt. Ein Negativ eines Bildes kann ebenso erzeugt werden, wie jede Farbe durch ihr Komplementär ersetzt werden kann. Weitere Möglichkeiten sind die Erzeugung von Falsch- oder Pseudofarben.

Mit »Farbtrennen« können Bilder in die Farben Schwarz, Gelb, Magenta und Cyan gesplittet werden. »Antik« gibt Ihren Bildern den Eindruck vergilbter Fotos. Ebenso können aus Farbbildern Schwarzweiß-Grafiken erzeugt werden. »Konsolidierung« erlaubt die Reduktion der Farben eines Bildes bei fast gleichbleibender Qualität.

Mit den integrierten Zeichenfunktionen können die Bilder manuell bearbeitet werden. Mit einer weiteren Funktion wird der verfügbare Arbeitsspeicher angezeigt. Bei Speichermangel wird auf Wunsch nur ein Teil des Programms geladen. Der Rest wird dann bei Bedarf nachgeladen.

Besonderheiten: deutsche Benutzerführung, PAL-Auflösung

++ Positiv: sehr leistungsfähige Befehlsvielfalt, Zeichenfunktion

- Negativ: wenig Möglichkeiten zum Nachbearbeiten von HAM-Bildern

Fazit: Butcher 2.0 unterstützt durch seine Funktions-Vielfalt den Anwender bei der Gestaltung und Bearbeitung seiner Bilder. Es sollte in keiner grafikorientierten Software-Sammlung fehlen.

Preis: 80 Mark

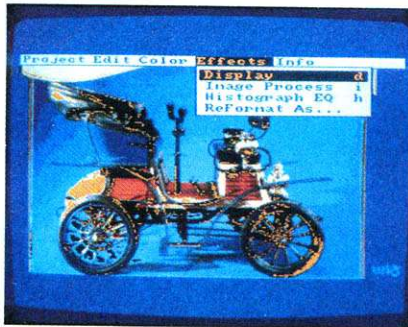
Hersteller: Eagle Tree Software

Anbieter: Jumbosoft, Horemansstraße 2, 8000 München 19, Tel. 089/1 23 40 65

Test in: AMIGA-Magazin 2/88, S. 113

Testnote: 10,8

Grafik: PIXMATE



Anwendungsgebiete: dient zur Manipulation von IFF-Bildern

Kurzbeschreibung: Pixmate verfügt über einen Image-Prozessor, mit dem Bilder mit einem Weichzeichner versehen oder schärfer gemacht werden können. Außerdem lassen sich damit Bilder automatisch umranden oder zwei Bilder miteinander logisch (AND, OR, EOR) miteinander verknüpfen.

Weitere Funktionen sind das Spiegeln, Vergrößern oder Verkleinern von Grafiken. Mit »Color Bias« kann ein Bild mit einem beliebigen Farbtönen überlagert werden. Weiterhin können HAM-Bilder in das Extra-Halfbrite- oder das normale 32-Farben-Format konvertiert werden. Die Histogramm-Funktion gibt Aufschluß, wie oft bestimmte Farben in einem Bild vorkommen.

Eine interessante Möglichkeit ist, Bilder aus gleichzeitig laufenden anderen Programmen direkt zu übernehmen.

Im Info-Menü erhalten Sie Auskunft über den freien Speicherplatz sowie das Format des gerade bearbeiteten Bildes. Es können gleichzeitig zwei Bilder im Speicher stehen.

Besonderheiten: auch HAM-Bilder können bearbeitet werden

++ Positiv: schnelle Konvertierung zwischen verschiedenen Grafikmodi

- Negativ: Probleme mit der Funktion »Color Bias« in der PAL-Auflösung

Fazit: Pixmate ist bei der Bearbeitung von Grafiken ein sehr nützliches Programm. Einige besondere Fähigkeiten heben es von der Konkurrenz ab.

Preis: 110 Mark

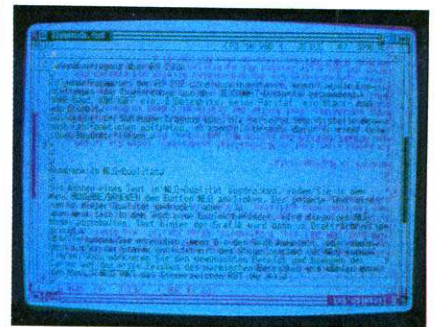
Hersteller: Progressive Peripherals & Software

Anbieter: Jumbosoft, Horemansstraße 2, 8000 München 19, Tel. 089/1 23 40 65

Test in: AMIGA-Magazin 5/88, S. 128

Testnote: 10,4

Textverarbeitung: BECKERTEXT



Anwendungsgebiete: Textverarbeitung mit WYSIWYG und Grafikeinbindung

Kurzbeschreibung: »Beckertext« arbeitet nach dem WYSIWYG-Prinzip, man sieht den Text am Bildschirm also bereits in der Form, wie er später auf dem Drucker erscheint.

In Beckertext können – wenn auch eingeschränkt – Grafiken eingebunden werden. Eine Trennfunktion ermöglicht automatische Silbentrennung am Zeilenende. Das eingebaute Lexikon kontrolliert den Text auf etwaige Rechtschreibfehler.

Weitere Funktionen ermöglichen das automatische Erstellen eines Inhalts- oder Stichwortverzeichnisses. Die Funktionstasten können sowohl mit Befehlen als auch Textbausteinen – etwa für Formbriefe – belegt werden.

Der Text kann linksbündig, zentriert, mit Randausgleich oder auch rechtsbündig formatiert werden.

Eine nützliche Funktion ist das automatische Erstellen einer Sicherheitskopie von abgespeicherten Texten.

Beckertext verwendet eigene Druckertreiber, die einfach editiert werden können.

Besonderheiten: Rechenoperationen im Text möglich

++ Positiv: deutsche Benutzerführung, umfangreiches, gutes Handbuch

- Negativ: umständlicher File-Requester, keine Fußnotenverwaltung

Fazit: Beckertext ist eine schnelle, einfach zu bedienende Textverarbeitung, die universell einsetzbar ist.

Preis: etwa 200 Mark

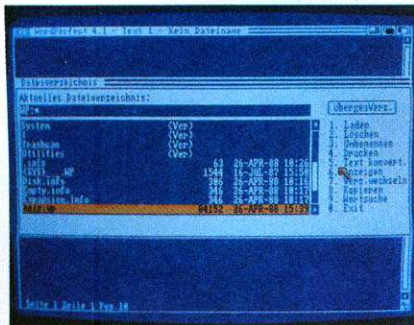
Hersteller: Data Becker

Anbieter: Data Becker, Merowingerstr. 30, 4000 Düsseldorf 1, Tel. 02 11/33 00 10

Test in: AMIGA-Magazin 3/89, S. 28

Testnote: 8,5

Textverarbeitung: WORDPERFECT 4.1



Anwendungsgebiete: Textverarbeitung für den Profi-Bereich

Kurzbeschreibung: Wordperfect bietet ein nahezu unüberschaubares Ausmaß an Funktionen, das hier nur angerissen werden kann. So verfügt es über ein eingebautes Lexikon mit über 400 000 Einträgen. Dieses wird durch einen Thesaurus ergänzt, also ein Wörterverzeichnis, das Synonyme und Antonyme eines Begriffs aufführt und Wortwiederholungen vermeiden hilft.

Neben der Möglichkeit zur Erstellung von Fuß- und Kopf-Zeilen besitzt Wordperfect eine Fußnotenverwaltung. Diese sorgt automatisch für die seitenrichtige Platzierung von Kommentaren. Wordperfect unterstützt auch das Verfassen von Inhalts- und Sachregistern.

Auf Wunsch wird der gerade bearbeitete Text in einem einstellbaren Zeitintervall automatisch gesichert. Das Überschreiben vorhandener Files ist nur nach Beantworten einer Sicherheitsabfrage möglich.

Des weiteren verfügt Wordperfect über alle Standardfunktionen einer Textverarbeitung und eine Vielzahl weiterer Möglichkeiten.

Besonderheiten: umfangreiches Lexikon/Thesaurus

++ Positiv: nahezu unbegrenzte Funktionsvielfalt, professionell einsetzbar

- Negativ: kein WYSIWYG, keine Grafikeinbindung

Fazit: Wordperfect ist mit Abstand der Profi unter den Textverarbeitungen für den Amiga. Es eignet sich hervorragend zum Schreiben von Büchern oder wissenschaftlichen Arbeiten, bedarf allerdings einiger Einarbeitungszeit.

Preis: etwa 800 Mark (für Studenten die Hälfte)

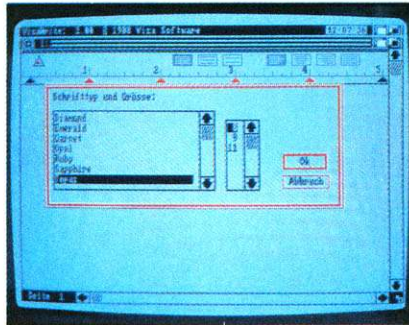
Hersteller: Wordperfect

Anbieter: Wordperfect Software GmbH, Frankfurter Str. 33-39, 6236 Eschborn, Tel. 06196/904-0

Test in: AMIGA-Magazin 3/89, S. 28

Testnote: 8,3

Textverarbeitung: VIZAWRITE 2.0 DESKTOP



Anwendungsgebiete: grafikorientierte Textverarbeitung

Kurzbeschreibung: Vizawrite arbeitet nicht mit dem Standard-Zeichensatz des Amiga, sondern mit Grafikzeichensätzen unterschiedlicher Breite und Höhe. Diese erscheinen nicht nur auf dem Bildschirm, sondern auch auf dem Drucker. Einige High-Quality-Fonts bringen besonders gute Druckergebnisse.

Vizawrite arbeitet nach dem WYSIWYG-Konzept. Der Text erscheint also am Drucker genauso wie am Bildschirm. Neben den Standard-Textverarbeitungs-Funktionen gibt es unter anderem die Möglichkeit, mehrere Texte gleichzeitig zu verarbeiten.

Vizawrite erlaubt auch, Text mit Grafik zu kombinieren. Dem Anwender stehen dazu vielfältige Funktionen zur Verfügung.

Vizawrite unterstützt die Arbeit mit Formbriefen durch die Fähigkeit, Textbausteine zu verarbeiten. Auch eine flexible Serienbrief-Funktion steht zur Verfügung.

Besonderheiten: Textausdruck erfolgt im Grafikmodus mit High-Quality-Zeichensätzen

++ Positiv: 20 verschiedene Zeichensätze werden mitgeliefert, einfache Bedienung

- Negativ: keine Trennautomatik, kein Lexikon

Fazit: Eine Textverarbeitung für alle, die mit verschiedenen Zeichensätzen und -größen arbeiten wollen und zudem auf einfache Bedienung Wert legen.

Preis: 200 Mark

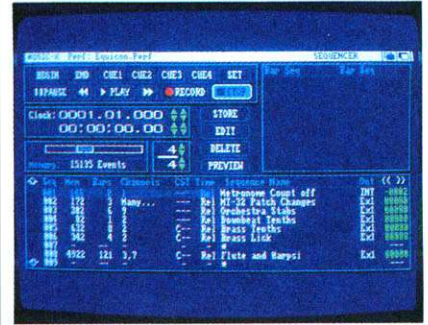
Hersteller: Viza Software

Anbieter: DTM, Poststr. 25, 6200 Wiesbaden-Bierstadt, Tel. 06121/560084

Test in: AMIGA-Magazin 3/89, S. 28

Testnote: 7,7

Musik: MUSIC X



Anwendungsgebiete: Komplexes Musik-/MIDI-Paket

Kurzbeschreibung: Music X ist ein Programm für jene, die den Amiga über MIDI mit einem Synthesizer etc. verbinden wollen. Es vereint die Funktionen verschiedener Programme auf einer Oberfläche.

In erster Linie ist Music X ein MIDI-Sequencer, dient also zum Aufzeichnen und Editieren von Musikstücken, die per Synthesizer eingespielt werden. Dabei können auch die Amiga-Sounds verwendet werden.

Weiterhin lassen sich mit Music X die Klänge von angeschlossenen MIDI-Geräten editieren, sofern das Gerät einem bestimmten Typ entspricht. Genauso können die Klänge von diesen auf Amiga-Disketten gespeichert und verwaltet werden. Dies geschieht mit dem sogenannten Librarian. Wessen Gerät bei diesen Funktionen nicht unterstützt wird, der kann selbst ein Übertragungsprotokoll mit dem »Protocol-Editor« erstellen.

Der eingebaute MIDI-Filter sorgt besonders bei mehreren angeschlossenen Geräten für korrekten Datenaustausch und erlaubt gezielte Manipulationen.

Besonderheiten: vereint die Funktionen mehrerer Programme in einem

++ Positiv: enorme Funktionsvielfalt, sehr gutes (englisches) Handbuch

- Negativ: fehlende Quantisierung bei der Aufnahme (Quantisierung nur im Event-Editor möglich)

Fazit: Music X ist ein vielseitiges Programm, das ermöglicht, schnell und einfach auf allen MIDI-Anwendungsbe-reichen zu arbeiten

Preis: etwa 450 Mark

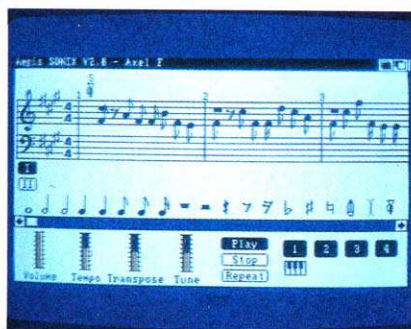
Hersteller: Micro Illusions

Anbieter: Soyka, Hattinger Str. 685, 4630 Bochum 5, Tel. 0234/49825-26

Test in: AMIGA-Magazin 10/89, S. 176

Testnote: 10,0

Musik: SONIX 2.0



Anwendungsgebiete: Kompositionsprogramm, Synthesizer

Kurzbeschreibung: Sonix eignet sich zum Komponieren von Musikstücken. Die Stücke werden per Maus in klassischer Notation eingegeben. Das Setzen der Noten ist sehr einfach zu handhaben. Es können bis zu acht Stimmen (vier interne und vier MIDI-Stimmen) verwendet werden. Die Noten werden auf dem Bildschirm dargestellt und können auch auf dem Drucker ausgegeben werden.

Der Synthesizerteil arbeitet nach dem Prinzip eines analogen Synthesizers und wird per Maus bedient. Mit den verschiedenen Reglern und Schaltern können Lautstärkeverlauf, Grundwellenform des Klangs usw. bestimmt werden. Viele Funktionen lassen sich mit einem Oszillator modulieren. Die Wellenform des Klangs kann mit der Maus gezeichnet werden. Die Möglichkeiten des Synthesizers lassen sich durchaus mit denen der großen Vorbilder messen. Für die Kompositionen können auch digitalisierte Klänge verwendet werden. MIDI-Funktionen ermöglichen den Anschluß eines Synthesizers.

Besonderheiten: MIDI-Funktionen, komplexer Synthesizer

++ Positiv: einfach zu bedienen, ausführliches Handbuch, niedriger Preis

- Negativ: für klassische Stücke weniger geeignet

Fazit: Sonix ist das Musikprogramm für den Amiga. Es besticht durch seine einfache Anwendung und Funktionsvielfalt. Es ist auch jenen zu empfehlen, die sich für Musik mit dem Amiga nur am Rande interessieren.

Preis: etwa 90 Mark

Hersteller: Aegis

Anbieter: Atlantis, Dunantstr. 53, 5030 Hürth, Tel. 02233/41081

Test in: AMIGA-Magazin 10/87

Testnote: 8,7 (Version 1.6)

Musik: AUDIOMASTER II



Anwendungsgebiete: Programm zum Digitalisieren und Nachbearbeiten von Sounds

Kurzbeschreibung: Mit einem angeschlossenen Sound-Sampler lassen sich mit Audiomaster II beliebige Klänge digitalisieren. Auf Wunsch kann in Stereo gesampelt werden. Es können auch bereits digitalisierte Klänge geladen und mit Audiomaster II bearbeitet werden.

In zwei Editierfenstern können die konservierten Klänge nachbearbeitet werden. Sie können dort geschnitten und mit verschiedenen Effekten versehen werden. So machen Echo- und Hall-Funktionen den Klang voller. »Fade in« und »Fade out« ermöglicht weiches Ein- Ausblenden der Töne. Eine Loop-Funktion ermöglicht das Wiederholen von Teilen des Klangs.

Die Anwendung der Funktionen auf den Klang ist einfach. Die im Editierfenster dargestellte Wellenform wird dazu einfach in den gewünschten Bereichen per Maus markiert. Zoom-Funktionen ermöglichen exaktes Bearbeiten der Klänge.

Zur Weiterverwendung können die Klänge als Instrumente im Sonix- oder IFF-Format gespeichert werden.

Besonderheiten: verarbeitet Stereo-Klänge

++ Positiv: vielfältige Nachbearbeitungsmöglichkeiten von Klängen

- Negativ: nicht mit allen Digitizer-Typen kann digitalisiert werden

Fazit: Audiomaster II ist eines der komplexesten Programme zum Nachbearbeiten von digitalisierten Klängen.

Preis: etwa 170 Mark

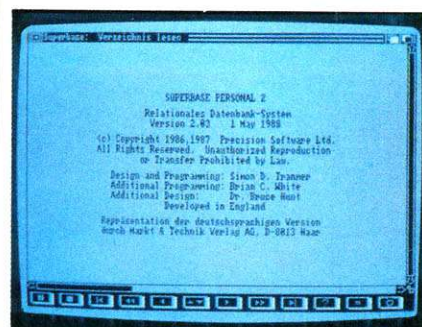
Hersteller: Aegis

Anbieter: Atlantis, Dunantstr. 53, 5030 Hürth, Tel. 02233/41081

Test in: AMIGA-Magazin 6/89, S. 162

Testnote: 10,2

Datenbank: SUPERBASE PERSONAL II



Anwendungsgebiete: relationale Datenbank für private und gewerbliche Anwendung

Kurzbeschreibung: Superbase Personal II läßt sich einfach an individuelle Anwendungen anpassen. Die Definition der Dateien erfolgt komfortabel mit Dialogrequestern. Für einzelne Daten können Indexdateien angelegt werden, die Informationen über die Sortierfolge der Daten enthalten.

Erfasste Datensätze können auf vielfältige Weise angezeigt, ausgewertet und verknüpft werden. Dies kann mit arithmetischen Operationen genauso wie mit Schlüsselwörtern (AND, OR, etc.) geschehen. Mit einer Schaltleiste, ähnlich wie bei Kassettenscannern, können Sie durch den Datenstand blättern. Eine Index-Suchfunktion erleichtert das Auffinden von Daten.

Ein weiterer Modus bietet die Möglichkeit, Daten in Tabellenform darzustellen. Alle Ausgaben können auch auf einem angeschlossenen Drucker erfolgen.

Ein einfacher Texteditor ergänzt die Funktionen von Superbase II Personal. In diesem Zusammenhang ist es auch möglich, Rundschreiben gleichen Inhalts an verschiedene Empfänger auszugeben.

Besonderheiten: Verwaltung auch von Sound- und Grafikdaten

++ Positiv: relationale Datenstruktur, einfache Bedienung

- Negativ: nicht programmierbar

Fazit: Eine leistungsfähige und vor allem bedienungsfreundliche Datenbank für private und einfache gewerbliche Anwendungen.

Preis: etwa 250 Mark

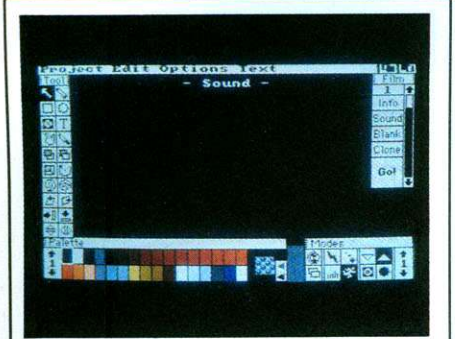
Hersteller: Precision Software

Anbieter: Markt & Technik AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar, Tel. 089/4613-0

Test in: AMIGA-Magazin 7/88, S. 110

Testnote: 10,0

Animation:
FANTAVISION



Anwendungsgebiete: dient zur Erstellung einfacher Trickfilme

Kurzbeschreibung: Fantavision ist ein Animationsprogramm, mit dem auf einfache Weise in erster Linie zweidimensionale Animationen erzeugt werden können.

Animationsobjekte bestehen bei Fantavision normalerweise aus 64 Punkten, deren Lage und Position per Maus bestimmt wird. Die Objekte sind zweidimensional, können jedoch im Raum gedreht werden. Dadurch lassen sich auch dreidimensionale Effekte erzeugen.

Interessant ist, daß Fantavision Zwischenbilder bei Animationen selbst berechnet. Um etwa einem Vogel den Flügel Schlag beizubringen, brauchen Sie dem Programm nur die beiden Flügelstellungen zeigen (Flügel oben, Flügel unten), die dazwischenliegenden Bilder werden berechnet: schon fliegt der Vogel. Soll ein Objekt von einem Punkt zum anderen bewegt werden, muß es nur von der Start- in die Zielposition bewegt werden. Die dazwischenliegende Bewegung wird automatisch berechnet.

Nicht nur Objekte, auch IFF-Grafiken (Bitmaps) können bewegt, aber nicht animiert werden. Als Hintergrund für Animationen lassen sich beliebige Bilder - auch HAM - verwenden.

Besonderheiten: Synchron zu den Animationen können auch digitalisierte Klangkulissen gespielt werden

++ Positiv: fast alle Grafikmodi, PAL-Bildschirm, gutes Handbuch, einfache Bedienung

- **Negativ:** starke Einschränkungen für Bitmaps und Textdarstellung

Fazit: Fantavision ist ein gut durchdachtes Animationsprogramm, das durch Bedienungskomfort glänzt.

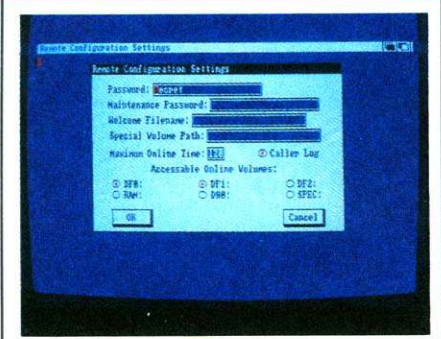
Preis: 150 Mark

Hersteller: Broderbund

Anbieter: HS&Y, Herderstr. 94, 5000
Köln 41, Tel. 0221/431687

Test in: AMIGA-Magazin 1/89, S. 176

Testnote: 8,7

DFÜ:
DIGA

Anwendungsgebiete: Programm zur Datenfernübertragung mit verschiedenen Terminal-Emulationen

Kurzbeschreibung: Mit Diga lassen sich Datenübertragungsraten bis 31 250 Baud (Bit/s) erreichen. Es unterstützt sechs verschiedene Übertragungsmodi, darunter XModem, YModem oder normale ASCII-Übertragung.

Ein spezieller Modus ist das Double-talk-Protokoll. Er ermöglicht das gleichzeitige Übertragen von Daten in beide Richtungen. Während der Datenübertragung können die Benutzer an beiden Enden der Leitung auch noch miteinander kommunizieren.

Diga verwendet ausgeklügelte Prüfsummen, was auch bei schlechten (Telefon-)Leitungen eine sichere Datenübertragung gewährleistet.

Unter den Terminals haben sich verschiedene Standards etabliert. Diga unterstützt neben VT52- und VT100-Terminals und dem ANSI-Farb-Terminal und einem Modus, bei dem alle eingehenden Daten akustisch ausgesprochen werden, die Emulation eines Tektronix 4010 Grafik-Terminals. Der Benutzer kann also mit den verschiedensten Partner-(Geräten) kommunizieren. Eine eingebaute kleine Mailbox rundet den Funktionsumfang ab.

Besonderheiten: emuliert viele Terminals, darunter Tektronix 4010 Grafik-Terminal.

++ Positiv: Doubletalk-Modus, umfassendes Handbuch

- **Negativ:** keine deutschen Umlaute

Fazit: Diga ist das Referenzprogramm für Datenfernübertragung auf dem Amiga. Die durchdachten vielfältigen Funktionen sind sowohl für den Einsteiger als auch den Profi von Nutzen.

Preis: etwa 130 Mark

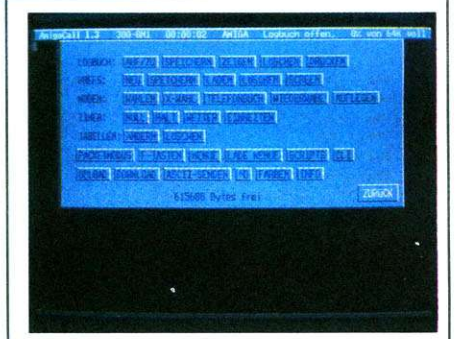
Hersteller: Aegis

Anbieter: Jumbosoft, Horemansstr. 2,
8000 München 19, Tel. 089/1234065

Test in: AMIGA-Magazin 10/87, S. 101

Testnote: 11,0

**DFÜ:
AMIGACALL**



Anwendungsgebiete: DFÜ-Programm, das die Bedienung von Mailboxen vereinfacht

Kurzbeschreibung: Die Übertragungsraten von AmigaCall reichen von 300 bis 19200 Baud. Als Emulation steht ein VT100 zur Verfügung.

Die meisten Mailboxen sind menügesteuert, was mit lästiger Tipparbeit verbunden ist. AmigaCall löst das Problem mit »Textklicking«. Durch Anklicken von selbstdefinierten Menüpunkten per Maus können Zeichen oder ganze Wörter gesendet werden. Da man die Menüs selbst definieren kann, ist eine Anpassung an jede Box möglich.

Ein Gebührenzähler informiert den Anwender über die während der Verbindung entstehenden Telefonkosten. Das eingebaute Telefonbuch wählt automatisch den gewünschten Teilnehmer an.

Die Programmiersprache »Script« erlaubt es, Mailbox-Sitzungen weitgehend zu automatisieren.

Besonderheiten: Definition eigener Menüs zur einfacheren Bedienung von Mailboxen

++ Positiv: großer Funktionsumfang, niedriger Preis, umfangreiches Handbuch

- **Negativ:** es werden keine Pull-Down-Menüs (sondern ein Pop-Up-Menü) verwendet, was gewöhnungsbedürftig ist

Fazit: Wenn AmigaCall auch nicht so viele Funktionen aufweist wie Diga, ist es doch eine interessante Alternative, vor allem bei dem günstigen Preis.

Preis: etwa 90 Mark

Hersteller: Atlantis

Anbieter: Markt & Technik, Hans-Pinsel-Str. 2, 8013 Haar, Tel. 089/4613-0

Test in: AMIGA-Magazin 11/88, S. 164

Testnote: 9,2

Utility: DISKMASTER



Anwendungsgebiete: Benutzeroberfläche für das CLI

Kurzbeschreibung: Diskmaster ist ein Utility, das Dateioperationen wesentlich vereinfacht. Die wichtigsten DOS-Befehle befinden sich auf einer Schalterleiste in der Mitte des Bildschirms und brauchen zur Ausführung nur angeklickt werden. Neben herkömmlichen Befehlen gibt es auch einige Spezial-Befehle: »Read« erlaubt das komfortable Lesen von Texten, »Print« gibt Textdateien auf dem Drucker aus und »ShowPic« dient zur Anzeige von IFF-Bildern. Das Kommando »Move« erlaubt das Verschieben von Dateien. Befindet sich z.B. eine Datei in einem falschen Verzeichnis, braucht sie nicht erst in das richtige kopiert und im alten gelöscht, sondern nur »moved« werden.

Links und rechts von der Gadget-Leiste wird der Inhalt von Quell- und Zielverzeichnis angezeigt. Mit verschiedenen Optionen wie das Verändern der Auflösung oder der Zeichensatzgröße läßt sich das Programm an eigene Bedürfnisse anpassen.

Besonderheiten: Diskmaster beherrscht das Packen und Entpacken von Dateien

++ Positiv: bis zu zwölf Laufwerke werden unterstützt, übersichtliche Bildschirmdarstellung, PAL-Auflösung

- Negativ: Umschalten zwischen Quell- und Zielverzeichnis anfangs gewöhnungsbedürftig

Fazit: Diskmaster ist ein ausgereiftes Disketten-Utility zum täglichen Gebrauch für jeden Amiga-Benutzer.

Preis: 120 Mark

Hersteller: PP&S

Anbieter: GTI, Zimmermühlenweg 73, 6370 Oberursel, Tel. 061 71/73048

Test in: AMIGA-Magazin 9/88, S. 148

Testnote: 9,7

Utility: PROJECT D



Anwendungsgebiete: Kopierprogramm (auch für Fremdformate)

Kurzbeschreibung: Project D ist ein Paket, das aus drei Modulen besteht.

- »BackupTool« ist ein schnelles Kopierprogramm zum Erstellen von Diskettenkopien. Dabei läßt sich die Anzahl der zu kopierenden Spuren festlegen, auch ob ein- oder zweiseitig kopiert werden soll. Mit drei Laufwerken können gleichzeitig zwei Kopien erstellt werden.

- »OmniTool« erlaubt das Kopieren von Fremdformaten. Folgende Formate stehen zur Auswahl: Atari ST, CP/M, MS-DOS und Xenix. Es lassen sich auch Disketten im Ensoniq-Mirage-Format kopieren. Auch im OmniTool kann zwischen einseitiger und doppelseitiger Kopie gewählt werden.

- Das »EditorTool« ist ein einfacher Diskettenmonitor zur Manipulation von Disketten. Die Anzeige der Daten erfolgt sowohl im Hex- als auch im ASCII-Format.

Besonderheiten: kopiert auch Fremdformate, eingebauter Diskettenmonitor

++ Positiv: kopiert Disketten auch einseitig, wählbares Verify, umfangreiche Parameterliste, arbeitet mit ein, zwei oder drei Disketten-Laufwerken

- Negativ: Handbuch in Englisch

Fazit: Project D gehört zu den besten Kopierprogrammen für den Amiga. Besonders die Möglichkeit, verschiedene Fremdformate zu kopieren, ist einzigartig.

Preis: etwa 80 Mark

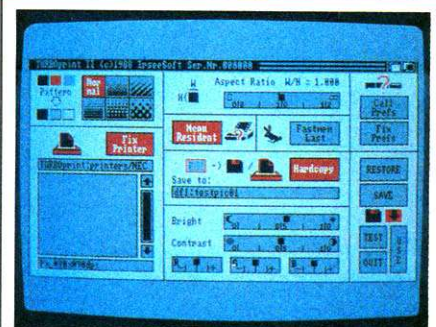
Hersteller: Fuller Computer Systems

Anbieter: Soyka Datentechnik, Hattinger Str. 685, 4630 Bochum, Tel. 0234/49825-26

Test in: AMIGA-Magazin 6/88, S. 136

Testnote: 9,3

Utility: TURBOPRINT II



Anwendungsgebiete: Druck-Utility

Kurzbeschreibung: Turboprint unterstützt den Anwender beim Ausdruck von Bildern. Es läuft im Hintergrund und erlaubt mittels Tastenkombination den Ausdruck oder das Abspeichern beliebiger Bilder – auch Ausschnitte – aus laufenden Programmen heraus.

Turboprint II verbessert und beschleunigt außerdem den Ausdruck aller Programme, welche die Preferences-Treiber zum Drucken verwenden.

Mittels eigener Preferences kann das Programm optimal an verschiedene Drucker angepaßt werden. Hier lassen sich etwa die gewünschte Auflösung im Druck oder Helligkeit und Kontrast des Ausdruckes festlegen. Außerdem können verschiedene Mustertypen für den Druck bestimmt werden. Antialiasing gleicht »Pixeltreppen« aus. Mit weiteren Schaltern läßt sich die Größe des Ausdrucks festlegen.

Turboprint ist resetfest, daher kann es auch zusammen mit Programmen verwendet werden, die sich nur per Reset starten lassen. Auf Wunsch sind die Preferences jederzeit per Tastenkombination abrufbereit.

Besonderheiten: ersetzt die Workbench-Druckertreiber

++ Positiv: schnelle Treiber in verschiedenen Auflösungen, Hardcopy – auf Tastendruck, Bildspeicherfunktion, resetfest

- Negativ: Kopierschutz

Fazit: Turboprint II ist ein leistungsfähiges Werkzeug. Es bietet vielfältige Funktionen und beschleunigt den Ausdruck von Grafiken.

Preis: etwa 100 Mark

Hersteller: Irseesoft

Anbieter: Irseesoft, Grüntenstraße 6, 8951 Irsee, Tel. 08341/74327

Test in: AMIGA-Magazin 1/89, S. 187

Testnote: 9,8

Der gläserne

In Paula steckt Musik –
er ist der Manager für Sound und
Diskettenoperationen

Die CIA 8520-Port-Bausteine sind die Verbind-
ung zur Außenwelt, sie steuern sowohl die parallele
und serielle Schnittstelle als auch die Gameports

Denise ist für
die gesamte Videodar-
stellung verantwort-
lich, einschließlich
der Sprite-Kontrolle
und der Playfield-
Darstellung

Der FatAgnus ist
ein hochintegrierter
Baustein und ist
für die brillante Grafik-
darstellung des
Amiga verantwortlich

Der 68000-Prozessor von Motorola
ist die CPU des
Amiga und damit
sein Herzstück

Der Expansion-
Port macht den Amiga
fast unbeschränkt
ausbaufähig

Das Kickstart-ROM
enthält das Betriebs-
system des Amiga

Der 512-KByte-
RAM-Speicher (auch
Chip-Memory ge-
nannt) ist aus vier
1-MByte-RAM-Chips
aufgebaut

Die Escape-Taste
wird oft zum Umschal-
ten in einen anderen
Modus oder für
das Beenden eines
Programms genutzt

Mit der Tab-Taste
springt man Tabulato-
ren im Text an

Diese Tasten haben nur in Verbind-
ung mit einer anderen Sinn

Die deutsche Stan-
dard-Tastatur

ne Amiga **H**INTERGRUND

Viele analoge und digitale Bauteile sind aus Platzgründen in Gary zusammengefaßt

Das interne 3 1/2-Zoll-Disketten-Laufwerk kann 880 KByte formatieren

Dies ist der Steckplatz für die interne Speichererweiterung

Die Funktionstasten können frei belegt werden

Der 6570-Chip kontrolliert die Tastatur des Amiga

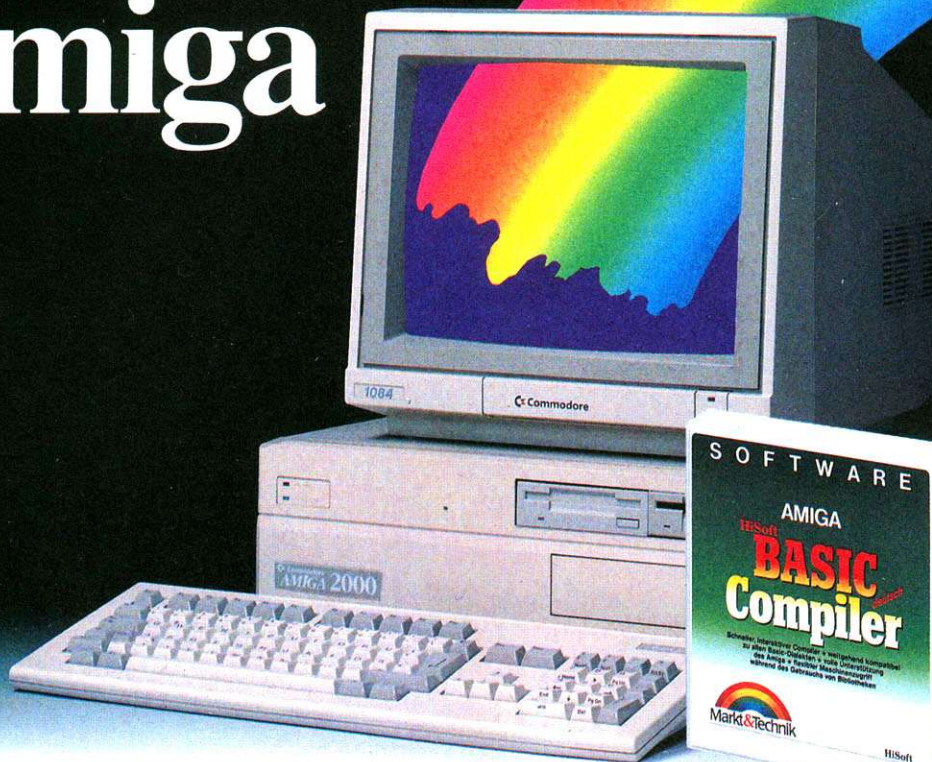
Die beiden LEDs zeigen den Betriebszustand des Amiga an. Rot, der Amiga ist in Betrieb. Grün, das Disketten-Laufwerk arbeitet.

Die Return-Taste schließt Befehlseingaben ab oder löst bei einer Textverarbeitung einen Wagenrücklauf mit Zeilenvorschub aus

Der abgesetzte Zehnerblock steht im Mittelpunkt, wenn es um Zahlen geht

Mit den Cursortasten können Sie den Cursor beliebig positionieren

Stoff für Ihren Amiga



HiSoft-Basic-Compiler

Mit HiSoft Basic gibt es endlich auch einen Basic-Compiler für den Amiga. Der interaktive Editier-, Kompilier- und Laufzeitzyklus entspricht dem eines Interpreters.

Der integrierte Editor erlaubt eine komfortable Eingabe. HiSoft Basic unterstützt die Eigenschaften des Amiga mit Fenstern, Grafik-Kommandos, Sprite-Handling und Maschinenzugriffen während des Gebrauchs von Bibliotheken. Es ist voll kompatibel mit Amiga-Basic, andere Standarddialekte für den PC, wie z.B. Microsoft QuickBasic, und den Atari ST können mit geringen Modifikationen kompiliert werden. Rekursive

Unterprogramme und Funktionen sind möglich.

Eine Anzahl strukturierter Ausdrücke wie z.B. WHILE...WEND, DO...LOOP UNTIL und SELECT...CASE lassen Sie jede Programmierhürde meistern. Die Größe von Variablen ist nicht beschränkt. Aussagekräftige Fehlermeldungen und Korrekturmöglichkeiten tragen zur komfortablen Handhabung bei.

Hardware-Anforderungen:

Amiga 500, 1000 oder 2000 mit Kickstart 1.2 oder höher. 3 1/2"-Diskette, Bestell-Nr. 54127

DM 179,-*
(sFr 161,- / öS 1790,-*)

Devpac Assembler 2.0

Devpac 2.0 ist ein Entwicklungspaket für den Amiga mit komfortablem Editor/Assembler, symbolischem Debugger und Linker zum Einbinden von Hochsprache-Modulen.

GenAm ist ein 68000er-Makroassembler mit integriertem Bildschirmeditor, der bis zu 75.000 Zeilen pro Minute assemblieren kann. Der 2-Paß-Assembler erzeugt sowohl linkbaren als auch direkt ausführbaren Code. Er unterstützt lokale Labels, die Signifikanz beträgt bis zu 127 Zeichen.

Makros können bis zu 36 Parameter beinhalten und – Rekursion inbegriffen – so tief verschachtelt werden, wie Speicherplatz vorhanden ist. MonAm, der Debugger, erlaubt das Setzen von Breakpoints, das Disassemblieren auf Diskette und noch weitere Features, die das Debuggen zum Vergnügen machen.

Hardware-Anforderungen:

Amiga 500, 1000 oder 2000 mit mindestens 512 Kbyte, ein Diskettenlaufwerk. 3 1/2"-Diskette, Bestell-Nr. 54131

DM 149,-*
(sFr 135,- / öS 1490,-*)

Devpac-Assembler 2.0 im Test! Amiga-Magazin 4/89: Eines der besten Programme seiner Art für den Amiga.

SuperED C

Multitasking-fähiger Editor als Programmierumgebung für den Aztec-C-Compiler (V 3.6). Bestell-Nr. 54139

DM 39,-*
(sFr 35,- / öS 390,-*)

* Unverbindliche Preisempfehlung


Markt & Technik
Zeitschriften · Bücher
Software · Schulung

INFO-COUPON

Bitte senden Sie mir weiteres Informationsmaterial zu
☐ Devpac-Assembler ☐ HiSoft-Basic-Compiler

Name _____

Straße _____

PLZ/Ort _____

Bitte ausschneiden und einsenden an: Markt & Technik Verlag AG, Buch- und Software-Verlag, Hans-Pinsel-Straße 2, 8013 Haar bei München

2211/807



Ami 1

AMIGA

SONDERHEFT

1 Rund um den Amiga

Ausführliche Kurse, Bauleitungen, Tips & Tricks

Das Sonderheft 1 enthält einen ausführlichen Basic-Kurs - alles über den richtigen Umgang der alternativen Benutzeroberfläche "Command Interface Line". Bauleitung: Mit dem Digitizer und der zugehörigen Software "Digisoft Plus" tauchen Sie ein in die Welt der digitalen Klänge. Tips & Tricks: aufgeteilt nach vier Bereichen: Einsteiger, Insider, Amiga-Basic und CLI.



2 Listings der Spitzenklasse

Tolle Anwendungen, Tips, Tricks und Tools

"Fontdesigner", "Objekt-Editor", "Haushaltsbuch", "Keyboard-Master", "Fast Load-Copy", "Super-Copy" sind nur einige von vielen Anwendungen im Sonderheft 2. Außerdem: Ratschläge zu Superbase und zwei außergewöhnliche Spiele.

3 C und Assembler

Zwei Super-Kurse und viele Tips und Tricks

Der große C-Kurs für alle Umsteiger. Programmieren Sie ab sofort mit kräftiger Unterstützung durch Betriebssystem und Libraries. Der Assembler-Kurs enthüllt alle Geheimnisse des 68000er Prozessors.



4 Basic und Spiele

10 Spitzenspiele zum Abtippen in AMIGA-Basic

Die absoluten Spiele Top-Hits für den AMIGA in einer Übersicht. Mit Klassikern wie "Shanghai" und vielen brandneuen Hits. Ein ausführlicher Basic-Kurs führt zu Programmen mit rasanten Geschwindigkeiten.

5 Public Domain

Software zum Nulltarif: Das Beste aus aller Welt

Die ganze Faszination der Grafik in einem Projekt. Wichtigster Themenbereich werden Utilities mit zahlreichen Hilfsprogrammen sein.



7 Spieleprogrammierung

Knowhow, Projekte, Spiele

Knowhow: Wir lüften die Geheimnisse von Blitter, Copper & Co. Projekte: Die erste 3D-Vektor Library für den Amiga ermöglicht völlig neue Erlebnisse in der Grafik-Welt. Lesen Sie über die Entstehung neuer Spiele von der Idee bis zum Programm. Spiele: Fünf brandheiße Spiele aus diesem Heft sind - neben vielen Tools und Utilities - auf Diskette erhältlich.



6 Anwendungen

Workshops: Experten verraten alle Tricks im Umgang mit den professionellen Programmen Sonix, Superbase und Beckertext. Überblick: Anwendungsprogramme und die Vorstellung der besten Programme. Top-Programme fast umsonst: Hochkarätige Anwendungen zur Textverarbeitung, Kontenführung und Dateiverwaltung bestechen durch ausgereifte Benutzeroberfläche.

Ich bestelle: _____ Ausgaben AMIGA SH Nr. _____
 _____ " " " Nr. _____
 _____ " " " Nr. _____
 _____ " " " Nr. _____
 _____ " " " Nr. _____

Insgesamt: _____ Ausgaben für 16,-DM pro Exemplar

Name, Vorname _____

Straße, Hausnummer _____

PLZ, Wohnort _____

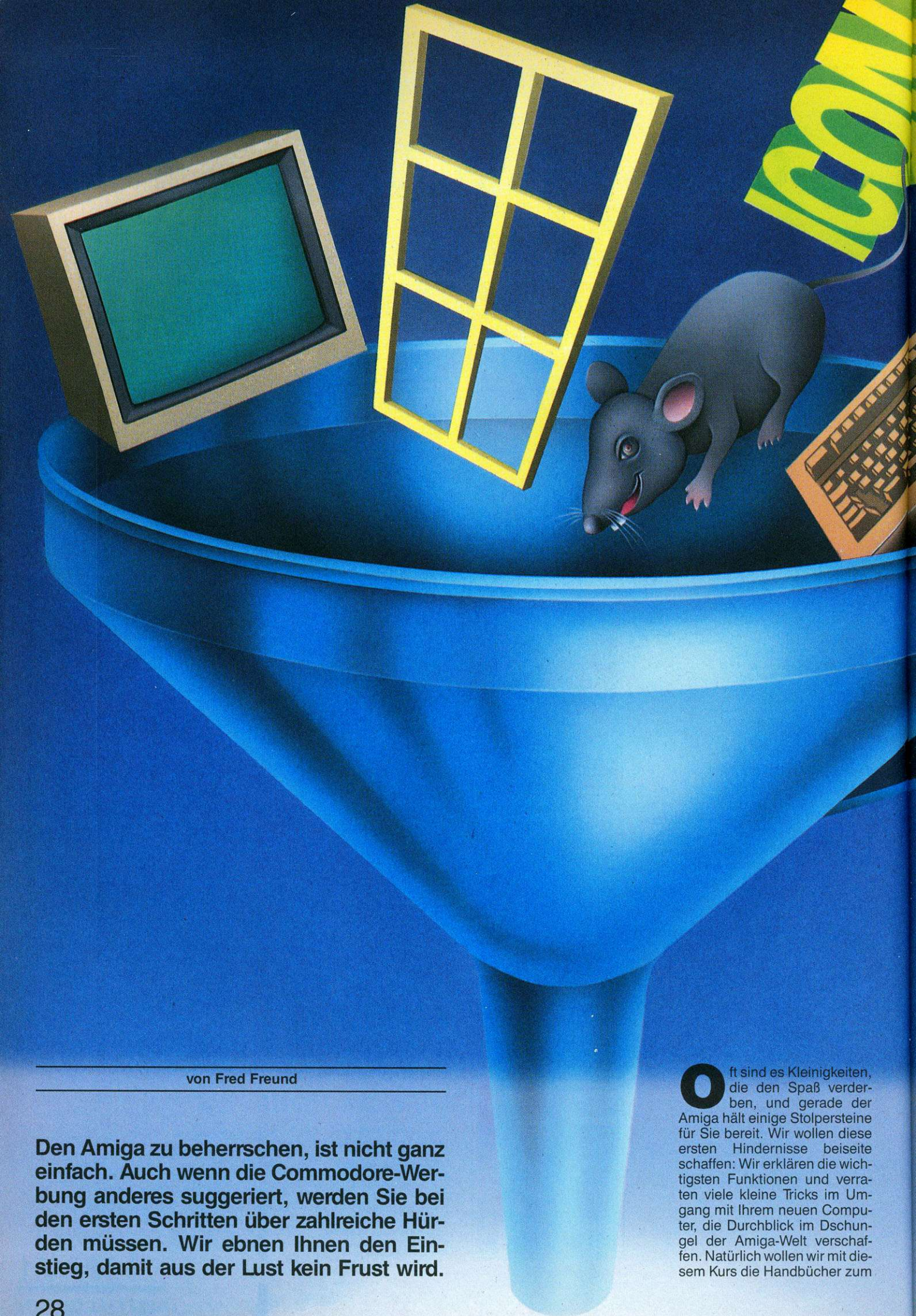
Rechnungssumme _____ DM

zzgl. 3,-DM Versandkosten _____ 3,-DM

Rechnungsbetrag insgesamt _____ DM

Schicken Sie bitte die ausgefüllte Karte zusammen mit dem Rechnungsbetrag als Verrechnungsscheck im Briefumschlag an: Markt & Technik Verlag AG, Leserservice, Hans-Pinsel-Str.2 in 8013 Haar bei München

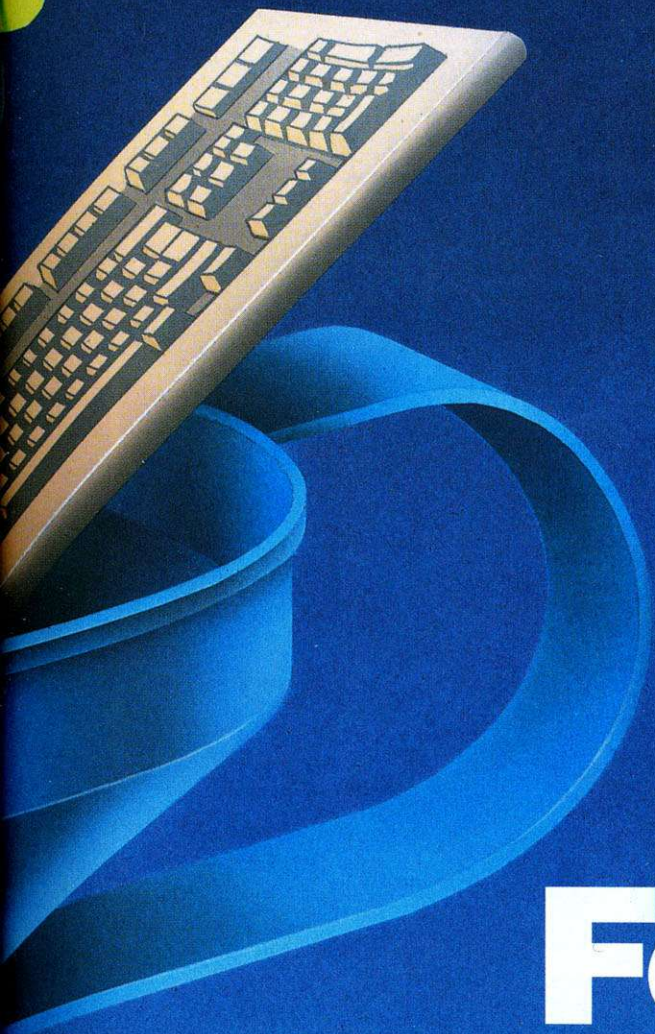
AMIGA
BESTELLCOUPON



von Fred Freund

Den Amiga zu beherrschen, ist nicht ganz einfach. Auch wenn die Commodore-Werbung anderes suggeriert, werden Sie bei den ersten Schritten über zahlreiche Hürden müssen. Wir ebnen Ihnen den Einstieg, damit aus der Lust kein Frust wird.

Oft sind es Kleinigkeiten, die den Spaß verderben, und gerade der Amiga hält einige Stolpersteine für Sie bereit. Wir wollen diese ersten Hindernisse beiseite schaffen: Wir erklären die wichtigsten Funktionen und verraten viele kleine Tricks im Umgang mit Ihrem neuen Computer, die Durchblick im Dschungel der Amiga-Welt verschaffen. Natürlich wollen wir mit diesem Kurs die Handbücher zum



Erste Schritte

Von Fenstern, Mäusen und Konsolen

Amiga nicht ersetzen. Aber während man in den mehr als 300 Seiten dicken Wälzern leicht die Übersicht verliert, bieten wir auf den Punkt bezogene Tips für die ersten Schritte.

Einsteigerthema Nummer eins: Welchen Computer brauche ich, welcher Amiga ist der richtige? Sie haben die Wahl: Der Hersteller Commodore bietet zwei Computer dieser Klasse an, den Amiga 500 (Bild 1) und den Amiga 2000 (Bild 2).

Der wichtigste Unterschied: Der Amiga 2000 (ebenso wie seine neueste Variante der 2500) kann durch Steckkarten erweitert werden; so läßt er sich durch eine entsprechende Karte zu einem MS-DOS-kompatiblen PC aufrüsten: Sie besitzen danach zwei Computer in einem Gehäuse. Auch der Amiga 500 kann zwar durch zusätzliche Diskettenlaufwerke, Festplatten und vieles mehr ausgebaut werden, allerdings müs-

sen neu hinzukommende Geräte außen an den Computer angeschlossen werden. Kabelsalat und Platznot auf dem Schreibtisch sind da unvermeidlich. Die Umrüstung in einen MS-DOS-PC ist nicht möglich. Sie erkennen den Amiga 500 schon rein äußerlich daran, daß die Tastatur, das Diskettenlaufwerk und der Computer selbst in einem einzigen Gehäuse untergebracht sind, er ist ein sogenannter Tastatur-Com-

puter. Der Amiga 2000 dagegen besitzt eine externe Tastatur, einen größeren Hauptspeicher – dafür ist er allerdings auch um ca. 900 Mark teurer als sein kleiner Bruder.

Neben diesen beiden Systemen gibt es noch den Ur-Amiga, Amiga 1000 (Bild 3) genannt. Viele Kleinanzeigen bieten diesen Veteran, der nicht mehr ausgeliefert wird, recht preisgünstig als Gebrauchtgerät an. Nach den Weiterent-

AMIGA

BOOK- WARE

Profi-Software unter 100 DM



R. Arbing, I. Krüger

Scriptum

Das leistungsfähige Textverarbeitungsprogramm für den Amiga. Dazu eine ausführliche Programmbeschreibung im Buch, die bei allen Fragen weiterhilft.

1989, 141 Seiten,
inkl. Programmdiskette.
ISBN 3-89090-650-8

DM 79,-* (sFr 72,70*/öS 672,-*)

Precision Software

Amiga Superbase

Die Einsteiger-Datenbank. Relationales System zum Verwalten von Daten aller Art, auch von IFF-Grafiken; mit integrierter Diashow-Funktion.

1989, 176 Seiten,
inkl. Programmdiskette.
ISBN 3-89090-791-1

DM 89,-* (sFr 81,90*/öS 757,-*)

Atlantis **AmigaCall**

Treten Sie ein in die faszinierende Welt der Datenfernübertragung.

Kommunizieren Sie über Mailboxen mit erfahrenen Computer-Anwendern, die Ihnen bei Ihren Problemen weiterhelfen können, oder Sie erhalten auf diesem Wege leistungsfähige Public-Domain-Software.

1988, 133 Seiten,
inklusive Programmdiskette.
ISBN 3-89090-716-4,
DM 99,-* (sFr 91,-*/öS 842,-*)

C. Fuchs **Reflections**

Traumwelt und Realismus - Bilder im Raytracing-Verfahren auf dem Amiga. Dazu im Buch eine ausführliche Bedienungsanleitung

1989, 156 Seiten,
inklusive Programmdiskette.
ISBN 3-89090-727-X,
DM 98,-* (sFr 90,20*, öS 834,-*)

Atlantis **Trickstudio A**

Animationen in Bild und Ton. Ein Programm zum einfachen Erstellen und Abspielen von bewegten Bildern, die synchron mit Sound unterlegt werden können. Dazu eine ausführliche Dokumentation für die effektive Anwendung.

1988, 86 Seiten,
inklusive Programmdiskette.
ISBN 3-89090-715-6,
DM 99,-* (sFr 91,-*/öS 842,-*)

H. Knappe **Amiga Sounder**

Der Amiga Sounder ist ein Komplett-paket für den Einstieg in die Welt der digitalen Klänge.

1989, 336 Seiten, inkl. 2 Programm-disketten.
ISBN 3-89090-709-1,
DM 98,-* (sFr 90,20*/öS 834,-*)

In Vorbereitung:

Dr. Glaeser/T. Grohser
3-D-Sprinter Amiga, Version 1.3

interaktive Echtzeit-Animation
Lieferbar 4. Quartal 1989
ca. 250 Seiten, inkl. Diskette
ISBN 3-89090-109-3
ca. DM 98,-* (sFr 90,20*/öS 834,-*)

In Vorbereitung

N. Wirsing

Amiga Audio Entwickler-Paket

Dieses Buch macht Sie zum perfekten Amiga-Tontechniker.
Lieferbar 1. Quartal '90,
ca. 400 Seiten, inkl. 2 Programm-disketten.

ISBN 3-89090-765-2,
ca. DM 98,-* (sFr 90,20*/öS 834,-*)

*Unverbindliche Preisempfehlung



Markt & Technik
Zeitschriften · Bücher
Software · Schulung

INFO-COUPON

Bitte senden Sie mir Ihr Gesamtverzeichnis mit über 500 aktuellen Computerbüchern und Software.

Name _____

Straße _____

PLZ/Ort _____

Bitte ausschneiden und einsenden an: Markt & Technik Verlag AG, Buch- und Software-Verlag, Hans-Pinsel-Straße 2, 8013 Haar bei München

Ami SH 8

1205/907

Markt & Technik-Bücher und -Software erhalten Sie in den Fachabteilungen der Warenhäuser, im Versandhandel, in Computer-Fachgeschäften oder bei Ihrem Buchhändler.

wicklungen der 500er- bzw. 2000er-Reihe entspricht das alte Modell nicht mehr dem aktuellen Leistungsstand.

Sowohl vom Amiga 500 als auch vom Amiga 2000 sind verschiedene Versionen auf dem Markt. Äußerlich nicht zu erkennen sind Unterschiede, die das Betriebssystem betreffen. Das ist ein Programm, das die Betriebsmittel des Computers (z.B. Zentraleinheit, Diskettenlaufwerke, Tastatur) verwaltet. Ohne Betriebssystem ist der Amiga nicht einsatzbereit – wie ein Auto ohne Motor. Ein Teil des Amiga-Betriebssystems besteht aus der sog. Kickstart-Software, die in Form eines kleinen Bausteins im Computer verborgen ist.

Die neueste Kickstart-Version für Amiga 500 und Amiga 2000 ist die Version 1.3; sie bringt gegenüber dem Vorgänger, der Version 1.2, einige Verbesserungen, die Sie sich nicht entgehen lassen sollten. Ihr entscheidender Vorteil: Der Amiga kann mit dem neuen Betriebssystem von einer sog. RAM-Disk oder einer zusätzlichen Festplatte aus gestartet (gebootet, sprich »gebootet«) werden. Eine Startdiskette (wie die dem System beiliegende Workbench) ist dann nicht mehr nötig. Hier und da werden noch ältere Modelle in den Lagern der Händler stehen; vergewissern Sie sich also, ob der Amiga Ihrer Wahl bereits über das verbesserte Betriebssystem verfügt. Am Einschaltbild (erscheint, wenn keine Diskette im Laufwerk liegt – siehe Bild 4) erkennen Sie, welche Kickstart-Version eingebaut ist. Sie werden durch eine Grafik aufgefordert, die Workbench einzulegen. Achten Sie beim Kauf darauf, daß Ihr Amiga die Version 1.3 der Workbench-Diskette verlangt und daß diese auch dem Computer beiliegt.

Immer up to date

Seit rund einem Jahr werden die Amiga 500-Modelle leider mit einer qualitativ schlechteren Tastatur ausgeliefert. Besaß die alte Tastatur einen harten, präzisen Anschlag, so vermittelt die neue ein schwammiges Schreibgefühl. Vielschreiber, die auf die zusätzlichen Funktionen von Kickstart 1.3 verzichten können, sollten also den Kauf des älteren, über den Gebrauchtmarkt erhältlichen Mo-

dells in ihre Überlegungen mit einbeziehen.

Zum Monitor: Als preiswerte Alternative zu den Original-Monitoren von Commodore, dem »1084« und dem »1084S« (Stereo-Version mit zwei eingebauten Lautsprechern) bietet sich beispielsweise das Modell »CM 8833« von Philips an. Seine technischen Eigenschaften sind

Blautönen gehaltene Einschaltbild erscheinen. Nein? Dann öffnen Sie die Klappe an der Vorderfront des Monitors (sie klappt nach unten hin auf) und drücken den schwarzen Schalter ganz rechts. Nun sollte das Einschaltbild zu sehen sein. Dem System liegen drei Disketten bei, eine mit der Aufschrift »Amiga Workbench 1.X« und ei-

liche Bilder) kennzeichnend für eine grafische Benutzeroberfläche. Die Maus ist dafür das wichtigste Eingabegerät. Legen Sie die Maus so auf den Tisch, daß in jede Richtung genügend Spielraum für Bewegung bleibt. Legen Sie eine Hand auf die Maus und bewegen Sie sie hin und her. Erkennen Sie die Wirkung auf dem Bildschirm? Richtig, der rote Pfeil wandert Ihren Bewegungen entsprechend über die »Mattscheibe«. Drücken Sie

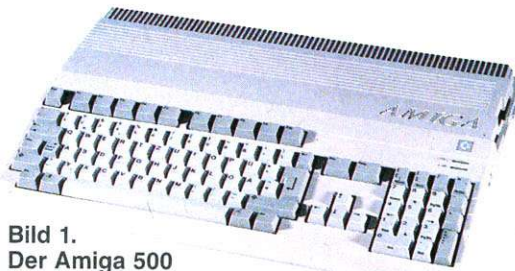


Bild 1.
Der Amiga 500
ist ein sog.
Tastaturcomputer



Bild 2. Der
2000er ist mit
Hilfe von
Steckkarten
sehr leicht erweiterbar



Bild 3. Der
Urvater der Amiga-Familie:
Amiga 1000

denen des Commodore-Stereo-Monitors ebenbürtig, er wird aber häufig etwas preiswerter angeboten.

Wenn Sie den Monitor eines Fremdherstellers kaufen, wird allerdings zusätzlich der Preis für ein Verbindungskabel fällig, also rund 50 bis 60 Mark. Commodore legt das Kabel nämlich nicht dem Amiga bei, sondern den dazu passenden Monitoren. Der Amiga kann über einen zusätzlichen TV-Modulator zwar auch an einen Fernseher angeschlossen werden, wegen des damit verbundenen schlechten Bildes ist dieser Weg allerdings nicht empfehlenswert.

Inzwischen sind Sie zum Händler marschiert und haben sich nach reiflicher Überlegung für Ihren Wunsch-Amiga entschieden. Jetzt steht er zu Hause auf dem Schreibtisch, und Sie müssen die einzelnen Bestandteile erst einmal miteinander verbinden. Ab Seite 52 finden Sie einen Artikel, der die Installation Ihres neu erstellten Traumcomputers erklärt.

Startklar? Dann schalten Sie die Geräte an. Beim Amiga 500 befindet sich der Schalter am Netzteil, beim Amiga 2000 an der Rückfront des Gerätes. Drücken Sie beim Monitor auf die Taste mit der Beschriftung »Power«, dann müßte das in

ne weitere mit der Bezeichnung »Amiga Extras 1.X«. Die dritte heißt »Erste Schritte«.

Jetzt signalisiert Ihnen der Amiga mit der oben beschriebenen Einschaltmeldung, daß er gerne die Workbench-Diskette im internen Diskettenlaufwerk hätte. Beim Amiga 500 legen Sie die Startdiskette immer in das eingebaute Laufwerk.

Den Amiga 2000 bekommt man auch mit zwei eingebauten Laufwerken. Schieben Sie Ihre Workbench-Diskette in das rechte, falls Sie stolzer Besitzer zweier Laufwerke sein sollten. Legen Sie die Diskette ein, leuchtet die grüne (beim Amiga 2000 die rote) Anzeige für Diskettenbetrieb auf. Achtung: Sie dürfen **niemals** eine Diskette aus einem Laufwerk herausnehmen, solange diese Anzeige leuchtet. Alle auf einer Diskette gespeicherten Daten könnten dabei verlorengehen.

Der Amiga lädt nun die sog. Workbench (dt. Werkbank) in den Arbeitsspeicher. Diese Workbench ist eine sog. grafische Benutzeroberfläche, auf der Sie (fast) alle wichtigen Arbeitsschritte erledigen können – im Gegensatz zu einer Kommandooberfläche, auf der wie beispielsweise beim C64 nur Text zu sehen ist, sind kleine Piktogramme (einfache symbo-

jetzt auf die rechte Maustaste, Menütaste genannt, und halten diese fest. Die weiße Zeile am oberen Bildschirmrand verändert sich: Dort tauchen die Begriffe »Workbench, Disk, Special« auf. Das sind die Namen der sog. Menüs, aus denen man verschiedene Befehle auswählen kann. Fahren Sie den Mauszeiger (bei gedrückter Menütaste) auf den Begriff »Special«. Unter dem gewählten Menü klappt ein Rechteck auf, das fünf sog. Menüpunkte enthält.

Befehle à la carte

Halten Sie die rechte Maustaste weiterhin fest und bewegen Sie den Zeiger auf den Begriff »Version«. Lassen Sie die Taste los, verschwindet das Rechteck und in der weißen Leiste am oberen Bildschirmrand erscheint die Angabe »Kickstart Version XX.X. Workbench Version XX.XX«. Haben Sie gemerkt, daß Sie eben den ersten Befehl eingegeben haben? Sie haben den Amiga dazu gebracht, die Versionsnummern des eingebauten Kickstarts und der Workbench anzuzeigen, von der Sie das System gestartet haben. Das war doch ganz einfach! Dieses Konzept der Menüs und Menüpunkte finden Sie bei fast allen Programmen für den Amiga wieder. Dabei gilt immer die Regel, daß die rechte

Maustaste gedrückt werden muß, damit das Menü in der oberen Zeile angezeigt wird.

Das Geräusch, das man beim Drücken einer Maustaste hört, ist ein markantes »Klick«. Einigen wir uns also darauf, das Anwählen von Punkten einfach Anklicken zu nennen. Auf diesen Ausdruck werden Sie während Ihrer Amiga-Laufbahn immer wieder stoßen.

Die Bedeutung der rechten Maustaste ist jetzt klar. Die linke Taste, Auswahlstaste genannt, ist allerdings ebenso wichtig. Fahren Sie mit dem Mauszeiger auf das schwarze Diskettensymbol, unter dem der Begriff »Workbench X.XX« steht.

Klicken Sie jetzt einmal kurz mit der linken Taste. Die Farbe des Symbols, das in der Computer-Sprache Icon (sprich »Eikon«) genannt wird, verändert sich. Bewegen Sie den Zeiger auf den freien blauen Hintergrund und drücken erneut die Auswahlstaste: Das Diskettensymbol erscheint wieder in der ursprünglichen Farbe.

Natürlich hat das Anklicken mit der linken Maustaste noch einen weiteren Effekt als den des Farbwechsels. Klicken Sie noch einmal das Diskettensymbol an, bewegen Sie den Mauszeiger in die Menüleiste. Halten Sie nun die rechte Maustaste gedrückt und fahren mit dem Zeiger auf den Punkt »Info« unter dem Menü mit dem Namen »Workbench«. Lassen Sie die Maustaste los. Wenn Sie alles richtig gemacht haben, erscheint statt des Zeigers ein anderes Symbol, eine Art Sprechblase: »zz«; sie signalisiert Ihnen, daß Sie einen Moment warten müssen. Außerdem leuchtet die Laufwerksanzeige grün (bzw. rot) auf.

Sie haben dem Amiga mit diesen Schritten erneut einen Befehl gegeben: Der Bildschirm verändert sich, es erscheint ein buntes Feld, ein sogenanntes Fenster mit INFORMATIONEN zu der Diskette im Laufwerk (Bild 5). Wichtig ist für uns, daß wir den Umgang mit der Workbench kennenlernen, der Inhalt des Info-Fensters soll uns hier noch nicht weiter interessieren. Wollen Sie einmal den gesamten Rahmen des Fensters verschieben? Das geht ganz einfach: Bewegen Sie den Mauszeiger in die Mitte der Oberzeile des Fensters. Die Oberzeile erkennen Sie daran, daß sie zwei waagerechte blaue Balken enthält. Halten Sie nun die Auswahlstaste (linke Maustaste) fest und bewegen die Maus. Das

Fenster verändert seine Lage auf dem Bildschirm. Sollte dabei wieder ein Symbol mit zwei Z's erscheinen, bewegen Sie den Mauszeiger in das Info-Fenster und drücken die Auswahlstaste. Der Zeiger hat dann wieder sein gewohntes Erscheinungsbild.

Nicht nur die Lage des Fensters ist variabel, auch die Größe können Sie verändern. Bewegen Sie den Mauszeiger in die rechte untere Ecke des Fensters auf das Symbol mit den zwei unterschiedlich großen Rechtecken. Halten Sie die linke Maustaste gedrückt und schieben Sie die Maus nach links hinten. Lassen Sie anschließend die Taste los: Das Fenster ist kleiner geworden und zeigt nun nur noch einen Ausschnitt der bisherigen Informationen an.

Durchblick mit Fenstern

Jetzt verstehen Sie vielleicht auch, warum der Kasten »Fenster« genannt wird: Wie durch ein Fenster hindurch sieht der Anwender beim Amiga immer einen Ausschnitt der dahinterliegenden Informationen.

Spielen Sie ruhig noch etwas mit dem Fenster herum. Haben Sie genug davon, schließen Sie das Fenster. Wie? Ganz einfach,

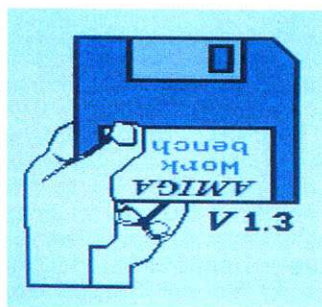


Bild 4. Die Einschaltmeldung des neuesten Amiga verlangt nach der Workbench 1.3

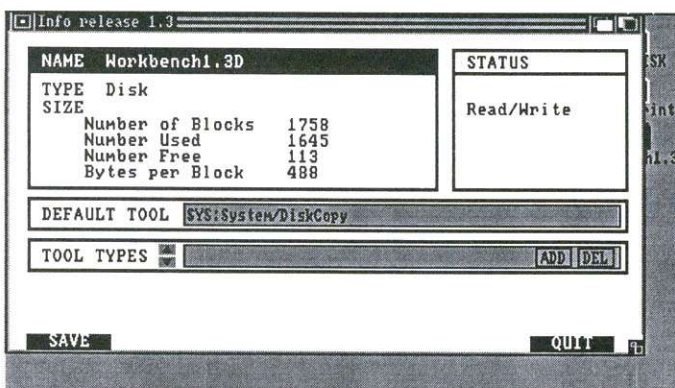


Bild 5. Das Info-Fenster für die Workbench-Diskette

bewegen Sie den Mauszeiger auf das Kästchen mit dem Punkt in der Mitte, das die linke obere Ecke des Fensters bildet. Klicken Sie mit der linken Maustaste, das Fenster verschwindet vom Bildschirm.

Klicken Sie noch einmal in die blaue Fläche der Workbench, ist der Ausgangszustand wiederhergestellt. Sie haben damit ganz spielerisch einige Funktionen zur Bedienung des Amiga kennengelernt: die Funktion der Maustasten, die Menüleisten und -punkte, Sie können jetzt das Info-Fenster öffnen und Fenster verschieben oder schließen.

Bevor wir weitermachen, sollten Sie sich zur Sicherheit erst einmal eine Kopie der Workbench-Diskette machen. Sie brauchen dazu eine weitere, leere Diskette; der kleine schwarze Schieber in der Ecke der leeren Diskette muß so stehen, daß man durch die Öffnung nicht hindurchsehen kann. Befolgen Sie einfach folgende Schritte, dann kann nichts schiefgehen:

- nehmen Sie die Workbench-Diskette aus dem Laufwerk
- aktivieren Sie den Schreibschutz der Workbench-Diskette (schwarzen Schieber auf Durchlaß stellen)
- legen Sie die leere Diskette in das Laufwerk (ein neues Symbol erscheint)
- fahren Sie mit dem Mauszeiger auf das Diskettensymbol namens »Workbench 1.X«
- halten Sie die linke Maustaste fest und ziehen dabei das Symbol auf das neue Symbol der leeren Diskette
- lassen Sie die Maustaste los
- warten Sie die Aufforderung in der linken oberen Ecke ab (Bild 6) und legen Sie die Workbench-Diskette ein
- fahren Sie mit dem Mauszeiger auf das eben erschienene kleine Kästchen mit der Bezeichnung »Retry« und drücken die linke Maustaste

- wenn der Mauszeiger auf »Continue« steht, drücken Sie wieder die linke Maustaste - jedesmal, wenn der Kasten »Put...disk...« erscheint (Bild 7), wechseln Sie die Disketten und drücken die linke Maustaste (Zeiger auf »Continue«)

Die Kopie erhält automatisch den Namen »copy of Workbench 1.X«. Legen Sie nun die



Bild 6. Mit sog. Systemrequestern werden Sie aufgefordert, eine Diskette einzulegen



Bild 7. Beim Kopieren mit einem Laufwerk sind mehrere Diskettenwechsel notwendig

Kopie in das Laufwerk und starten Ihren Amiga von neuem, indem Sie gleichzeitig die drei Tasten »CTRL«, »linke Amiga« und »rechte Amiga« drücken.

Bisher haben wir uns nur mit dem Info-Fenster beschäftigt. Wie kann man ein anderes als das Info-Fenster öffnen? Auch das ist wieder sehr leicht: Steuern Sie erneut mit dem Mauszeiger das schon häufiger strazierte Diskettensymbol an, klicken einmal mit der Auswahlstaste und wählen dann in der Menüleiste unter der Rubrik »Workbench« den Punkt »OPEN«. Ein bisher unbekanntes Fenster öffnet sich, in dem Schubladensymbole, ein Symbol für ein Fenster (Untertitel »Shell« oder »CLI«) sowie ein Mülleimer (Trashcan) erscheinen. Wir sehen jetzt zum ersten Mal den Inhalt einer Diskette, und zwar unserer Workbench-Diskette. Probieren Sie das gleiche mit den Schubladen aus. Auch hier öffnet sich jeweils ein neues Fenster. Nach einiger Zeit könnte Ihr Bildschirm etwa so wie in Bild 8 aussehen.

Das Hin und Her mit dem Mauszeiger zum Öffnen eines Fensters geht Ihnen auf die Nerven? Dann wählen wir jetzt eine schnellere Methode zum Öffnen: Klicken Sie das Schließsymbol des Fensters mit dem Namen »Utilities« (frei übersetzt Hilfsprogramme) an.

AMIGA
SONDERHEFT

PROGRAMM-SERVICE

Direkt bestellen statt abtippen!

Amiga-Sonderheft 8

Spiel Nr. 1:

»Maze«

»Maze« ist das Spiel für Denker und Lenker. Schnelles Auffassungsvermögen und eine ausgeklügelte Taktik entscheiden dieses Spiel. Ein langes Spielvergnügen ist durch den eingebauten Level-Editor garantiert.

Weitere Informationen über dieses faszinierende Spiel finden Sie auf Seite 106 in diesem Heft.

Spiel Nr. 2:

»Oel-Fieber«

Steigen Sie ein in das Millionen-geschäft um Öl und Aktien. Ihre Fähigkeiten als Manager werden auf immer neue Bewährungsproben gestellt. Das amüsante Würfelspiel erfordert dabei eine Menge Kreativität, um immer neue Strategien zu entwickeln. Wie Sie mit Öl- und Aktiengeschäften reich werden können, lesen Sie in diesem Heft ab Seite 89.

Sicher ist sicher

Unumgänglich für jeden Amiga-User ist ein leistungsfähiges Anti-Viren-Programm. »Virus-Control v 1.4« schützt Sie vor allen bekannten Bootblock- und Linkviren sowie durch seine Lernfähigkeit auch vor zukünftigen »Infektionskrankheiten«. Schaffen Sie sich mit »Virus-Control v 1.4« ein zuverlässiges Immunsystem für Ihren Amiga. Die Anleitung finden Sie auf der Diskette.

Bestell-Nr. 45908

DM 29,90* sFr 24,90*/öS 299,-*

*Unverbindliche Preisempfehlung



**Weitere Angebote
auf der Rückseite!**

AMIGA PROGRAMMSERVICE

Sie suchen hilfreiche Utilities und professionelle Anwendungen für Ihren Computer? Sie wünschen sich gute Software zu vernünftigen Preisen? Hier finden Sie beides!

Unser stetig wachsendes Sortiment enthält interessante Listing-Software für alle gängigen Computertypen. Jeden Monat erweitert sich unser aktuelles Angebot um eine weitere interessante Programmsammlung für jeweils einen Computertyp. Bei Fragen zu Bestellung und Versand der Programmservice-Disketten wählen Sie bitte:

Telefon (089) 46 13-232.

Bestellungen bitte nur gegen Vorauskasse an: Markt & Technik Verlag AG, Buch- und Software-Verlag, Hans-Pinsel-Straße 2, D-8013 Haar, Telefon (089) 46 13-0. Schweiz: Markt & Technik Vertriebs AG, Kollerstrasse 37, CH-6300 Zug, Telefon (042) 440 550. Österreich: Markt & Technik Verlag Gesellschaft m.b.H., Große Neugasse 28, A-1040 Wien, Telefon (0222) 587 1393-0. Microcomput-ique, E. Schiller, Gögglstraße 17, A-3500 Krems, Telefon (02732) 74193. MES-Versand, Postfach 15, A-3485 Haitzendorf, Bücherzentrum Meidling, Schönbrunner Straße 261, A-1120 Wien, Telefon (0222) 833196. Bestellungen aus anderen Ländern bitte nur schriftlich an: Markt & Technik Verlag AG, Abt. Buchvertrieb, Hans-Pinsel-Straße 2, D-8013 Haar. Nur gegen Bezahlung der Rechnung im voraus.

Bitte verwenden Sie für Ihre Bestellung und Überweisung die beigeheftete Postgiro-Zahlkarte, oder senden Sie uns einen Verrechnungsscheck mit Ihrer Bestellung. Sie erleichtern uns die Auftragsabwicklung, und dafür berechnen wir Ihnen keine Versandkosten.

Bit für Bit nur Hits... ...das Beste aus zwei Jahren Amiga-Magazin!

Wir haben auf zwei Disketten die interessantesten Programmangebote der Jahre 1988 und 1989 für die Bereiche Anwendungen/Tools und Spiele gesammelt. Mit dieser Sammlung legen Sie einen leistungsfähigen Grundstein oder bilden eine sinnvolle Erweiterung für Ihre Programmbibliothek. Und dies alles zum besonders günstigen Paketpreis.

DISK I: Anwendungen/Tools

»AmigaDat« – Die Dateiverwaltung für alle Zwecke, von der Schallplattensammlung über Adressen bis hin zur Videosammlung.

»Manager« – Das komfortable Haushaltsbuch.

»Disketi« – Drucken von Diskettenlabels.

»MasterCruncher« – Leistungsfähige Daten- und Programmkomprimierung.

»Recover« – Retten von gelöschten Daten.

»Resix« – Macht Programme resetfest.

»MouseCreator und PointerMaker« – Generieren Sie Ihre eigenen Mauszeiger.

»DiskSpy« – Problemloses Ändern von Daten direkt auf Diskette.

»AmigaSort« – Bringt Ordnung in Ihre Diskettensammlung.

»Fade« – Einfaches Ein- und Ausblenden von Bildschirmen ermöglicht tolle Effekte.

»VirusControl« – Der komfortable Virenschutz.

»TrapHandle« – Die Möglichkeit bei auftretenden Fehlern Ihr C-Programm sauber zu beenden.

DISK II: Die Spielesammlung, die man haben muß

Action

»Troof« – Das spannende Motorradrennen der Zukunft.

»Quadron« – Geschicklichkeit bei höchsten Geschwindigkeiten sind gefragt.

Spieleumsetzungen

»Poker« – Wann bekommen Sie den Royal-Flush?

»Billard« – Tolle Grafik erwartet Sie bei dieser fantastischen Umsetzung.

»Domino« – Verblüffend Umsetzung des bekannten Spiels.

»Kniffel« – Eine grafisch verblüffende Würfelspielumsetzung für bis zu vier Spieler.

»3D-Tic-Tac-Toe« – Dreidimensionales Spiel zum Kombinieren und Denken.

Best of the Rest

»Eliza« – Der Amiga als Psychotherapeut.

»Arriba« – Die Tastatur lernt sprechen.

Die Beschreibungen der Programme sind als Readme-Files auf den jeweiligen Disketten.

Bestell-Nr. 47901

Zwei Disketten randvoll mit tollen Programmen zum Paketpreis von:

DM 39,90* sFr 39,-*/öS 390,-*

Direkt bestellen statt abtippen! Die aktuelle Diskette zum Heft: Amiga-Sonderheft 7

Reflex: Das neueste Projekt von Arno Gölzer. »Reflex« ist der Nachfolger der beliebten Billardumsetzung. Lassen Sie sich von einer einzigartigen Spielidee und der tollen grafischen Umsetzung begeistern. *Beschreibung auf Seite 66.* **Noch mehr starke Spiele...** Ob Action- oder Strategiespiel – hier finden Sie alles, was das Spielerherz begehrt. **»Fast Freddie«** Die Beschreibung dieses Actionspiels finden Sie auf Seite 86. **»Memory«** Eine interessante grafische Umsetzung für Leute mit gutem Gedächtnis. *Beschreibung auf Seite 97.* **»Reactor«** Das Strategiespiel, das auch Atomkraftgegner fesseln wird. *Beschreibung auf Seite 107.* **»Schiebespiel«** Strategisches Denken und Tüfteln ist bei diesem Spiel das A und O. *Beschreibung auf Seite 88.*

Vector-Grafik-Library: Nutzen Sie die ganze Faszination der Vektor-Grafik. Mit dieser Bibliothek bauen Sie Vector-Grafiken auf einfache und schnelle Weise in Ihre Programme ein. Der Vorteil: Es wird nur wenig Speicherplatz und Rechenzeit verbraucht. *Beschreibung auf Seite 22f.*

Zwei Disketten

Bestell-Nr. 45907

DM 34,90* sFr 29,50*/öS 349,-*

Amiga-Sonderheft 3: Basic, Spiele

Broker: Erleben Sie die Faszination der Börse hautnah. Diese Simulation für 2 bis 4 Spieler ist einzigartig. Der Autor setzt seine fundierten Kenntnisse in spannendes Spielgeschehen um. **Ping-Pong:** Dieses Sportspiel bringt Wettkampfstimmung ins Wohnzimmer. Dreidimensionale Darstellung, realistische Soundeffekte und rasante Ballwechsel führen zu lang anhaltendem Spielspaß. **Anpfiff:** Als Manager in der Fußball-Bundesliga führen Sie Ihr Lieblingsteam durch die Saison. Zusätzliche Spiele im UEFA-Cup verhelfen Ihrer Mannschaft zu Ruhm und Ihnen zu vielen Manager-Punkten. **Basic-Routinen:** Die Basic-Kurse im Sonderheft 3 bieten zahlreiche, hilfreiche Routinen. Alle dort vorgestellten Programme finden Sie auch auf dieser Programmservice-Diskette. Weiterhin befinden sich auf der Diskette alle Programme, die im Inhaltsverzeichnis des Amiga-Sonderhefts 3 mit einem Diskettensymbol gekennzeichnet sind.

3 1/2"-Diskette für Amiga

Bestell-Nr. 45903 **DM 29,90*** sFr 24,90*/öS 299,-*

Amiga-Sonderheft 2: Grafik, Anwendung

Object-Editor: Animierte Figuren, beispielsweise für eigene Spiele, entwickeln Sie mit diesem Editor auf komfortable Weise. Sogar mit Deluxe Paint erstellte Pinsel lassen sich einlesen.

Haushaltsbuch: Mit diesem hervorragenden Anwendungsprogramm verwalten Sie alle Einnahmen und Ausgaben auf übersichtliche Weise. Eine Monats- oder Jahresstatistik zeigt, in welchen Bereichen Sie zukünftig sparen können. Jetzt haben Sie Ihre Finanzen im Griff.

Keyboard-Master: Lernen Sie im Zehn-Finger-System zu tippen. Mit diesem didaktisch ausgereiften Programm ist dies kein Problem. Für Programmierer sind sogar Spezial-Lektionen mit wichtigen Sonderzeichen vorhanden.

FastLoadCopy: Dieses Tool bringt den DIR-Befehl auf Trab. Nach der »Operation« wird das Inhaltsverzeichnis einer Diskette im D-Zug-Tempo eingelesen. Zusätzlich kopiert das Programm Disketten und versieht diese mit dem schnellen Directory.

Weiterhin befinden sich auf der Diskette alle Programme, die im Inhaltsverzeichnis des Amiga-Sonderhefts 2 mit einem Diskettensymbol gekennzeichnet sind.

3 1/2"-Diskette für Amiga

Bestell-Nr. 45802 **DM 29,90*** sFr 24,90*/öS 299,-*

Die Wiederbelebung für die C64-Peripherie

Viele Amiga-Besitzer haben noch einen C64 mit Peripheriegeräten zu Hause stehen. Mit ein bißchen Hard- und Software können Sie diese zu neuem Leben erwecken und Ihre Daten so weiterbenutzen. Dabei ist die Bedienung wirklich einfach.

Der fertig aufgebaute IEC-Handler erlaubt es, alle C64-Geräte wie die Floppy 1541 oder 1571, Commodore-MPS-Drucker und natürlich auch den C64 (zur Datenübertragung) am Amiga zu betreiben. Das Gesamtpaket besteht aus der fertig aufgebauten Platine mit Verbindungskabel, der Treibersoftware auf 3 1/2"-Diskette sowie einer entsprechenden Dokumentation.

Bestell-Nr. 39101 **DM 79,-*** sFr 71,-*/öS 790,-*

* Unverbindliche Preisempfehlung

Wichtig: Mit den Gutscheinen aus dem »Super-Software-Scheckheft« zu DM 149,- können Sie Software-Disketten Ihrer Wahl aus dem Preiswerte Super-Software-Angebot im Wert von DM 180,- bestellen – egal, ob diese DM 19,90, DM 29,90 oder DM 89,- kosten. Sie sparen DM 30,-!

Das Super-Software-Angebot finden Sie in den Zeitschriften

Computer Persönlich, PC Magazin Plus, Amiga-Magazin, Amiga-Sonderheft, 64'er-Magazin, 64'er-Sonderheft, ST-Magazin, PC Magazin, Happy-Computer.

Übrigens: Die Gutscheine können Sie auch übertragen oder verschenken!

Das Scheckheft können Sie per Verrechnungsscheck oder mit der eingelebten Zahlkarte direkt beim Verlag bestellen.

Kennwort: »Super-Software-Scheckheft«, Bestell-Nr. W156

P.S.: Bei Bestellungen bitte kein Bargeld einschicken!

Workbench release. 2856288 free memory

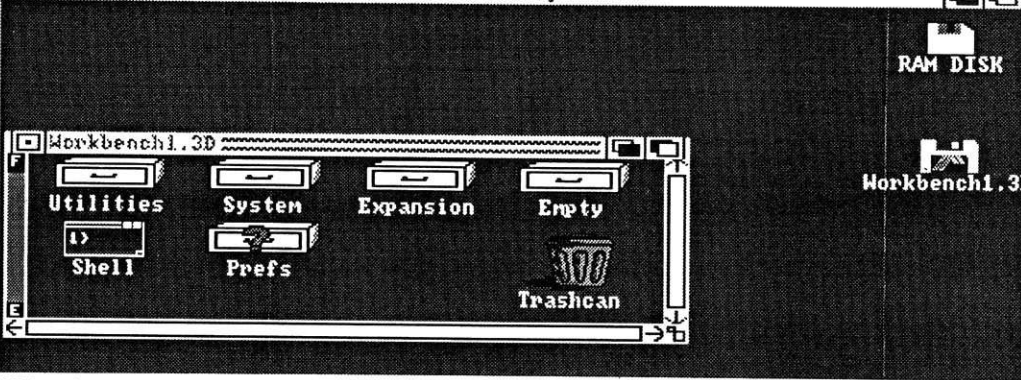


Bild 8. Schon nach kurzem »Klicken« kann der Bildschirm voller Fenster sein

Das Schließsymbol ist das Quadrat mit dem Punkt in der linken oberen Ecke des Fensters, das jetzt wieder verschwindet. Klicken Sie danach die Schublade Utilities mit der linken Maustaste an, und zwar in rascher Folge zweimal hintereinander. Das Utilities-Fenster öffnet sich erneut, Sie haben soeben eine Alternative zum »Open«-Befehl entdeckt. Hat es nicht geklappt? Dann versuchen Sie es ruhig noch einmal, wahrscheinlich haben Sie zwischen den beiden Mausklicks zuviel Zeit verstreichen lassen. Bald schon wird Ihnen das Öffnen eines Fensters durch Doppelklicken in Fleisch und Blut übergegangen sein.

Bisher haben wir uns auf Fenster beschränkt, die als Inhaltsverzeichnis darstellten (Schubladen). Klicken Sie zweimal auf die Schublade mit dem Namen »Prefs«. In dem zugehörigen Fenster finden Sie ein Icon mit dem Namen »Preferences« und einem großen Fragezeichen. Klicken Sie zweimal auf dieses Symbol. Nach kurzer Zeit erscheint ein fast bildschirmfüllendes Fenster (Bild 9). Sie haben soeben Ihr erstes Programm gestartet. Das Fenster von Preferences unterscheidet sich von den bisherigen Fenstern:

- Die Größe läßt sich nicht verstellen.
- Das Fenster läßt sich nur minimal nach oben oder unten verschieben.
- Ein Schließsymbol werden Sie vergeblich suchen.

Bevor Sie jetzt in Panik geraten, weil Sie vermuten, dieses Fenster könnte Ihnen bis zum Ende Ihrer Amiga-Karriere erhalten bleiben, sehen Sie sich die rechte untere Ecke an. »Cancel« bedeutet »Beenden«, und ein Klick in dieses Feld macht genau das.

Starten Sie Preferences erneut. Sie finden rechts oben in

der Titelleiste zwei Symbole. Ein Klick auf das rechte bewirkt auf den ersten Blick überhaupt nichts. Klicken Sie auf das linke Symbol. Sie werden feststellen, daß das Preferences-Fenster hinter alle anderen gerade geöffneten Fenster befördert wird. Bevor Sie anfangen, jedes einzelne Fenster mittels des gleichen Symbols hinter das Preferences-Fenster zu klicken, testen Sie erst das zweite Symbol (ganz rechts oben im Preferences-Fenster). Ist der Sinn dieser Felder damit geklärt?

Spüren Sie jedoch etwas mit den einzelnen Elementen von Preferences. Sie riskieren dabei nichts. Sollte einmal etwas schiefgehen und Sie finden sich nicht mehr zurecht, so finden Sie entweder links oben oder rechts unten den Helfer in

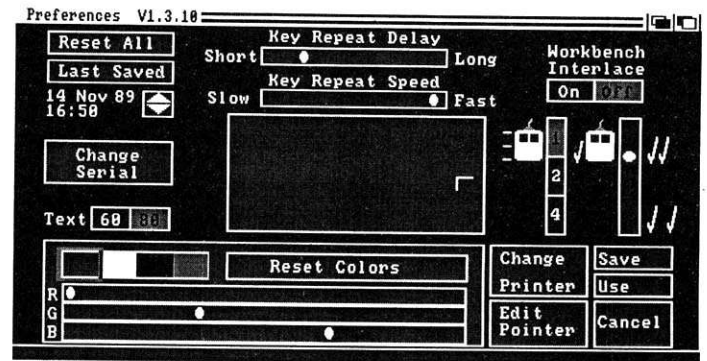


Bild 9. Mit »Preferences« legen Sie die Voreinstellungen der Workbench (Farben, Drucker usw.) fest

der Not: Ein Klick auf »Reset All« stellt die Startwerte wieder her, und mit »Cancel« beenden Sie Preferences, ohne die bis dahin veränderten Werte beizubehalten. Falls Sie mehr über das Preferences-Programm wis-

sen wollen, empfehlen wir Ihnen den Artikel auf Seite 46.

Preferences ist jedoch nicht das einzige Programm auf Ihrer Workbench-Diskette. Klicken Sie zweimal auf das stilisierte Fenster mit dem Namen »Shell« (oder »CLI«). Etwas über der Mitte des Bildschirms öffnet sich ein Fenster mit dem Namen »AmigaShell« (Bild 10) – sollte dort ein anderer Name stehen, beispielsweise »New Cli Window«, so ist dies kein Grund zur Panik, Sie haben lediglich das Programm »CLI« statt »Shell« gestartet, beide

sind jedoch weitgehend identisch.

Dieses Fenster hat mich am Anfang meiner Amiga-Karriere so manchen unnötigen Schweißtropfen gekostet. Mit gutem Grund: Sie können dieses Fenster mit der Maus verschieben, in den Hintergrund klicken, wieder nach vorne holen, vergrößern und verkleinern. Aber Sie werden vergeblich nach einem Schließsymbol oder »Cancel«-Gadget suchen.

Können Sie sich einen frustrierten Einsteiger vorstellen, wie er stundenlang vor seinem neu erstandenen Amiga sitzt und versucht ein eben geöffnetes Fenster wieder zu schließen? Langjährige C64-Erfahrung ließ mich immerhin bei dem hellroten Cursor an eine Aufforderung zur Texteingabe denken. Aber auch Versuche mit »End«, »Quit« und »Exit« fruchteten nichts (außer dem hämischen Kommentar »unknown command exit«) – ebenso die stundenlange Suche im Handbuch, so daß am Ende nur noch der Griff in die Trickkiste blieb: Immerhin konnte ich davon ausgehen, daß dieses widerborstige Fenster nach einem kurzen Aus- und wieder Einschalten verschwunden sein würde. Das Problem war damit jedoch noch nicht gelöst. Denn meine Neugier ließ mich ein-

Menüs der Workbench

Menü	Workbench
Menüpunkt	Funktion
Open	Öffnet Fenster
Close	Schließt Fenster
Duplicate	Kopiert ausgewählte Diskette
Rename	Namen des ausgewählten Symbols ändern
Info	Informationen und Voreinstellungen zu Symbolen
Discard	Schubladen oder Programme endgültig von Diskette löschen
Menü	Disk
Menüpunkt	Funktion
Empty Trash	Löscht alle Daten im Mülleimer, wenn dieser durch Anklicken aktiviert ist
Initialize	Formatiert neue Diskette, alle Daten auf dieser werden gelöscht!
Menü	Special
Menüpunkt	Funktion
Clean Up	Ordnet die Symbole in einem Fenster
Last Error	Zeigt die letzte Fehlermeldung in der Menüleiste an
Redraw	Zeichnet den Bildschirm neu
Snapshot	Hält Neuordnung in einem Fenster auf Dauer fest
Version	Zeigt die aktuelle Kickstart- und Workbench-Version an

Tabelle 1. Alle Menüs der Workbench auf einen Blick

fach nicht in Ruhe. Immer wieder klicke ich auf das Shell-Icon. Immerhin habe ich relativ schnell eine elegante Lösung, die das lästige Booten vermeidet: Ich verkleinere das Fenster soweit wie möglich und schiebe es nach rechts unten, wo es friedlich auf meine Eingaben wartet – Multitasking ist doch etwas Feines. Aber die Optimallösung kann das auch nicht sein. Starten Sie das gleiche Programm mehrmals nacheinander, spätestens beim zwanzigsten Fenster werden Sie beobach-

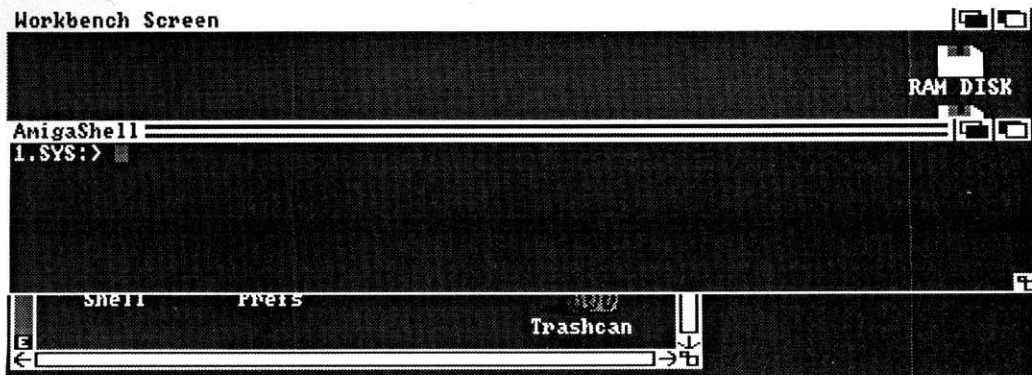


Bild 10. Das Cli-Fenster hat nur wenige der Workbench-typischen Bedienungselemente

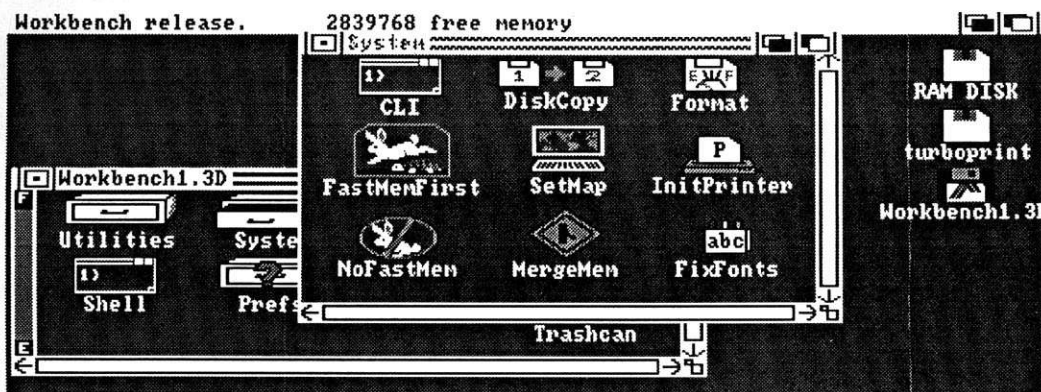


Bild 11. Der System-Ordner der Workbench 1.3 enthält mindestens neun Programme

ten, wie Ihr Amiga deutlich langsamer wird – Multitasking hat auch Nachteile.

Zwar bleibt Ihr Amiga mit 20 Fenstern durchaus arbeitsfähig, aber die mangelhafte Geschwindigkeit macht sich doch recht unangenehm bemerkbar. Wie beseitigen Sie also das »Fenster des Anstoßes«, ohne gleich aus- und wieder einzuschalten? Des Rätsels Lösung heißt »endcli«. Geben Sie diese sechs Zeichen in einem aktivierten Shell-Fenster ein (um sicherzustellen, daß ein Fenster aktiviert ist, klicken Sie einmal innerhalb des Fensters – darauf erscheint die Titelleiste in deutlicher Schrift, während sie vorher leicht verwischt war). Bestätigen Sie Ihre Eingabe mit <RETURN> – das ist die große Taste mit dem geknickten Pfeil auf der rechten Seite der Tastatur. Nach »endcli« verschwindet das Fenster anstandslos.

Gehen Sie jetzt für jedes Fenster auf die gleiche Weise vor (Anklicken – Endcli – <RETURN>), bis alle Fenster verschwunden sind. Umständlich meinen Sie? Okay, aber das sind die Eigenarten Ihres »Command Line Interface« (CLI bzw. Shell). Übrigens gibt es auch dieselbe umständliche Methode, ein Fenster zu öffnen: Starten Sie eine Shell, aktivie-

ren Sie das Fenster, geben Sie »newcli« ein und drücken Sie <RETURN>. Der Effekt ist derselbe wie nach einem Doppelklick auf das Icon – dann schon lieber mit der Maus? Genau meine Meinung.

Bevor Sie weiterlesen, experimentieren Sie etwas mit dem bisher Gelernten. Öffnen Sie einige Fenster, vergrößern und verkleinern Sie diese, klicken Sie sie nach vorne und nach hinten. Sehen Sie sich ruhig auch die Menüs etwas genauer an und vor allem bedenken Sie: Sie können nichts falsch machen – spätestens nach dem nächsten Ausschalten ist alles wieder beim alten.

Haben Sie einige Zeit mit dem bisher Gelernten experimentiert? Dann wollen wir den

nächsten Schritt unternehmen. Sie wissen bereits, wie Sie Fenster auf der Arbeitsoberfläche (der Workbench) öffnen, Disketten kopieren und Programme starten.

Natürlich können Sie mit den Fenstern noch weit mehr anstellen, als Sie bisher sehen konnten. Öffnen Sie beispielsweise das Fenster der Workbench (mit einem Doppelklick auf das Diskettensymbol). Sie sehen am linken Rand einen roten Balken, der anzeigt, wie weit diese Diskette gefüllt ist. Reicht er bis zum »F«, so ist Ihre Workbench randvoll, sollte er – was wenig wahrscheinlich ist – nur ein dünner Strich über dem »E« sein, dann ist noch massenweise Platz. Diese Anzeige werden Sie nur beim Fenster einer Dis-

kette finden, Unterverzeichnisse haben diese Anzeige nicht (sie wäre auch reichlich sinnlos, da der zur Verfügung stehende Platz im Unterverzeichnis direkt von dem auf der Diskette abhängt).

Öffnen Sie die »System«-Schublade. Sie finden darin mindestens neun Programme (Bild 11). Verkleinern Sie das Fenster und beobachten Sie dabei den Fensterrahmen auf der rechten und der unteren Seite. Was passiert mit den beiden Balken? Offensichtlich zeigen diese an, welcher Anteil des Fensterinhalts gerade sichtbar ist (Bild 12). Neben dem Gadget zur Veränderung der Größe finden Sie zwei kleine Pfeile. Klicken Sie diese mit der linken Maustaste an, halten sie fest und beobachten Sie den Fensterinhalt. Jeweils am anderen Ende des Balkens finden Sie den Pfeil, um diese Änderungen wieder rückgängig zu machen.

Vergrößern Sie jetzt Ihr Fenster wieder (am besten etwas größer als zu Anfang). Verschieben Sie die darin enthaltenen Symbole (diese dürfen sich auch überlappen). Ziehen Sie das Fenster etwas beiseite und aktivieren Sie die Schublade »System«, mit der Sie das Fenster geöffnet haben (erinnern Sie sich? »Aktivieren« heißt einmal anklicken). Wählen Sie jetzt

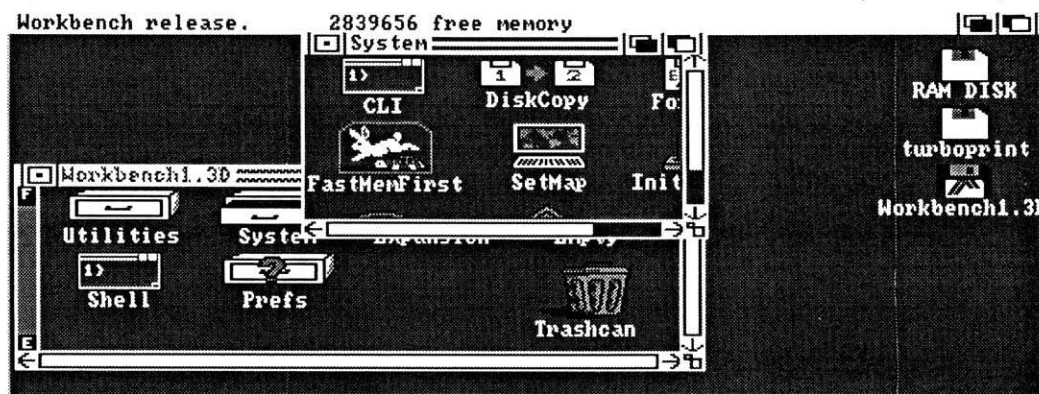


Bild 12. Nach dem Verkleinern sehen Sie nur noch einen Ausschnitt

aus dem »Special«-Menü die Funktion »CleanUp«. Egal wie groß das Chaos war, das Sie vorher angerichtet haben, innerhalb kürzester Zeit ist Ihr Fenster wieder aufgeräumt.

Wollen Sie erreichen, daß ein bestimmtes Icon beim Öffnen des Fensters immer an derselben Stelle erscheint, so verschieben Sie es mit der Maus dorthin, lassen es im aktivierten Zustand und wählen Sie aus dem Special-Menü die Funktion »Snapshot«. Diese speichert – stark vereinfacht – einen

Funktion, die Sie vermutlich von ihm erwarten: Er ist gedacht für alles, was Sie nicht mehr gebrauchen können. Haben wir uns nicht vorhin über die Unsitten der Shell (bzw. des CLI) geärgert? Widerlich zu bedienen und auch noch pingelig in der Annahme gewisser Befehle (es mußte »endcli« heißen, »end« genügt ja nicht). Was soll damit geschehen? – Okay, ab damit in den »Trashcan«. Klicken Sie dazu mit der linken Maustaste auf das Shell-Icon und halten die Maustaste ge-

werfen, lesen Sie erst weiter. Wie Sie vielleicht von anderen Gelegenheiten wissen: Nicht alles, was Sie wegwerfen, ist wirklich verschwunden. Sie brauchen also die Shell noch? Werfen Sie einen Blick in den Mülleimer und holen Sie die Shell wieder hervor. Wie das geht? Klicken Sie zweimal auf das »Trashcan«-Icon. Sie sehen den Inhalt Ihrer Mülltonne auf dem Bildschirm ausgebreitet. Holen Sie sich die Shell wieder zurück (auf dem gleichen Weg wie vorhin mit der Maus). Bevor

Der Inhalt einer Schublade wird auf die oben beschriebene Weise recht anschaulich dargestellt. Aber schon bald werden Sie feststellen, daß Sie dabei nicht alles sehen, was vorhanden ist. Das Betriebssystem macht für Sie nur das sichtbar, womit Sie von der Workbench aus etwas anfangen können. Alles andere wird »versteckt«. Dennoch muß es eine Möglichkeit geben, auch an diese Informationen zu kommen.

Diese Möglichkeit heißt – Sie haben richtig befürchtet – Shell. Starten Sie also Ihre Shell mit einem Doppelklick auf das Icon und legen Sie die Maus beiseite, denn jetzt brauchen Sie die Tastatur. Wie Sie bereits erfahren haben, fordert der rote Cursor (sprich: »Körper« – so nennt man die Markierung für die aktuelle Position im Text) zur Eingabe eines Befehls auf. Sicherlich haben Sie längst vermutet, daß das Öffnen und Schließen eines Fensters nicht die einzigen Aufgaben sind, die auf diesem mühsamen Weg erledigt werden. Beispielsweise erhalten Sie hier eine Menge Informationen, auf die Sie anders keinen Zugriff hätten.

Sie wollen wissen, ob auf Ihrer Diskette noch mehr Informationen enthalten sind als die auf der Workbench sichtbaren? Kein Problem – vergrößern Sie zunächst Ihr Shell-Fenster auf die volle Bildschirmgröße, geben Sie folgendes Kommando ein und drücken Sie danach <RETURN>:

dir

Überrascht? Nicht gerade wenig Informationen, die hier über Sie hereinbrechen. Aber sehen wir uns Stück für Stück an (Bild 13). Im oberen Bereich der Liste finden Sie einige Namen mit einem nachgestellten eingeklammerten »DIR«. Dabei handelt es sich um nichts anderes als Schubladen. »DIR« ist eine Abkürzung für »Directory«, zu deutsch »Verzeichnis« – ein anderer Ausdruck für unsere (Workbench-)Schublade. Sehen Sie sich die Namen an, Sie werden einige wiedererkennen (beispielsweise »System« oder »Utilities«). Andere dagegen sind neu, da sie auf der Workbench nicht sichtbar waren. Unter der Directory-Liste finden Sie eine weitere, zwispaltige. Was alles in diese Liste aufgenommen wird, braucht uns vorerst noch nicht zu interessieren, sehen Sie sich nur kurz die Namen an, die auf »info« enden und vergleichen Sie sie mit

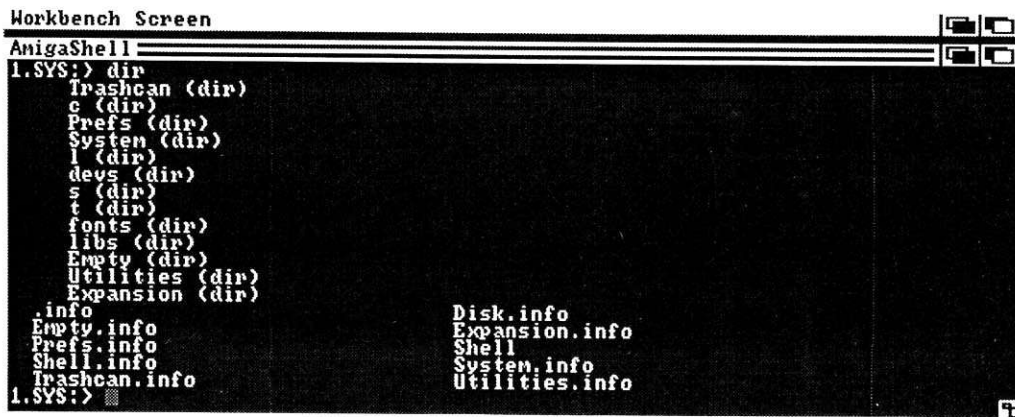


Bild 13. Der Dir-Befehl macht auf der Workbench Verborgenes sichtbar

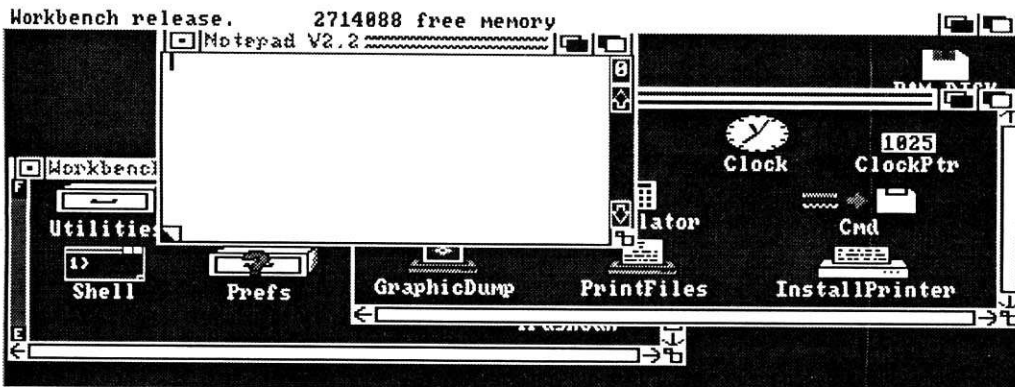


Bild 14. Der eingebaute Notizblock ist direkt nach dem Start noch leer

»Schnappschuß« aller aktivierten Icons auf Diskette. Beim nächsten Öffnen des Fensters bleiben diese an der von Ihnen festgelegten Stelle.

Bevor Sie jetzt anfangen, für jedes Icon einzeln die Position festzulegen und diese zu speichern, testen Sie folgende Methode: Halten Sie die linke »Shift«-Taste fest (die mit dem Pfeil auf der linken Seite der Tastatur zwischen <Ctrl> und <Alt>) und klicken Sie ein Icon nach dem anderen an. Sie werden feststellen, daß alle aktiviert bleiben, solange <SHIFT> gedrückt ist – versuchen Sie einmal, das gleiche ohne <SHIFT> zu erreichen.

Haben Sie sich schon einmal über den stilisierten Mülleimer gewundert? Er erfüllt genau die

drückt, während Sie das Icon zum Mülleimer bewegen. Dort lassen Sie es dann einfach »fallen«, indem Sie die linke Maustaste loslassen.

Somit haben Sie die ungeliebte Shell ins Nirwana verbannt. Falls Sie noch einige Programme auf der Workbench kennen, von denen Sie nicht wissen, wozu sie dienen, werfen Sie sie einfach in den Trashcan (denken Sie aber bitte daran, daß Sie auch weiterhin nur mit der Kopie Ihrer Workbench arbeiten).

Wußten Sie eigentlich, daß eine Shell für viele Aufgaben sehr nützlich sein kann? Bevor Sie jetzt wutentbrannt dieses Heft zerreißen, weil wir Sie dazu verleitet haben, dieses wertvolle Programm auf den Müll zu

Sie jetzt in den Trugschluß verfallen, beim Amiga könne man Sachen wegwerfen wie man will – man kommt ja immer wieder dran –, möchte ich Sie dringend warnen:

Sollte eine Diskette einmal zu voll sein (roter Balken dicht am »F«), können Sie sich mit einem einfachen Befehl Platz schaffen – »empty trash«. Aktivieren Sie dazu das Trashcan-Icon und sehen Sie sich mit gedrückter rechter Maustaste das »Special«-Menü an. Der dritte Punkt von oben ist die gesuchte Funktion. Bevor Sie diese Funktion wählen, sollten Sie sich aber stets vergewissern, daß nichts Wichtiges mehr im Trashcan ist, denn was Sie dieser »Müllabfuhr« übergeben, ist endgültig verschwunden.

dem, was Sie auf der Workbench sehen. Haben Sie einige wiedererkannt? Was auf »info« endet, ist also auf der Workbench sichtbar. Dieser Schluß ist ebenso naheliegend wie richtig.

Bis jetzt haben wir uns auf eine einzige Schublade beschränkt. Von Ihren Erfahrungen mit der Workbench wissen Sie aber, daß Sie auch in »fremde« Schubladen schauen können. Dazu gibt es ein eigenes Kommando: »cd« (für Current Directory). Dieser Befehl alleine eingegeben zeigt Ihnen, welches Directory das aktuelle (»current«) ist. Testen Sie es:

```
cd
und abschließend: <RETURN>. Sie sehen die Anzeige »copy of WorkbenchX.XX«. cd kann aber weit mehr, geben Sie ein:
```

```
cd devs
```

Auf den ersten Blick geschieht nichts, aber wenn Sie sich jetzt mit »dir« das aktuelle Directory ansehen, werden Sie feststellen, daß sich etwas verändert hat. Ein weiteres Mal »cd« bringt uns der Erklärung ein Stück näher: »copy of WorkbenchX.XX/devs« lautet die neue Anzeige. Sie finden in devs noch weitere Unterverzeichnisse. Wenn Sie wollen, sehen Sie sich diese alle an. Mit dem Befehl

```
cd /
kommen Sie aus einem Unterverzeichnis jeweils ins nächst höhere. Mit
cd :
```

landen Sie direkt im obersten. Experimentieren Sie eine Weile mit diesen beiden Befehlen. Sie werden später noch weitere kennenlernen. Und denken Sie stets daran: Sollten Sie sich einmal völlig verfrannt haben, kommen Sie mit »endcli« wieder heraus aus dem Schlamassel.

Notizblock eingebaut

Die von Commodore ausgelieferte Workbench-Diskette ist weit mehr als nur die profane System-Diskette, die Sie vielleicht von anderen Computern kennen. Sie finden dort außer den Programmen, die das System installieren, auch viele nützliche (oder zumindest ganz nette) Hilfsprogramme. Eines davon ist das Programm Notepad (sprich »Nohtpadd«, zu deutsch Notizblock), eine klei-

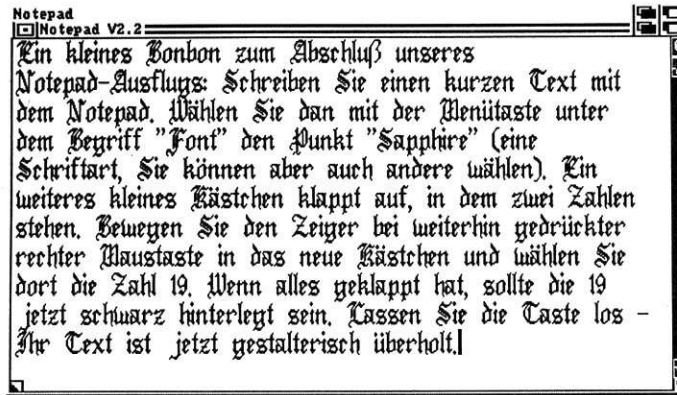


Bild 15. Der Amiga stellt Text in verschiedenen Schriftarten dar

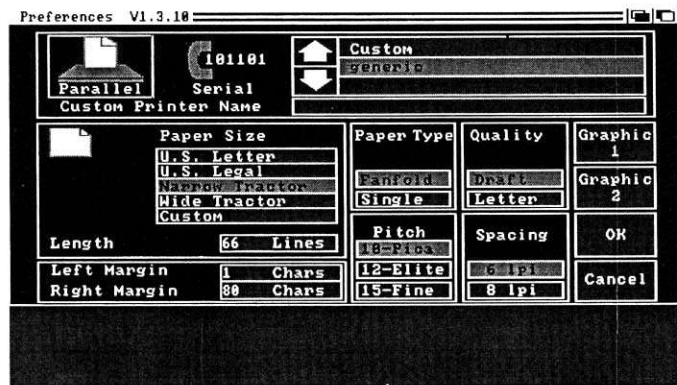


Bild 16. Mit »change Printer« stellen Sie in Preferences den Drucker Ihrer Wahl ein

ne Textverarbeitung, die auf der Workbench-Diskette in der Schublade mit dem Namen »Utilities« versteckt ist. Öffnen Sie das Fenster der Workbench und anschließend die Schublade Utilities mit einem Doppelklick. Ein neues Fenster öffnet sich, in dem zahlreiche Symbole zu finden sind, darunter auch ein kleines Notizbuch, unter dem der Name »Notepad« steht.

Öffnen Sie dessen Fenster: Zeiger auf das Notizblock-Symbol, Doppelklick – und schon erscheint die Oberfläche des Notepad auf dem Bildschirm (Bild 14). Verschieben Sie zuerst das Fenster des Notepad in die linke obere Ecke des Bildschirms und vergrößern es dann auf die volle Bildschirmgröße.

Halten Sie nun die rechte Maustaste (Menüaste) gedrückt, so erscheinen in der oberen Bildschirmzeile fünf neue Begriffe: »Project«, »Edit«, »Font«, »Style« und »Format«. So heißen die Menüs, in denen alle Funktionen zur Auswahl angeboten sind. In der linken oberen Ecke sehen Sie einen senkrechten schwarzen Strich, den Cursor, der in diesem Fall die Stelle anzeigt, an der das nächste eingegebene Zeichen

erscheinen wird. Tippen Sie jetzt über die Tastatur ein beliebiges Zeichen ein, erscheint es auf dem Bildschirm.

Möchten Sie die Farbe der Schreibfläche ändern? Kein Problem – wählen Sie im Menü »Format« den Punkt »Paper Color«. Zwischen vier Farben können Sie hier auswählen, entscheiden Sie sich jetzt einmal für die Farbe Schwarz. Stellen Sie anschließend im Format-Menü die Schriftfarbe mit dem Punkt »Pen Color« ein, wählen Sie am besten Weiß. Schreiben Sie nun einen längeren Text. Wenn Sie einen Absatz machen oder eine Leerzeile einfügen wollen, drücken Sie einfach <RETURN> – das ist die große graue Taste am rechten Rand des Buchstabenfeldes auf der Tastatur. Nachdem Sie jetzt ein paar Sätze geschrieben haben, können Sie üben, wie man den Cursor im Text bewegt: Mit der Tastenkombination <SHIFT Cursor-aufwärts> springen Sie an den Anfang des Textes, mit <SHIFT Cursor-abwärts> an dessen Ende. Die Kombinationen <SHIFT Cursor-rechts> bzw. <SHIFT Cursor-links> bewegt die Eingabemarke an das Ende bzw. den Anfang einer Zeile. Sie kön-

nen allerdings auch an jede beliebige andere Stelle im Text springen, indem Sie den Mauszeiger auf die gewünschte Position setzen und die linke Maustaste drücken.

Einen schwerwiegenden Nachteil hat das Notepad allerdings auch: Ist eine Seite voll, zeigt er die per Tastatur eingegebenen Zeichen nicht mehr an.

Sie können dann eine von neun weiteren Seiten aufschlagen, indem Sie das Eselsohr in der linken unteren Fensterecke mit der linken Maustaste anklicken. Danach ändert sich die Anzeige der Seitenzahl in der rechten oberen Ecke, hier steht nun der Wert 1. Um auf die Seite mit der Nummer 0 zurückzukommen, klicken Sie mit dem Cursor in das Feld, das die Seitenzahl anzeigt.

Das Notepad ist für kurze Notizen gedacht, die der Anwender während der Arbeit am Amiga festhalten möchte (man spricht dabei vom Speichern des Textes). Das Speichern ist ganz einfach: Wählen Sie im Menü Projekt den Punkt »Save« (zu deutsch »sichern«). Geben Sie Ihrer Notiz einen Namen, beispielsweise »Test«, und drücken die RETURN-Taste. Wählen Sie anschließend im Projekt-Menü die Funktion »New« – Ihr Text verschwindet vom Bildschirm, um einer neuen Arbeitsfläche Platz zu machen.

Vielfalt der Schriften

Laden Sie anschließend den gesicherten Text mit der Funktion »Open« (zu deutsch »öffnen«) im Menü »Projekt«, erscheint er wieder auf dem Bildschirm. Sie brauchen nur auf die Frage des Betriebssystems mit »Test« antworten.

Ein kleines Bonbon zum Abschluß unseres Notepad-Ausflugs: Schreiben Sie einen kurzen Text mit dem Notepad. Wählen Sie dann mit der Menüaste unter dem Begriff »Font« den Punkt »sapphire« (eine Schriftart, Sie können aber auch andere wählen). Ein weiteres kleines Kästchen klappt auf, in dem zwei Zahlen stehen. Bewegen Sie den Zeiger bei weiterhin gedrückter rechter Maustaste in das neue Kästchen und wählen dort die Zahl 19. Wenn alles geklappt hat, sollte die 19 jetzt schwarz hinterlegt sein. Lassen Sie die Taste los – Ihr Text ist jetzt gestalte-

risch frisch überholt: Alles erscheint jetzt in Sapphire (Bild 15). Spielen Sie auf diese Weise mit den verschiedenen Schriftarten. Die Einstellung »Topaz, 8« bringt den Ausgangszustand zurück. Sie können sogar eine Textseite in unterschiedlichen Schriftarten darstellen. Dazu wählen Sie im Menü »Format« den Punkt »Global Font«.

Halten Sie die rechte Maustaste gedrückt, bis der Zeiger auf dem Begriff steht. Klicken Sie anschließend einmal mit der linken Maustaste: Das Häkchen vor dem Begriff »Global Font« verschwindet. Häkchen bezeichnen beim Amiga häufig Funktionen, die aktiv sind.

Übrigens: den Weg aus dem Notepad heraus sollten Sie jetzt selber finden, schaffen Sie's? Einige Seiten weiter vorne können Sie noch einmal nachlesen, wie Fenster auf der Workbench geöffnet und geschlossen werden. Das Notepad reicht voll und ganz für einige kurze Notizen oder auch für Briefe oder ähnliches aus. Er bietet die wichtigsten Funktio-

Sie finden im Preferences-Fenster eine große Vielfalt an Einstellmöglichkeiten. Sehen Sie sich all diese sog. »Gadgets« (sprich: »Gädschet«) nochmals kurz an (übrigens: Gadget bedeutet zu deutsch etwa »Dingsda«, unserer Meinung nach ein deutliches Zeichen, daß auch den Commodore-Entwicklern hier kein besonders treffender Name einfiel, einigen wir uns also für unsere Zwecke ebenfalls auf »Gadget«).

Das Konzept der Preferences wurde eingeführt, weil man bei der Entwicklung eines Computers natürlich nicht voraussehen kann, welche Bedürfnisse der spätere Anwender mitbringt. Bevorzugt er auf seinem Bildschirm eher helle Grautöne oder knallige Bonbon-Farben? Welchen Drucker hat er und welche Schriften soll dieser darstellen? Wie kurz soll der Abstand für den Doppelklick sein? Die Antwort auf diese und noch jede Menge weitere Fragen wurden bewußt dem Anwender überlassen, denn nur dieser weiß, welche Einstellun-

gen am besten zu seinen Bedürfnissen passen.

Auf Seite 46 finden Sie einen Artikel, der sich mit der Bedienung der Preferences beschäftigt. Wir möchten uns an dieser Stelle daher auf die Aufforderung beschränken, alles einmal auszuprobieren, und Preferences ohne zu speichern zu verlassen (meiden Sie Felder mit dem Text »Save«). In Tabelle 2 finden Sie eine Aufstellung der wichtigsten Bedienungsfelder von Preferences.

Ihre Einstellung bitte

Haben Sie alles ausprobiert? Dann möchten wir Sie noch auf eine Kleinigkeit hinweisen, die jedoch sehr wichtig sein kann: Sicher ist Ihnen aufgefallen, daß im Drucker-Fenster (nachdem Sie auf »Change Printer« geklickt haben) rechts oben eine Liste von etwas merkwürdigen Namen enthalten ist (Sie können sie mit den beiden Pfeilen auf- und abbewegen). Wenn nicht, öffnen Sie nochmals kurz

das Preferences-Fenster und klicken Sie auf »Change Printer«. Die angezeigten Namen (Bild 16) sind eine Liste der zur Verfügung stehenden Drucker. Doch bevor Sie jetzt Ihren Schreibtisch nach den vier oder fünf in der Liste aufgeführten Geräten absuchen, bedenken Sie, daß in dieser Liste stets nur für die Workbench oder andere Programme bestimmte Beschreibungen (sog. Treiber) für die entsprechenden Drucker enthalten sind.

Sie brauchen nicht zu verzweifeln, wenn keiner der hier genannten Namen zu Ihrem Drucker paßt. Es erfordert nur einige Zeilen Tipparbeit in der Shell, um Ihren Drucker mit in die Liste aufzunehmen:

Starten Sie die Shell (siehe oben) und legen Sie die Workbench in das interne Laufwerk (»df0:«). Sie erinnern sich noch an den Befehl zum Ansehen eines Directory? Es geht dabei um »cd« und »dir«. Wechseln Sie in das Verzeichnis »devs« Ihrer Workbench-Diskette und schauen Sie es mit dem Dir-Befehl an. Sie finden zwei neue Unterverzeichnisse (»keymaps« und »printers«). Wenn Sie in »printers« die Drucker zur Einstellung in Preferences vermuten, liegen Sie genau richtig. Wechseln Sie dort hin und sehen Sie sich mit dem Dir-Befehl das Verzeichnis an. Sie finden dieselben Namen wie in Preferences. Ist Ihr Drucker dabei? Nehmen wir einfach an, Sie hätten einen ImagewriterII von Apple. Dieser taucht nicht in der Liste Ihrer Workbench auf. Sie müssen ihn erst dorthin kopieren.

Möglicherweise ist Ihre Workbench bereits bis zum Rand gefüllt. Sie können das feststellen, indem Sie den Befehl

info eingeben und mit <RETURN> abschließen. Beobachten Sie die Anzeige auf dem Bildschirm. Sie dürfte etwa so aussehen wie in Bild 17. Die Bedeutung der einzelnen Spalten entnehmen Sie bitte Tabelle 3. Uns interessiert vorerst nur die Spalte »Full«. Diese zeigt an, wieviel Platz auf den eingelegten Disketten belegt ist. So können Sie jederzeit feststellen, ob die eingelegten Disketten noch Platz für Ihre Daten haben oder nicht.

Um sicherzugehen, daß auf Ihrer Workbench Platz für den »ImagewriterII« ist, geben Sie folgende Befehle ein:

```
cd :devs/printers
delete #?
```

Workbench Screen

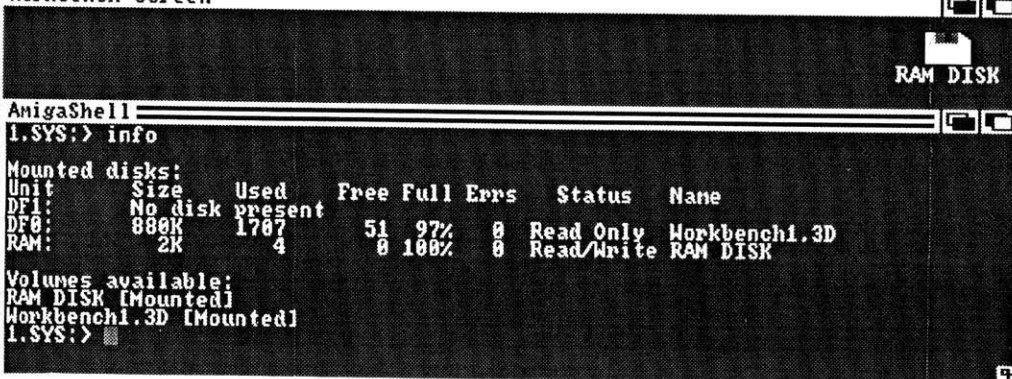


Bild 17. Der Info-Befehl zeigt an, ob Sie noch Platz auf den Disketten haben

nen einer Textverarbeitung, und er hat einen Vorteil, den Sie anfangs nicht unterschätzen sollten: Er kostet Sie keinen Pfennig.

Bevor Sie weiterlesen, sollten Sie wieder etwas mit Notepad und den neuen Shell-Befehlen experimentieren.

Nach dem elektronischen Notizblock lernen Sie die Voreinstellungen des Amiga kennen. Sie legen mit diesen Voreinstellungen fest, wie der Mauszeiger aussieht, welche Farben Ihre Workbench schmücken, welchen Drucker Sie anschließen und vieles mehr. In der Schublade »Prefs« auf der Workbench finden Sie das Programm »Preferences«, das Sie wie gewohnt mit einem Doppelklick aufrufen.

Überblick: Preferences

Begriff	Bedeutung
Reset All	Einstellungen der Standard-Workbench
Last Saved	zuletzt festgehaltene Einstellung
Change Serial	Parameter der seriellen Datenübertragung
Text	60 oder 80 Spalten
Reset Colors	Farbänderung verworfen
RGB	Farben der Workbench einstellen
Key Repeat Delay	Verzögerung, bis Tastenwiederholung aktiv
Key Repeat Speed	Geschwindigkeit der Tastenwiederholung
WB Interlace	Interlace (Flimmern) der Workbench ein/aus
Maussymbol links	Bewegungsverhältnis Maus zum Mauszeiger
Maussymbol rechts	akzeptierte Zeit zwischen Doppelklicks
Change Printer	Parameter für Drucker einstellen
Edit Pointer	Mauszeiger ändern
Save	Einstellungen festhalten
Use	Einstellungen benutzen, aber nicht sichern
Cancel	Verlassen der Preferences ohne Änderungen

Tabelle 2. Die wichtigsten Bedienungselemente von Preferences im Überblick

Der erste Befehl bringt Sie in das gewünschte Verzeichnis, der zweite (»delete«) löscht alle dort vorhandenen Druckertreiber. Normalerweise übergeben Sie an delete den Namen, den Sie löschen wollen. Wenn Sie aber genau wissen, daß Sie im aktuellen Verzeichnis alles löschen möchten (und bitte nur dann), verwenden Sie oben genannte Kombination aus »Gartenzaun« und Fragezeichen. Diese gilt als sog. Jokerzeichen. Damit legen Sie fest, daß alle Namen gemeint sind, die einem bestimmten Muster entsprechen. Das Muster war in unserem Fall »#?«, das bedeutet, es wurde kein Muster vorgegeben, also alle Namen. Sie könnten beispielsweise auch eine Vorgabe machen etwa in der Form: »epson #?«. Gemeint sind dann alle Namen, die mit »epson« beginnen. Oder »#?.info«. Hier wären alle Namen gemeint, die auf ».info« enden.

Sie haben mit den beiden Befehlen genügend Platz auf Ihrer Diskette geschaffen. Schauen Sie sicherheitshalber noch einmal mit »info« nach. Legen Sie jetzt die mitgelieferte »Extras«-Diskette ins Laufwerk. Diese enthält eine Menge weiterer Druckertreiber für fast alle gängigen Druckertypen. Sie finden sie an derselben Stelle wie auf der Workbench: Im Ordner »devs/printers«. Um dort hin zu gelangen, sollten Sie zunächst den genauen Namen der Extras-Diskette kennen (vermutlich heißt sie »Extras1.3D«). Sie erhalten diese Information über die Workbench. Der Name der Diskette steht unter ihrem Icon. Schieben Sie notfalls das Fenster der Shell zur Seite, um dieses Icon zu sehen.

Falls Sie nur ein Laufwerk besitzen, wird alles weitere mit einigen Diskettenwechseln verbunden sein. Besitzen Sie dagegen ein zweites Laufwerk, so lassen Sie die Workbench im internen und legen die Extras-Diskette in das zweite Laufwerk. Danach geben Sie in jedem Fall folgende Zeile ein:

```
cd Extras1.3D:
```

Sie werden in der linken oberen Ecke eine Aufforderung entdecken, die Diskette mit dem Namen »Extras1.3D« einzulegen. Vielleicht wundern Sie sich über den Doppelpunkt hinter dem Diskettenamen? Ohne diesen Doppelpunkt würde das Betriebssystem im aktuellen Verzeichnis nach einem Unterverzeichnis dieses Namens suchen. Genaugenommen han-

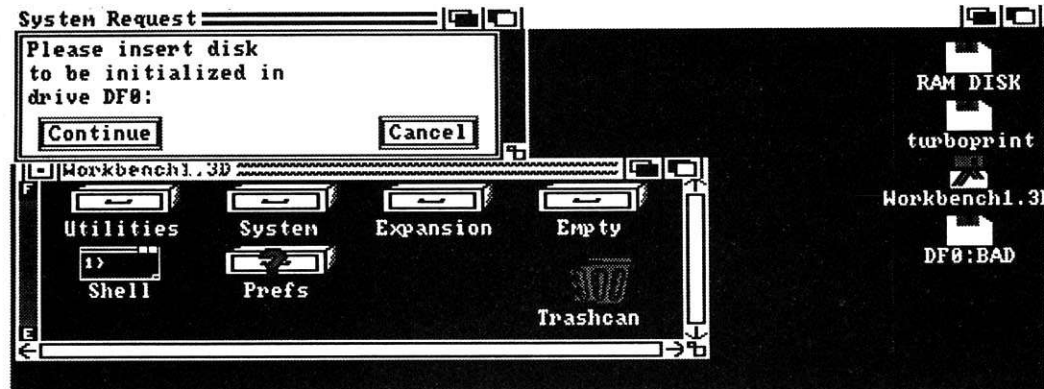


Bild 18. Beim Formatieren werden Sie mehrmals gewarnt, damit Sie nichts ungewollt löschen

delt es sich dabei nicht um den Namen der Diskette, sondern eines sog. »Volume«. Wo der Unterschied liegt werden Sie etwas weiter unten erfahren. Vorerst genügt es, zu wissen, daß mit jeder Diskette ein Name verbunden ist, über den sie benannt werden kann.

Sie befinden sich im Stammverzeichnis der Extras-Diskette. Sie erkennen dies entweder am sog. »Prompt« (der Text vor dem »größer-als«-Zeichen am Anfang der Eingabezeile) oder an der Meldung auf den »cd«-Befehl.

Jede Menge Drucker

Die Extras-Diskette hat ebenso wie die Workbench einen »devs«-Ordner und darin die »printers«. Wechseln Sie mit Hilfe von »cd« dorthin – falls Sie es nicht mehr genau wissen: Etwas weiter oben finden Sie die Beschreibung. Sehen Sie sich mit »dir« das Verzeichnis an. Sie werden eine lange Liste von Druckernamen finden, darunter auch »ImagewriterII«. Aber wie kommt dieser Imagewriter jetzt auf Ihre Workbench-Diskette? Indem Sie ihn kopieren:

```
copy ImagewriterII
devs: printers
```

Natürlich können Sie für »ImagewriterII« auch den Namen Ihres Druckers verwenden. Falls Ihr Drucker nicht ausdrücklich aufgeführt ist, eignet sich meist einer der »Epson«-Treiber – fast alle Drucker neueren Datums sind sog. »Epson-kompatibel« (sie sind stark vereinfacht gesagt annähernd baugleich). Sollten Sie nur ein Laufwerk besitzen, so werden Sie in diesem Moment erfahren, wie nervenaufreibend die Arbeit eines Diskjockeys sein kann. Sie werden immer wieder aufgefordert, die Diskette zu wechseln.

Leisten Sie jeder Aufforderung geduldig Folge, so dürfen Sie nach nicht mehr als 12 bis 15 Diskettenwechseln den gewünschten Druckertreiber auf Ihrer Workbench haben. Einfacher haben es da schon die Besitzer eines zweiten Laufwerks: Sie legen einfach die Extras-Diskette in das zweite und sparen sich das Wechseln.

Eine Kleinigkeit haben wir Ihnen bisher unterschlagen. Es handelt sich um den Ordner »devs:«. Da wurde soeben noch erklärt, mit Doppelpunkt seien Disketten (oder genauer »Volumes«) gekennzeichnet, und schon beim nächsten Befehl ist doch wieder ein normales Verzeichnis gemeint. Ich kann Sie beruhigen. »devs:« ist nichts weiter als ein »gedachtes« (oder in der Fachsprache: »logisches«) Volume. Diese werden vom Amiga beim Start automatisch auf der Startdiskette eingerichtet. So ist »devs:« immer das Devs-Verzeichnis auf der Startdiskette, »libs:« der Libs-Ordner der Startdiskette usw.

Auch bei »copy« können Sie die Jokerzeichen verwenden. So kopiert »copy epson #? devs:printers« alle Treiber auf Ihre Workbench, deren Name mit »epson« beginnt.

Beenden Sie jetzt die Arbeit mit dem CLI. Starten Sie noch-

mals Preferences und sehen Sie sich die Liste der Druckertreiber erneut an. Sie finden jetzt nur noch den ImagewriterII in der Liste (bzw. den Drucker, den Sie statt dessen eingesetzt haben). Beenden Sie Preferences wieder und experimentieren Sie noch etwas mit dem Copy-Befehl. Am besten kopieren Sie einiges in das Verzeichnis »RAM:«. Dort brauchen Sie sich nicht die Mühe zu machen, alles wieder zu löschen, denn dieses Verzeichnis wird beim nächsten Start Ihres Amiga automatisch entleert.

Wozu benötigt man Disketten überhaupt? Vereinfacht erklärt: Ein Computer wie der Amiga ist eine Maschine, die Informationen erhält, diese bearbeitet und wieder ausgibt. Daten gibt man beispielsweise über die Tastatur ein. Die Informationen werden dann im Hauptspeicher des Amiga abgelegt. Nach dem Ausschalten sind aber alle Informationen, die im Hauptspeicher parat waren, endgültig gelöscht.

Stellen Sie sich vor, Sie haben im Notepad sieben Seiten Text geschrieben, und nach dem Ausschalten wäre der gesamte Text verloren. Die Lösung für dieses grundsätzliche Problem bieten die sog. Massenspeicher, beispielsweise Dis-

Anzeige des Info-Befehls

Spalte	Bedeutung
Unit	steht für die vorhandenen Laufwerke
Size	gibt den zur Verfügung stehenden Speicherplatz in jedem Laufwerk an
Used	der belegte Speicherplatz in Blöcken
Free	der freie Speicherplatz in Blöcken
Full	der belegte Speicher in Prozent
Errs	die aufgetretenen Fehler
Status	gibt an, ob eine Diskette schreibgeschützt ist
Name	der Name der jeweiligen Diskette, wie er auch unter dem Disketten-Icon erscheint

Tabelle 3. Die einzelnen Spalten des Info-Befehls aufgeschlüsselt

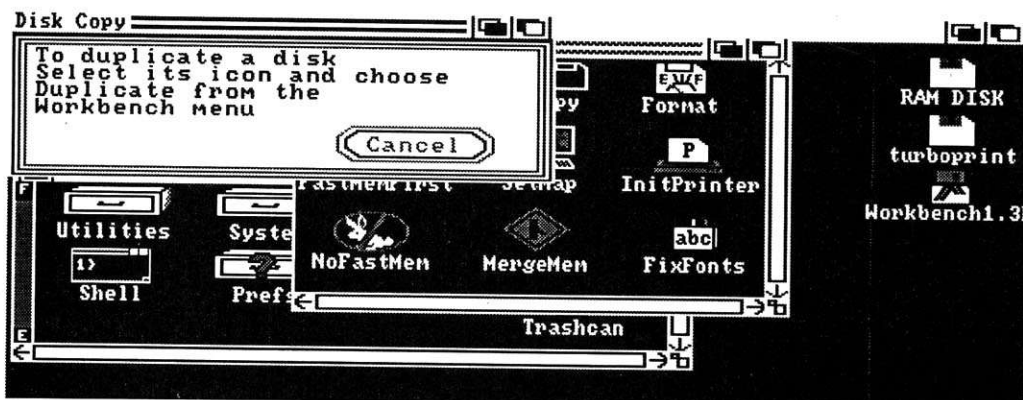


Bild 19. »Diskcopy« können Sie nicht direkt starten – es weigert sich mit einem kurzen Hinweis

ketten. Nach der Arbeit am Computer hält der Anwender auf einer Diskette die eingegebenen Daten fest, er speichert die Informationen. Auch Programme wie das Notepad oder die Preferences sind auf einer Diskette gespeichert. Wenn Sie

Wozu Massenspeicher?

später den Amiga wieder einschalten, können Programme und Daten wieder von der Diskette in den Hauptspeicher eingelesen werden: Sie werden geladen. Immer wenn die Anzeige des Disketten-Laufwerks aufleuchtet, werden Informationen von Diskette gelesen oder auf Diskette geschrieben. Achtung: Nehmen Sie niemals die Diskette heraus, wenn die Anzeige leuchtet! Gespeicherte Informationen, also Programme und Daten, können dabei unwiederbringlich verlorengehen. Wenn Sie mit einer neuen Diskette arbeiten wollen, müssen Sie diese allerdings erst einmal für die Arbeit mit dem Amiga vorbereiten.

Die Struktur der magnetischen Oberfläche einer Diskette muß zuerst auf das richtige Format getrimmt werden – im Computer-Fachdeutsch »formatieren« Sie Ihre Diskette. Bei jedem Formatieren löschen Sie allerdings auch alle Informationen, die vorher auf der Diskette standen. Haben Sie eine neue Diskette bereit? Dann legen Sie sie in das Laufwerk, klicken das erscheinende Diskettensymbol einmal an und wählen »Initialize« aus dem Workbench-Menü »Disk«.

Folgen Sie den Anweisungen auf dem Bildschirm, legen Sie erneut die Workbench-Diskette (der Amiga muß nämlich die für das Kopieren nötigen Informationen von der Workbench-Diskette laden) und anschließend

wieder die zu formatierende Diskette in das Laufwerk. Klicken Sie dann das Kästchen »Continue« an (Bild 18), vergewissern sich, daß die richtige Diskette im Laufwerk liegt. Klicken Sie erneut »Continue« an, das Formatieren beginnt. Im Kasten links oben erscheint die Information, welche Spur der Diskette formatiert wird. Insgesamt hat jede Diskette nach der Formatierung im Standard-Format 80 »Spuren« (40 konzentrische Kreise auf jeder Diskettenseite). Ist die Formatierung beendet, verschwindet das Kästchen, und das Symbol der neu formatierten Diskette trägt den Namen »Empty« (leer). Sie können Ihre Daten auf diese Diskette speichern.

Mit dem Befehl »Rename« aus dem »Workbench«-Menü benennen Sie die Diskette um: Klicken Sie das Symbol »Empty« einmal an, wählen »Rename« und geben über die Tastatur den neuen Namen in den inzwischen erschienenen Balken neu ein. Sie können dabei Groß- und Kleinschreibung mischen und sogar Leerzeichen (Space) verwenden. Ein Trick, der die Eingabe des neuen Namens beschleunigt: Halten Sie, wenn der Balken mit dem Namen »Empty« auftaucht, die rechte Amiga-Taste gedrückt und tippen auf »X«; das Wort wird mit einem Schlag gelöscht. Das funktioniert fast immer, wenn Texte in ähnlichen Feldern bearbeitet werden.

Alle Menüs brandheiß

Damit kennen Sie bereits die wichtigsten Menüs der Workbench. Lassen Sie uns – der Vollständigkeit halber – auch die anderen noch betrachten.

Aus dem Workbench-Menü kennen Sie bereits die Unterpunkte »Open«, »Close«, »Re-

name« und »Info«. Klicken Sie einmal auf einen freien Bereich der Workbench-Oberfläche (weder in ein Fenster noch auf ein Icon). Sehen Sie sich nochmals das »Workbench«-Menü an: Alle Punkte erscheinen in einer reichlich merkwürdigen Schrift. Menüs nehmen immer dann diese Schrift an, wenn Sie nicht wählbar sind. Das Betriebssystem schaltet also bestimmte Menüs zu bestimmten Zeiten ab. Es würde beispielsweise keinen Sinn machen, den Befehl »Open« zu geben, wenn niemand weiß, was geöffnet werden soll. Dasselbe gilt selbstverständlich für »Close«, »Rename« und die anderen in gleicher Weise. Auch in den beiden anderen Menüs werden Sie diese verschleierte Menüs finden. Die einzigen im Moment wählbaren sind »Redraw« und »Snapshot« aus dem »Special«-Menü.

Klicken Sie beispielsweise einmal auf das Shell-Icon und sehen Sie sich das Workbench-Menü nochmals an: Fast alle Funktionen sind aktiviert. Sie können die Shell per Menü öffnen, kopieren, umbenennen, sich weitere Informationen besorgen oder sie löschen (»Discard«). Auch können Sie jetzt mit »Snapshot« Ihre Position im Fenster festlegen.

Wenn Sie statt der Shell den Trashcan aktivieren, steht eine zusätzliche Funktion im »Disk«-Menü zur Verfügung. »Empty Trash«, die »Müllabfuhr« – Sie erinnern sich? Dagegen ist »initialize« nur wählbar, wenn vorher ein Disketten-Icon aktiviert wurde. Es wäre auch nicht besonders sinnvoll, beispielsweise Unterverzeichnisse einer Diskette zu formatieren.

Das einzige Menü, das Sie noch nicht kennengelernt haben (und das bei Ihrer Workbench vielleicht noch nie wählbar war), ist »Last Error«. Um in den Genuß dieses Menüs zu

kommen, müssen wir zunächst einen Fehler provozieren: Aktivieren Sie das Shell-Icon und wählen Sie das Info-Menü. Sie sehen in der unteren Hälfte des Fensters drei Textzeilen mit den Namen »Comment«, »Default Tool« und »Tool Types«. Die einzige Zeile mit einem Eintrag ist »Default Tool« – in dieser Zeile steht »System/CLI«. Das bedeutet, daß nach einem Doppelklick auf dieses Icon das Programm aus dem System-Ordner namens »CLI« gestartet werden soll. Wenn Sie diesen Namen verändern, findet das Betriebssystem das gewünschte Programm nicht mehr. Klicken Sie dazu hinter den erwähnten Text und löschen Sie ein Zeichen (das »l« von CLI). Speichern Sie unseren kleinen Unfug mit »Save« (links unten) und versuchen Sie nochmals, die Shell aufzurufen. Was geschieht? Sie erhalten einen kurzen Bildschirmblitz und in der Titelleiste der Workbench die Meldung »Error while opening :System/CL: 205« – was die 205 bedeutet, müssen Sie jetzt nicht unbedingt wissen, Sie sollten aber bei Gelegenheit (spätestens wenn der nächste gleichartige Fehler auftritt) in Ihrem Handbuch diese Fehlernummern nachschlagen. Nach dem nächsten Klick auf ein beliebiges Fenster oder auf die Workbench ist die Fehlermeldung wieder verschwunden.

Amiga mit System

Sehen Sie sich jetzt das »Special«-Menü nochmals an. Der Unterpunkt »Last Error« ist jetzt aktiviert. Wählen Sie ihn an, so erscheint wieder unsere Fehlermeldung in der Titelleiste.

Beseitigen Sie den Grund des Ärgernisses, indem Sie mit dem Info-Menü wieder den korrekten Namen eingeben. Sehen Sie sich dann mit »Version« (ebenfalls im Special-Menü) die Versionsnummer Ihrer Workbench an und rufen Sie danach nochmals »Last Error« auf. Überrascht? Auch der Amiga hat eben – wie fast jeder Computer – noch kleine Macken.

Haben Sie sich den System-Ordner einmal näher angesehen? Sie finden dort neun verschiedene Programme:

CLI ist das gleiche Programm, das auch für die Shell benutzt wird – mit einem kleinen Unterschied: Starten Sie die Shell, geben Sie einige Befehle ein und drücken Sie dann

die Pfeil-nach-oben-Taste auf der rechten Seite Ihrer Tastatur (unter »Del« und »Help«). Versuchen Sie das gleiche, wenn Sie CLI statt Shell gestartet haben. Das war der wichtige Unterschied.

Diskcopy ist verantwortlich für das Kopieren von Disketten. Wenn Sie das Programm per Doppelklick starten, werden Sie auf die korrekte Anwendung von »Diskcopy« hingewiesen (Bild 19): mit »Duplicate« aus dem Workbench-Menü. Eine einfachere Alternative ist die oben beschriebene Möglichkeit, ein Disketten-Icon direkt auf ein anderes zu legen. Der Amiga vermutet auch dann, daß Sie die zuerst angeklickte Diskette auf die zweite kopieren wollen.

Format ist zuständig für das Formatieren von Disketten. Dieses Programm wird aufgerufen, wenn Sie »Initialize« aus dem

Starten Sie darauf Setmap per Doppelklick. Hinterher steht Ihnen die deutsche Tastaturtabelle mit allen Umlauten und »ß« zur Verfügung.

Initprinter überträgt die in Preferences eingestellten Druckereigenschaften an einen angeschlossenen Drucker. An-

reich zusammenarbeiten.

MergeMem fügt einzelne Speicherbereiche zu größeren Stücken zusammen. Einige Programme versuchen, ihren benötigten Speicherplatz am Stück zu belegen. Wenn kein entsprechend großes Stück mehr vorhanden ist, kann das

chen Programmen finden Sie im Utilities-Ordner (Bild 22). Hier sind meist Programme abgelegt, die Ihnen ein gewohntes Hilfsmittel ersetzen können (beispielsweise das Notepad für Bleistift und Papier oder »Clock« für die Uhr).

Probieren Sie aus, was geschieht, wenn Sie diese Programme starten. Sehen Sie sich jedes davon genau an – es lohnt sich. Wir werden hier nicht groß darauf eingehen, da sie für das Verständnis des Amiga eher nebensächlich sind.

Shell zum letzten

Wenn Sie von der Shell schon genügend entnervt waren – keine Panik, es kommt nicht mehr viel. Wir werden Ihnen lediglich noch ein paar Tips geben, wie Sie sich bei der Arbeit mit diesem Werkzeug das Leben vereinfachen.

Zunächst sollten Sie wissen, welche Befehle Ihnen bei der Arbeit mit der Shell zur Verfügung stehen. Wechseln Sie in den C-Ordner Ihrer Workbench. Benutzen Sie dazu folgende Befehle: Entweder

```
cd SYS:c
```

oder

```
cd C:
```

Beide Befehle haben den gleichen Effekt. Das Volume »SYS:« (Sie erinnern sich noch an den Unterschied zwischen Volume und Diskette? – Wenn nicht, finden Sie auf Seite 40 die Erklärung) steht immer für die Diskette, von der Sie gebootet haben (also vermutlich Ihre Workbench). »C:« dagegen steht immer für den C-Ordner auf Ihrer Boot-Diskette.

Sehen Sie sich diesen Ordner einmal genauer an. Mit dem Dir-Befehl erhalten Sie eine bildschirmfüllende Anzeige.

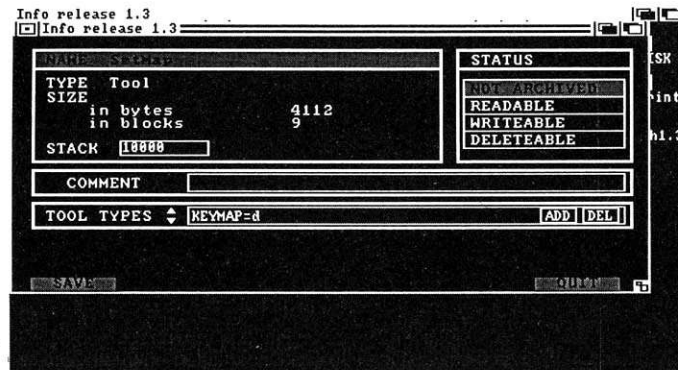


Bild 21. »Setmap« ist etwas kompliziert zu handhaben. Am besten binden Sie es in die »Startup-Sequence« ein.

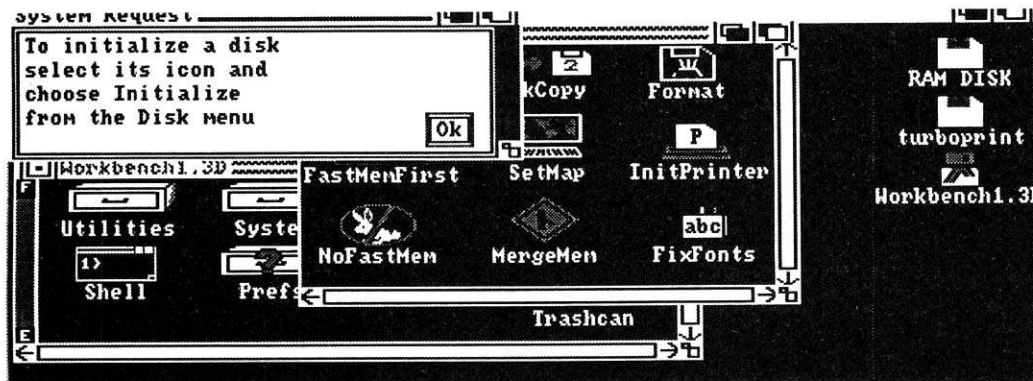


Bild 20. Das Formatierprogramm starten Sie über »Initialize« von der Workbench

Disk-Menü aufrufen. Wenn Sie es direkt starten, werden Sie darauf aufmerksam gemacht, daß dies nicht geht (Bild 20).

FastMemFirst führt einige Manipulationen in der Anordnung des Speichers durch. Der Amiga verwendet verschiedene Speicherbereiche: Das sog. »Chip«-Memory ist der Bereich, auf den jeder Baustein (»Chip«) des Amiga zugreifen kann. Dagegen darf im »Fast«-Memory nur der Prozessor (sozusagen der »Chef« des Ganzen) Daten ablegen und lesen. Dadurch ist der Zugriff auf diesen Speicherbereich für den Prozessor etwas schneller (er wird nicht von seinen »Konkurrenten« gestört).

Setmap meldet eine neue Tastatur-Tabelle an. So können Sie entweder den deutschen oder einen beliebigen anderen Zeichensatz anmelden, der auf Ihrer Workbench enthalten ist. Dazu tragen Sie im Info (Bild 21) die Zeichenfolge »KEYMAP=d« in der Zeile »Tool Types« ein.

sonsten würde dieser nicht zwangsläufig erkennen, daß Sie beispielsweise im »NLQ«-Modus (NLQ = Near Letter Quality) drucken möchten.

NoFastMem schaltet den Bereich des Fast-Memory (siehe oben) ab. Dies kann gelegentlich nötig sein, da manche (vor allem ältere) Programme nicht mit diesem Speicherbe-

Programm nicht gestartet werden, obwohl der Speicherplatz ausreichen würde. In diesem Fall hilft MergeMem.

FixFonts aktualisiert die Liste der zur Verfügung stehenden Schriftgrößen (»Fonts«). Programme, die unterschiedliche Fonts nutzen, brauchen diese Information. Eine ähnliche Ansammlung von nützlich-

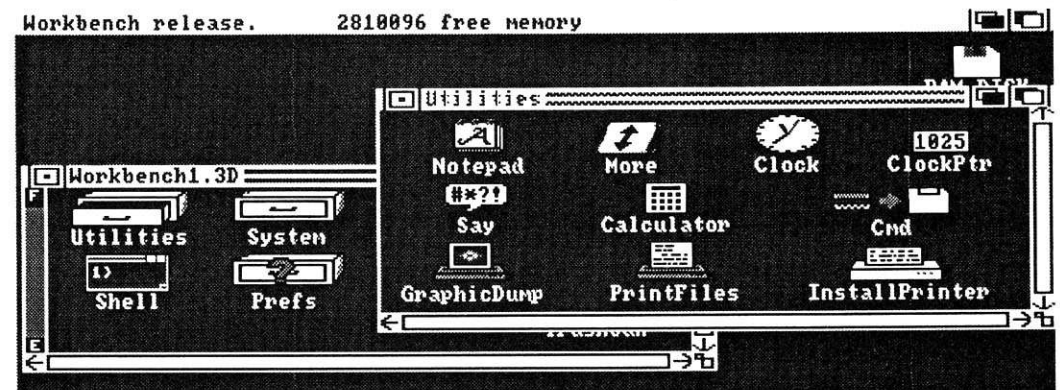


Bild 22. Der Utilities-Ordner ist randvoll mit nützlichen Programmen

Sie können mit der Space-Taste (die langgezogene Taste am unteren Rand der Tastatur) die Anzeige stoppen und mit Backspace (die graue Taste mit dem Pfeil nach links über <RETURN>) weiterlaufen lassen.

Haben Sie einige Namen in diesem Verzeichnis wiedererkannt? Zumindest »newcli«, »endcli«, »copy«, »dir« und »CD« sollten Sie kennen. Die Erklärung ist einfach: Im C-Ordner sind alle Befehle enthalten, die Sie in der Shell verwenden können. Jeder Name steht für ein kleines Programm, das von der Shell mit der Eingabe des Namens gestartet werden kann. Wir verzichten an dieser Stelle auf eine Erklärung der einzelnen Befehle – wir würden ein ganzes Heft dieses Umfangs damit füllen. Sie finden in Ihrem Amiga-DOS-Handbuch eine Aufstellung der einzelnen Befehle und ihrer Wirkungsweise.

Es ist allerdings wichtig zu wissen, daß Sie diese Befehle auch automatisch ausführen lassen können, ohne sie jedesmal neu einzugeben. Dazu legen Sie einfach fest, in welcher Reihenfolge die Befehle abgearbeitet werden und übertragen diese Aufgabe dem Betriebssystem.

Angenommen Sie möchten die vorhin beschriebenen Anweisungen zum Kopieren eines Druckertreibers auf diese Weise automatisieren. Zugegeben dieses Unterfangen ist reichlich sinnlos, da es meist nur einmal erledigt wird. Aber lassen Sie es uns zur Übung ruhig einmal probieren. Dazu schreiben Sie alle benötigten Anweisungen untereinander und speichern das Ergebnis mit einem Namen versehen auf Diskette. Das war etwas viel auf einmal, denn Sie brauchen dazu ein Werkzeug, das Sie noch nicht kennen – einen sog. Editor.

Dieser speichert die ihm anvertrauten Texte unter dem gleichzeitig mit angegebenen Namen auf der Diskette. Wir verwenden anfangs den mitgelieferten Editor aus dem C-Ordner. Haben Sie ihn schon gefunden? Sein Name ist »Ed«. Rufen Sie ihn probeweise einmal auf. Welches Ergebnis erhalten Sie? »Bad arguments« lautet der Kommentar. Ed möchte gleich wissen, woran er ist – sprich welchen Text er bearbeiten soll. Rufen Sie ihn also auf mit

Ed Kopiere_Drucker

Sie sehen ein neues Fenster mit einem roten Text in der Fuß-

zeile: »Creating new file« (Bild 23). Mit File ist nichts anderes gemeint als der Text, den Sie mit diesem Editor schreiben. »Creating« mag freilich etwas übertrieben klingen, denn kreativ müssen Sie schon selbst werden. Wenn Sie nichts schreiben, kann Ed auch nichts für Sie tun.

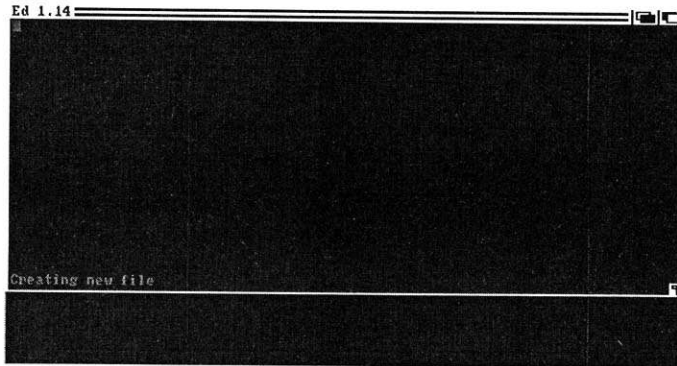


Bild 23. Mit dem »Ed« speichern Sie Texte auf Diskette und bearbeiten bereits bestehende

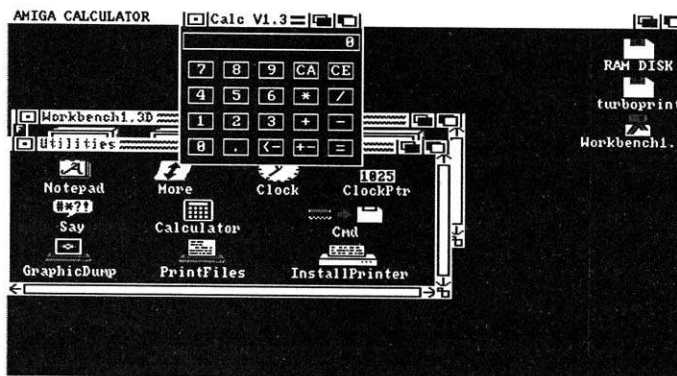


Bild 24. Eingebauter Taschenrechner: der »Calculator«

Übergeben Sie Ed also alle vorhin gebrauchten Zeilen in der gleichen Reihenfolge:

```
cd SYS:devs/printers
delete #?
cd Extras1.3D:devs/
  printers
copy ImagewriterII devs:
  printers
```

Speichern Sie das eben Geschriebene, indem Sie die Esc-Taste drücken (darauf erscheint ein kleiner Stern am unteren Fensterrand), geben »x« ein und drücken <RETURN>. Ihr Text wird damit auf Diskette verwahrt.

Sie können diese sog. »Batch«- oder »Script«-Datei (so nennt man eine Aneinanderreihung verschiedener Befehle, die dann selbständig abgearbeitet werden) ausführen lassen, indem Sie die Zeile

Execute Kopiere_Drucker

eingeben. Darauf wird ein Befehl nach dem anderen ausge-

führt, so wie Sie ihn eingegeben haben. Falls Sie den Wunsch verspüren, den Ablauf zu ändern (um mehrere Druckertreiber gleichzeitig zu kopieren), brauchen Sie nur mit ed Kopiere_Drucker

die Datei erneut bearbeiten. Aber die Shell hält einige an-

```
cd :trashcan
dl #?
```

Es geschieht das gleiche wie wenn Sie »delete« verwendet hätten. Mit »alias« können Sie sich also Synonyme definieren, die später alternativ zum ursprünglichen Befehl verwendet werden. Für die häufige Arbeit mit der Shell ist dies sicherlich notwendig.

Grundsätzlich gibt es zwei Möglichkeiten, sich mit allen Fähigkeiten des CLI vertraut zu machen: Sie können alle Befehle ausprobieren und sich so mit der Wirkungsweise beschäftigen oder Sie können das Handbuch von Anfang bis Ende durcharbeiten. Wir empfehlen eine Mischform: Probieren Sie einfach drauflos und legen Sie das Handbuch daneben. So können Sie jederzeit nachschlagen, falls Sie nicht mehr weiterwissen, ersparen sich aber das wiederholte Lesen von längst Bekanntem oder zumindest längst Vermutetem.

Beenden Sie jetzt die Shell (für diesen Kurs zum letzten Mal – ich höre einige unter Ihnen aufatmen?), denn wir wollen uns noch einige Programme betrachten, die auf der Extras-Diskette mitgeliefert werden.

Legen Sie »Extras1.3D« in ein Laufwerk und öffnen Sie das Disketten-Icon per Doppelklick. Sie sehen ein Fenster mit vier Schubladen und fünf anderen Symbolen. Starten Sie zunächst das Programm »Calculator« mit einem Doppelklick. Am oberen Rand des Bildschirms öffnet sich ein neues Fenster (Bild 24), das beinahe wie ein kleiner Taschenrechner aussieht. Genau darum handelt es sich tatsächlich. Sie bedienen diesen Taschenrechner entweder mit der Amiga-Tastatur oder mit der Maus (per Klick in die einzelnen Felder). Klicken Sie diese Spielerei wieder weg (in der linken oberen Ecke des Fensters). Weit hilfreicher ist das schon ein Programm namens »Terminal«. Es handelt sich dabei um ein Programm, mit dem Sie über die Telefonleitung Daten empfangen oder verschicken können (vorausgesetzt freilich, Sie haben ein sog. »Modem« sowohl mit dem Telefon als auch mit Ihrem Computer verbunden, und am anderen Ende der Leitung sind dieselben Voraussetzungen erfüllt). Sicher gehört dieses Programm nicht gerade zu den komfortabelsten auf dem Markt, aber dafür ist es ja auch umsonst.

Eigene Abkürzungen

Was geschieht? Die letzte eingegebene Zeile erscheint wieder, kann verändert und nochmals abgeschickt werden – sicherlich eine nette Aufmerksamkeit.

Ähnlich verhält es sich mit Abkürzungen. Geben Sie einmal folgende Zeile ein:

```
alias dl delete
```

Auf den ersten Blick passiert überhaupt nichts. Aber geben Sie jetzt ein:

Richtig interessant wird es jedoch in der Schublade »Tools«. So ist hier ein Programm enthalten (»Palette« – Bild 25), mit dem Sie die Workbench-Farben einfach und schnell verstellen können. Wenn Sie mit Preferences alles Nötige eingestellt haben und nur noch gelegentlich die Farben geändert haben möchten, können Sie Preferences löschen (die gespeicherten Einstellungen gehen dabei nicht verloren) und statt dessen dieses Programm auf die Workbench kopieren.

»KeyToy2000« (Bild 26) zeigt die aktuelle Tastaturbelegung an. Die drei grauen Tasten (»Ctrl«, »Alt« und »Shift«) haben nur eine Wirkung, wenn sie zusammen mit einer anderen gedrückt werden. Welche Wirkung, das erfahren Sie mit einem Klick auf die entsprechende Taste, die sich daraufhin rot verfärbt und die neue Belegung anzeigt.

Klicken Sie auf das Icon von »Fed«. Es handelt sich hier um einen sog. Font-Editor (Bild 27). Fonts sind uns vorhin schon einmal begegnet. Der Amiga verarbeitet prinzipiell jeden Zeichensatz, den Sie von ihm verlangen. Wenn Sie mit den mitgelieferten nicht zufrieden sind, malen Sie sich mit dem »Fed« Ihre eigenen, die Sie auch völlig problemlos beispielsweise mit dem Notepad benutzen können. Experimentieren Sie eine Weile mit diesem Programm – es wird Sie sicherlich faszinieren.

Vermutlich das wichtigste Programm in der Tools-Schublade ist der »MicroEmacs«. Starten Sie ihn einfach per Doppelklick. Nach einem kurzen Moment verschwindet die Workbench und wird durch einen neuen – etwas seltsam aussehenden – Bildschirm ersetzt (Bild 28). Daß die Workbench jedoch nicht ganz verschwunden ist, können Sie leicht feststellen, wenn Sie gleichzeitig die linke Amiga-Taste und »n« drücken. Stellen Sie mit »linke Amiga« und »m« den Ausgangszustand wieder her. Sie können statt dessen auch den MicroEmacs in seiner Titelleiste nach unten ziehen – auch dann wird die Workbench sichtbar (siehe Bild 29).

Aber um was handelt es sich bei diesem Programm? Ein Blick auf die Menüs macht Sie vermutlich immer noch nicht viel schlauer, läßt aber immerhin vermuten, daß dieser »Emacs« einen enormen Leistungsumfang haben muß – immerhin gibt es acht Menüs, alle-

Workbench Screen

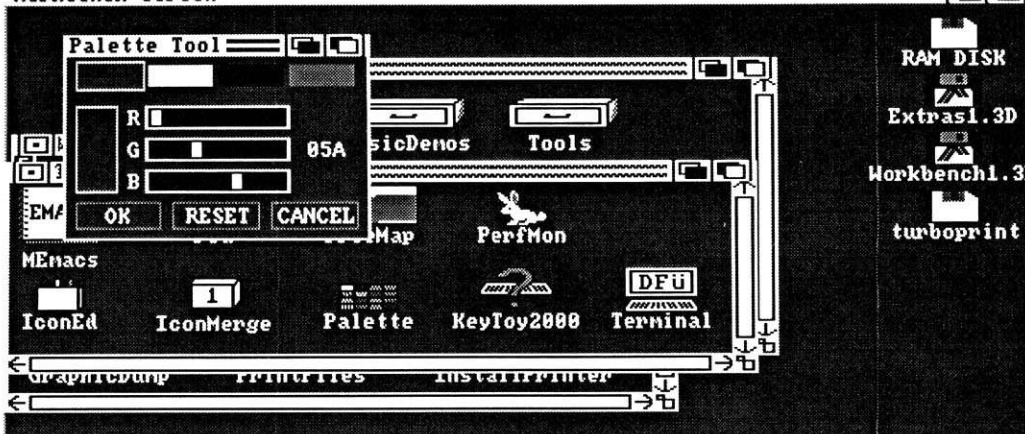


Bild 25. Mit »Palette« stellen Sie einfach und komfortabel die Workbench-Farben ein



Bild 26. Wo war doch gleich noch mal dieses oder jenes Sonderzeichen? »KeyToy2000« zeigt alle an.

samt vollgestopft mit Unteremenüs.

Emacs ist ein Editor ähnlich wie »ed«. Emacs hat jedoch einige fundamentale Unterschiede: Er ist wesentlich einfacher zu bedienen, hat einen weit größeren Funktionsumfang und braucht dafür allerdings auch wesentlich mehr Platz sowohl auf der Diskette als auch im Speicher.

Um sich mit der Arbeitsweise dieses Editors vertraut zu machen, schreiben Sie am besten einen kurzen Text. Mit der RETURN-Taste schließen Sie eine Zeile ab und springen an den

Anfang der nächsten. Sie können sich jetzt mit den Cursor-tasten (die vier kreuzweise angeordneten Tasten mit den kleinen Pfeilen rechts neben <RETURN>) durch den Text bewegen. Testen Sie die Wirkung der einzelnen Menüs. Wenn Sie irgendwo nicht mehr weiter wissen, finden Sie in Ihrem AmigaDOS-Handbuch eine Beschreibung der einzelnen Funktionen.

Selbstverständlich können Sie jeden beliebigen Text nach Belieben verändern und wieder auf Diskette sichern. Ein gutes Beispiel ist die Veränderung

Amiga font editor V1.3

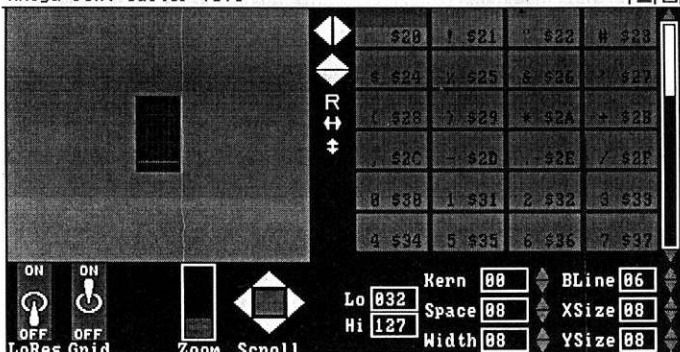


Bild 27. Der Amiga verarbeitet beliebige Fonts. Mit »Fed« entwerfen Sie Ihre eigenen Schriftarten.

der sog. »Startup-Sequence«: Laden Sie einen neuen Text, indem Sie die »Open«-Funktion aus dem Project-Menü wählen und mit »SYS:s/startup-sequen-« antworten. Dieser Text ist relativ kurz und einiges daraus kennen Sie vielleicht schon: Es handelt sich um eine Reihe von Befehlen aus dem C-Ordner, die in einer Script-Datei zusammengefaßt sind (Bild 30). Dieses Script wird beim Booten von einer Diskette automatisch abgearbeitet. Sie können hier bestimmte Voreinstellungen festlegen, die Sie auf Ihrer Startdiskette wünschen. So könnten Sie beispielsweise hier bereits eine Shell starten, die dann ständig zur Verfügung steht, oder ein Programm, das Sie jedesmal mit dieser Diskette benutzen (beispielsweise »MicroEmacs«).

Die Startup-Sequence auf Ihrer Original-Workbench sieht vieles vor, was Sie möglicherweise nie brauchen werden. Jeder Befehl verlängert jedoch die Zeit, die der Amiga zum Booten benötigt. Wenn Ihnen der langatmige Startvorgang auf die Nerven geht, können Sie ihn enorm beschleunigen, indem Sie vor alle Befehle, die Sie ausschließen möchten, ein Semikolon schreiben. Wir zeigen Ihnen im folgenden, was Sie unbedingt brauchen und was in welchem Fall wegfallen darf:

»Addbuffers« legt einen zusätzlichen Puffer-Speicher von zehn Blöcken (entspricht 5 KByte) für das interne Laufwerk fest. Sie sollten entweder diesen Wert höher setzen oder die Zeile streichen.

»c:Setpatch« korrigiert einige bekannte Fehler in den Systemroutinen. Allerdings sind diese so unbedeutend, daß Sie durchaus darauf verzichten können.

Die Zeile »Sys:System/FastMemFirst« sorgt dafür, daß alle Speicherzugriffe zunächst auf das schnellere Fast-Memory gelenkt werden. Den Effekt werden Sie vermutlich kaum spüren. Außerdem dürfte Ihr Amiga – wenn er nicht über 1 MByte Speicher hat oder es sich um ein älteres Modell handelt – kein Fast-Memory besitzen. Folglich können Sie diese Zeile bedenkenlos einsparen.

»Binddrivers« bindet zusätzliche Geräte ins System ein, deren Beschreibung im »Expansion«-Ordner liegt. Wenn Sie keine derartigen Geräte haben (schauen Sie sicherheitshalber noch einmal in Expansion nach), können Sie die Zeile ersatzlos streichen.

»Setclock load« holt die Systemzeit aus der batteriegepufferten Uhr. Für einen Amiga 500 ohne Speichererweiterung ist diese Zeile sinnlos, denn die Uhr wird erst mit der Speichererweiterung geliefert.

»FF >NIL: -0« beschleunigt die Darstellung von Text auf dem Bildschirm. Dies ist besonders bei der Arbeit mit Editoren und Textverarbeitungsprogrammen sinnvoll. Wenn Sie dagegen hauptsächlich Grafik mit Ihrem Amiga machen, können Sie auch diesen Befehl streichen.

Die nächsten drei Befehle erleichtern die Arbeit mit der Shell. Solange Sie alles von der Workbench aus erledigen, können Sie diese drei Zeilen durch ein vorangestelltes Semikolon überspringen.

Das gleiche gilt für die Zeile »run execute s:StartupII«. Diese startet ein zweites Script, in dem einige häufig gebrauchte Shell-Befehle dauerhaft in den Speicher geladen werden, so daß sie im Bedarfsfall schneller ausgeführt werden (normalerweise müßten sie erst von Diskette geladen werden).

Die Zeile »Sys:System/Setmap d« sollten Sie unbedingt stehenlassen. Sie aktiviert die deutsche Tastaturbelegung. Im anderen Fall hätten Sie anstelle des »Z« ein »y« und umgekehrt sowie keine deutschen Sonderzeichen (»ä«, »ö«, »ü« und »ß«).

Die nächste Zeile beginnt mit »Path« und legt die sog. Suchpfade fest. Wenn Sie ein Programm starten, so werden alle

hier genannten Verzeichnisse durchsucht. Lassen Sie »ram:«, »c:« und »sys:system« stehen und streichen Sie den Rest – Sie werden ihn sowieso kaum benötigen.

»Loadwb« sollten Sie in jedem Fall stehenlassen, wenn Sie mit der Workbench arbeiten wollen. Solange Sie nur mit der Shell arbeiten, können Sie sich diesen Befehl sparen. Sie soll-

rename S:Startup-Sequence S:Startup.alt

Ziehen Sie MicroEmacs wieder nach oben, speichern Sie die neue Startup-Sequence mit »Save« aus dem Project-Menü und starten Sie Ihren Amiga mit <Ctrl linke-Amiga rechte-Amiga> erneut. Sie werden feststellen, daß die Workbench wesentlich schneller zu sehen ist

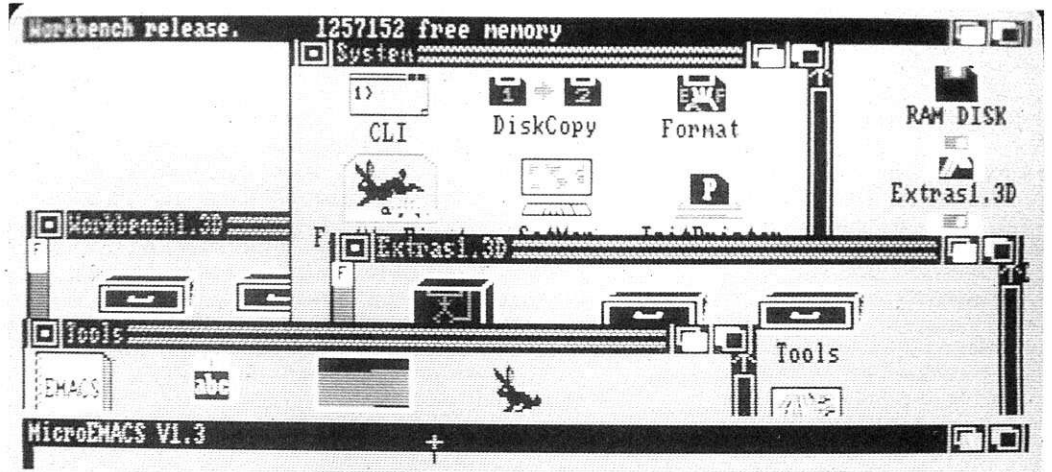


Bild 29. Schnell mal etwas auf der Workbench erledigen ist dank Multitasking kein Problem

```
MicroEMACS V1.3
ADBuffers d0: 10
c:SetPatch >NIL: ; Systemroutinen korrigieren
cd c:
echo "A500/A2000 Workbench disk. Release 1.3 Version 34.20*N"
Sys:System/FastMemFirst ; C00000-Speicher an Listenende
BindDrivers
SetClock load ; Systemzeit aus Echtzeituhr laden (A1000-Besitzer sollten
FF >NIL: -0 ; Textausgabe beschleunigen
resident CLI L:Shell-Seg SYSTEM pure add; Shell aktivieren
resident c:Execute pure
mount newcon:

failat 11
run execute s:StartupII ; Resident-Befehle aktivieren
wait >NIL: 5 mins ; Warten auf Ende von StartupII

SYS:System/SetMap d : Deutsche Tastaturbelegung aktivieren
path ram: c: sys:utilities sys:system s: sys:prefs add ; Pfad für Workbench fest
LoadWB delay ; Vor Weitermachen warten
endcli >NIL:

-- MicroEMACS -- main -- File: s:startup-sequence -----
[Read 21 lines]
```

Bild 30. Die »Startup-Sequence« wird bei jedem Start des Amiga gesucht und abgearbeitet

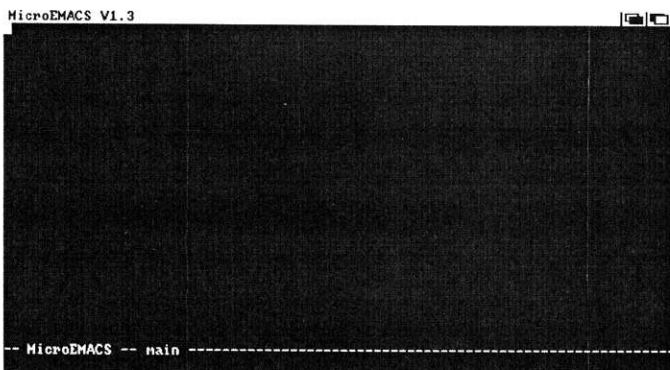


Bild 28. »MicroEmacs« ist eine komfortable Alternative zum wenig beliebten »Ed«

ten dann aber auch »Endcli« streichen, da Sie sonst vor einem leeren Bildschirm sitzen.

Wenn Sie Ihre Startup-Sequence auf diese Weise optimiert haben, speichern Sie sie und verlassen Sie den MicroEmacs. Wenn Sie die alte Startup-Sequence noch benutzen wollen, sollten Sie vor dem Speichern eine Shell starten (indem Sie MicroEmacs mit der linken Maustaste nach unten ziehen, kommen Sie an das Icon) und die alte Startup-Sequence mit folgender Zeile umbenennen:

als es vorher der Fall war. Wir hoffen, Ihnen mit diesen Seiten über die ersten Hürden Ihres Amiga geholfen zu haben. Denken Sie immer daran: Sie können – solange Sie immer nur mit Kopien Ihrer Originaldisketten arbeiten – keinen dauerhaften Schaden anrichten. Probieren Sie also alles aus, was Sie interessiert.

Wenn Sie versehentlich etwas löschen, holen Sie es sich einfach wieder von Ihrem Original. Und denken Sie immer daran: Die ersten Hürden sind die größten. so

Wenn Sie schon mit Ihrer Workbench-Diskette gespielt haben, ist Ihnen sicher die Schublade mit dem Fragezeichen und dem Namen Prefs ins Auge gestochen. Dieser Artikel zeigt Ihnen, wie Sie die darin enthaltenen Programme sinnvoll einsetzen.

Um die folgenden Erläuterungen ausprobieren zu können, starten Sie bitte Ihren Amiga, indem Sie Ihre Workbench-Diskette einlegen und die Schublade »Prefs« durch einen Doppelklick öffnen. In dem nun geöffneten Fenster starten Sie durch zweimaliges Anklicken mit der linken Maustaste das Programm Preferences.

Auf dem Bildschirm erscheint ein Fenster (Bild 1) mit vielen Schaltern und Symbolen, die im folgenden Text ausführlich erklärt werden.

Aber zunächst einmal etwas Grundsätzliches zu diesem Hilfsprogramm:

Als Sie Ihren Amiga gekauft haben und zum ersten Mal starteten, zeigte sich die Workbench in einem hellen Blau, der Mauszeiger war ein roter Pfeil, der bei jeder Mausbewegung recht gemächlich über den Bildschirm wanderte. Wenn Sie bisher noch sehr wenig oder überhaupt nicht mit den Preferences gearbeitet haben, wird das auch jetzt noch so sein. Vielleicht aber haben Sie Ihren Amiga schon einmal von einer anderen Diskette gestartet, sei es bei einem Spielprogramm oder einer Textverarbeitung. Dabei zeigte sich der Bildschirm vollkommen ungewohnt, z.B. war der Bildschirm in einer anderen Farbe, an Stelle des sonst so gewohnten Mauspeils zeigte sich ein anderes Symbol, das sich beim Bewegen der Maus vielleicht auch ganz anders verhielt und mit großem Tempo über den Bildschirm flitzte. All dies und noch vieles mehr können Sie mit den Preferences einstellen.

Preferences ist in erster Linie dafür gedacht, die Arbeitsumgebung Ihren ganz persönlichen Wünschen entsprechend einzustellen. Der große Vorteil dabei ist, daß diese Einstellungen auf der Diskette gespeichert werden und sofort nach dem Einschalten und Hochfahren mit der Workbench-Diskette eingestellt sind. Sie brauchen das Programm nur einmal zu starten, um alle gewünschten Einstellungen vorzunehmen.

Nebenstehend sehen Sie das Fenster (Bild 1), welches nach dem Starten der Preferences auf Ihrem Bildschirm erscheinen sollte. Auffallend ist zunächst der große Kasten in der Mitte, mit dem Sie die Bildzentrierung einstellen können. Probieren Sie diesen Kasten einmal aus, indem Sie das kleine Eck mit der Maus anklicken und bei gedrückter linker Maustaste hin- und herschieben. Wie Sie sehen, bewegt sich der ganze Bildschirminhalt synchron in die entsprechende Richtung mit. Um hierbei die richtige Einstellung zu finden, gehen Sie am besten wie folgt vor: Fahren Sie mit dem Mauspeil die äußeren Ränder des Bildschirms ab. Hierbei sollte immer noch ein Teil des Pfeils sichtbar sein. Mit dieser Methode können Sie sicher gehen, daß auch bei anderen Programmen immer die vollen 256 Bildschirmzeilen zu sehen sind.

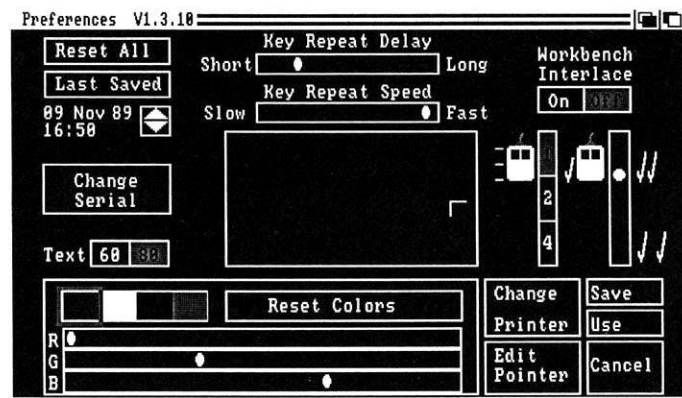


Bild 1. Hier geben Sie Ihrer Workbench das richtige Outfit

Nun gehen wir weiter zu den beiden Schieberegler oberhalb des großen Kastens. Damit können Sie beeinflussen, wie Ihr Amiga auf Eingaben reagiert. Soll der Mauszeiger auf Tasten drücken eine Wiederholungsfunktion, wenn sie länger gedrückt werden. Mit dem oberen der beiden Schieberegler stellen Sie ein, wie lange abgewartet werden soll, bis das gedrückte Zeichen wiederholt wird. Je weiter Sie die Markierung nach rechts schieben, desto mehr Zeit vergeht, bis das Zeichen wiederholt wird. Der zweite Schieberegler beeinflusst hingegen, wie schnell nach dieser Zeitverzögerung die weiteren Zeichen erscheinen, einfacher gesagt wie schnell der Cursor beim Niederhalten einer Taste über den Bildschirm huscht. Diesmal vergrößert sich die Geschwindigkeit des Cursors, je weiter

der Regler nach rechts geschoben wird. Sie fragen sich nun sicher, wie Sie die für Ihren Geschmack richtige Einstellung finden, denn die Schieberegler sind zwar ganz schön und gut, aber sie besitzen keine Skalierung, nach der man sich orientieren könnte. Hierzu ein kleiner Tip: Starten Sie ein anderes Programm, bei dem Buchstaben- oder Zahleneingaben möglich sind, ohne daß Sie das Programm Preferences verlassen (z.B. Notepad). Nun können Sie verschiedene Reglerstellungen ausprobieren, bis Sie die richtige Geschwindigkeit gefunden haben.

Neben dem großen Kasten befinden sich zwei senkrechte Rahmen, an deren oberen Enden jeweils eine Maus abgebildet ist. Die linke Säule ist in drei Felder mit den Ziffern 1, 2 und 4 unterteilt. Hiermit geben Sie an, wie hoch die Geschwindigkeit der Maus sein soll. Probieren Sie dies am besten aus. Aktivieren Sie die »1« und der Maus-

Preferences läßt die Mäuse tanzen

pfeil bewegt sich mit rasantem Tempo, wählen Sie die »2« und Sie haben wieder die normale Geschwindigkeit, oder wählen Sie die »4«, um festzustellen, daß kein Schreibtisch groß genug sein kann, um den Mauszeiger diagonal über die Bildschirmfläche zu bewegen.

Nun zum rechten Schieberegler: Sicher ist Ihnen das dop-

Die Schalt

Der Amiga läßt die Bedürfnisse des so flexibel ist er bei den Benutzern. Dafür verantwortlich das Programm

pelte Anklicken eines Symbols auf der Workbench zum Starten eines Programms schon in Fleisch und Blut übergegangen. Hier stellen Sie ein, wie kurz aufeinander die Maustaste gedrückt werden muß, um als Doppelklick anerkannt zu werden. Im Klartext heißt das, je weiter der Regler nach oben geschoben wird, desto weniger Zeit darf zwischen den beiden Klicks vergehen, je weiter er nach unten geschoben wird, desto mehr Zeit können Sie sich lassen. Wählen Sie diesen Zeitabstand jedoch nicht zu groß, da es sonst passieren kann, daß ungewollt ein Programm gestartet wird, wenn Sie auf der Workbench andere Arbeiten verrichten.

Eine weitere wichtige Einstellung finden Sie unterhalb des großen Kastens: die Farbeinstellung. Im unteren Teil befinden sich drei lange Schieberegler, denen die Buchstaben R, G und B vorangestellt sind. Diese entsprechen den Abkürzungen für Rot, Grün und Blau. Der Amiga bietet den Vorteil, viele verschiedene Farben darstellen zu können. Jede dieser Grundfarben kann in 16 verschiedenen Helligkeitsstufen eingestellt werden, wobei aus deren Kombination bis zu 4096 andere Farbtöne entstehen können. Je weiter Sie den Schieberegler nach rechts schieben, desto heller wird die entsprechende Grundfarbe. Probieren Sie am besten eine Weile aus, bis Sie eine für Sie angenehme Einstellung gefunden haben. Über den Schieberegler befinden sich links vier Kästchen, von denen eines umrahmt ist. Hier können Sie wählen, welche Farbe Sie verändern möchten (Hintergrund, Schreibstift, Zeichenfarbe 1 und 2). Schließlich finden Sie rechts daneben einen Schalter mit der Beschriftung Reset Colors. Hiermit stellen Sie die Farben wie-

mit Preferences

zentrale

sich spielend an
Users anpassen, eben-
der Wahl des Druk-
wortlich ist ein einziges
Preferences.

der auf die zuletzt gespeicherten Werte zurück.

Wenn wir schon beim Zurücksetzen von Werten sind, schauen Sie doch einmal nach ganz links oben zu den beiden Schaltern mit den Beschriftungen Reset All und Last Saved. Mit Reset All stellen Sie alle Werte zurück auf die Einstellungen, die voreingestellt waren, als Sie Ihren Amiga kauften. Dies bezieht sich nicht nur auf die bisher besprochenen Voreinstellungen, sondern auch auf alle, die noch folgen werden. Bei Last Saved werden automatisch die Werte eingestellt, welche Sie zuletzt gespeichert haben.

Direkt unterhalb befinden sich das Datum und die Uhrzeit. Um beides einzustellen, klicken Sie die gewünschte Ziffer an und verändern sie danach mit dem danebenliegenden Auf- und Ab-Pfeil. Leider bleibt diese Einstellung nur so lange erhalten, bis der Amiga

»60« und »80«. Hier können Sie wählen, ob Texte auf dem Bildschirm mit 60 oder 80 Zeichen pro Zeile dargestellt werden sollen. Ersteres ist besonders dann zu empfehlen, wenn Sie Ihren Amiga über einen TV-Modulator an einen Fernseher angeschlossen haben. Beachten Sie jedoch, daß einige Programme (z.B. manche Textverarbeitungen) mit dieser größeren Darstellung nicht zurechtkommen. In diesem Fall müssen Sie auf 80 Zeichen pro Zeile zurückschalten.

Nun kommen wir zu dem Schalter ganz rechts oben. Hier können Sie wählen, ob die Workbench im Interlace-Modus dargestellt wird oder in der nor-

ter Save, alle Einstellungen werden dadurch auf Diskette gespeichert. Beim nächsten Hochfahren des Amiga werden automatisch diese Werte übernommen.

Wollen Sie hingegen eine bestimmte Einstellung nur kurzfristig benutzen, wählen Sie den Schalter Use. Zwar werden auch in diesem Fall die neu gewählten Werte benutzt, diese sind aber nach dem Ausschalten des Amiga verloren.

Wenn Sie schließlich überhaupt keine Veränderung vornehmen wollen, klicken Sie Cancel an. Alle Veränderungen, die Sie gemacht haben, werden nicht beachtet.

Der Schalter Change Serial (links in der Mitte) bringt das Fenster (Bild 2) für die serielle Schnittstelle zum Vorschein. Um die Beschreibung der Einstellungsmöglichkeiten dieser für die meisten noch uninteressanten Schnittstelle kurz zu halten, sind alle Funktion in dem folgenden Kasten aufgeführt.

Für Sie viel interessanter ist dagegen wie der Mauspfel in seinem Aussehen verändert werden kann. Klicken Sie dazu bitte zunächst den Schalter Edit Pointer (rechts unten) an. Sie erhalten einen vollkommen neuen Bildschirm (Bild 3), auf dem in der linken Hälfte der Mauszeiger vergrößert dargestellt ist. Im unteren Teil befindet sich, wie schon auf dem Hauptbildschirm der Preferences, die Farbeinstellung. Diese Farben

Neues Outfit für den Pointer

können vollkommen unabhängig zu den vorher gewählten Farben eingestellt werden. Diese Farben gelten nur für den Mauszeiger.

Sie können nun den Mauszeiger wie mit einem Malprogramm verändern. Wählen Sie zunächst die gewünschte Farbe und verändern Sie den großen Pfeil nach Ihren Vorstellungen. Durch einmaliges Drücken der linken Maustaste wird ein Punkt gesetzt, wenn Sie die Maustaste gedrückt halten, können Sie auch Linien ziehen. In der Mitte sehen Sie den Mauspfel in normaler Größe auf den verschiedenen Hintergründen. Hiermit können Sie beurteilen, ob Ihr Pointer (Mauszeiger) auf allen Hintergründen gut erkennbar ist.

Wenn Sie mit Ihrer Arbeit unzufrieden sind, drücken Sie einfach rechts oben den Taster Restore, wodurch der ursprüngliche Maus-Pfeil wieder hergestellt wird. Durch »Clear« löschen Sie den Pfeil vollständig. Dies ist z.B. dann nützlich, wenn Sie einen völlig neuen Pfeil kreieren möchten. Sollten Sie mit Ihrer neuen Farbwahl unzufrieden sein, können Sie auch dies mit Reset Color rückgängig machen.

Schließlich noch zu dem Taster »Set Point«: Wie Sie sicherlich schon bemerkt haben, lassen sich nur mit der Pfeilspitze Fenster öffnen und Programme starten. Das liegt daran, daß der sogenannte Hotpoint (kleiner oranger Punkt im großen Pfeil) in der Pfeilspitze liegt. Denn nur mit ihm können Icons aktiviert und Programme gestartet werden. Genau diesen Hotpoint legen Sie mit Set Point fest. Nachdem Sie diesen Taster betätigt haben, können Sie diesen Punkt beliebig positionieren. Die neue Position wird mit einem kleinen orangefarbenen Rechteck markiert.

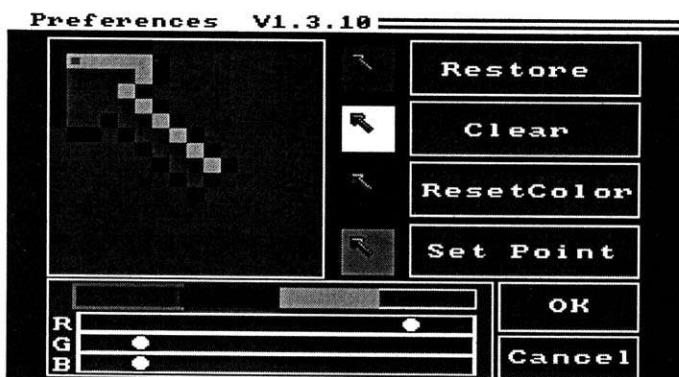


Bild 3. Das richtige Styling für Ihren Mauspointer

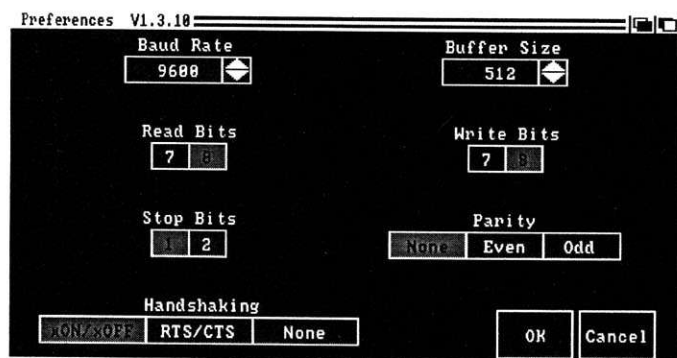


Bild 2. Das Schaltfeld für die serielle Schnittstelle

abgeschaltet wird. Jedoch wenn Ihr Amiga eine batteriegepufferte Uhr hat, können Sie die richtige Zeit über das CLI fest einstellen. Die Uhrzeit und das Datum werden dann auch beim erneuten Einschalten des Rechners richtig angezeigt.

Links oberhalb der Farbeinsteller finden Sie das Wort Text, gefolgt von zwei Schaltern mit

malen Darstellung. Im Interlace-Modus wird die Anzahl der horizontalen Bildschirmzeilen verdoppelt, was bei den meisten Monitoren und Fernsehern ein starkes Flimmern hervorruft. Sie sollten daher hier die Einstellung auf Off belassen.

Um die so gewählten Werte auf Diskette festzuhalten, befindet sich rechts unten der Schal-

- **Baud Rate:**
Gibt die Anzahl der pro Sekunde übertragenen Bits an.
- **Buffer Size:**
Gibt die Größe des im RAM definierten Speicherraumes an.
- **Read Bits:**
Gibt an, wie viele Bits der Amiga einlesen soll, um ein Zeichen darzustellen.
- **Write Bits:**
Gibt an, wie viele Bits der Amiga sendet, um ein Zeichen darzustellen.
- **Stop Bits:**
Anzahl der Stop Bits, die nach einem Zeichen gesetzt werden sollen.
- **Parity:**
Diese Funktion dient zur Erkennung von Übertragungsfehlern.
- **Handshaking:**
Dient zur Kontrolle des Datenflusses der seriellen Schnittstelle.

Haben Sie einen schönen und farbenfrohen Mauspfad gezeichnet? Dann klicken Sie OK an (oder Cancel, wenn Ihnen Ihr Werk doch nicht so sehr zusagt). Wie Sie sehen, befinden wir uns nun wieder auf dem Hauptbildschirm der Preferences, doch wollen wir diesen auch gleich wieder verlassen, indem Sie Change Printer anklicken.

Wiederum öffnet sich ein neues Fenster (Bild 4) mit einer Vielzahl von Schaltern und Symbolen, die es zu klären gilt. Wie Sie sicher schon bemerkt haben, befinden wir uns nun in dem sehr umfangreichen Fenster zum Einstellen des Druckers und aller dazu notwendigen Parameter.

Zunächst einmal legen Sie fest, ob Ihr Drucker am Parallel-Port oder am Serial-Port Ihres Amiga angeschlossen ist. In den meisten Fällen ist dies der Parallel-Port, wenn Sie es aber nicht genau wissen, schlagen Sie bitte in Ihrem Druckerhandbuch nach. Die Anschlüsse auf der Rückseite des Amiga sind beschriftet, so daß es hierbei keine Probleme geben dürfte.

Rechts oben finden Sie eine Auflistung verschiedener Drucker. Anfangs stehen hier nur zwei Druckertreiber zur Auswahl, »Custom« und »generic«. Um eine größere Auswahl zu bekommen, müssen Sie einige Treiber von Ihrer Extras-Diskette auf Ihre Workbench in das Unterverzeichnis »Devs/printers« kopieren. Wie dies genau geht, lesen Sie bitte in unseren Einsteigerkurs (Seite 28) oder in Ihrem Handbuch nach. Wenn Sie das erledigt haben, sollten Ihnen nun mehrere Druckertreiber zur Verfügung stehen. Durch das Klicken auf einen der links stehenden Pfeile können Sie die Treiberliste um eine Spalte in die jeweilige Richtung weiter-schalten. Welche Sie dort finden, kann unterschiedlich sein. Benutzen Sie jedoch die normale, unveränderte Workbench 1.3, sollte dies wie in der unteren Abbildung (Bild 2) aussehen. Sehen Sie zunächst nach, ob unter den angegebenen Druckern auch der Name Ihres Druckers aufgeführt ist. Wenn dies der Fall ist, bringen Sie diesen Namen bitte in die mittlere Position, so daß dieser orange unterlegt ist. Damit ist er selektiert.

Vielfach sind Drucker zum Epson-Standard kompatibel, so daß auch ein Epson-Druckertreiber mit Ihrem Drucker zusammenarbeiten kann. Schla-

gen Sie dies am besten ebenfalls in Ihrem Druckerhandbuch nach. Die Erfahrung zeigt außerdem, daß viele Drucker auch mit anderen Druckertreibern arbeiten, hier hilft am besten Ausprobieren.

Wenn Sie für Ihren Drucker einen speziellen Druckertreiber auf einer anderen Diskette haben (dies ist teilweise bei guten Textverarbeitungen der Fall), können Sie auch diesen Treiber verwenden. Hierzu wählen Sie zunächst den Begriff Custom als Druckertreiberangabe und tragen daraufhin rechts neben Custom Printer Name den Namen Ihres Druckertreibers ein.

Sollten Sie einen derart exotischen Drucker besitzen, daß alle bisherigen Versuche fehlerhaft oder überhaupt nicht funktionierten, bleibt noch eine weitere Variante, wobei jedoch keinerlei spezielle Schriftstile und -arten möglich sind. Wählen Sie hierzu generic an.

Nachdem nun der entsprechende Druckertreiber eingestellt ist, können wir uns den Feinheiten der Druckerparameter zuwenden. Links in der Mitte kann zunächst die Größe der Druckseite bestimmt werden. Hier können Sie wählen zwischen vier vorgegebenen Formaten und einer beliebigen. In folgender Tabelle finden Sie die Größen der Seitenformate:

Bild 4. Mit Prefs ist die Druckeranpassung ein Kinderspiel

U.S. Letter (U.S. Briefformat) 8 1/2 Zoll breit, 11 Zoll lang
U.S. Legal (U.S. Normalformat) 8 1/2 Zoll breit, 14 Zoll lang
Narrow Tractor (Schmaler Traktor) 9 1/2 Zoll breit, 11 Zoll lang
Wide Tractor (Breiter Traktor) 14 7/8 Zoll Breit, 11 Zoll lang

Wenn Sie hingegen explizit die Zeilenanzahl pro Seite angeben möchten, wählen Sie Custom und tragen danach rechts neben Length die Anzahl der Zeilen ein.

Bei vielen Dokumenten, speziell bei Briefen ist es wünschenswert links und rechts des Blattes einen Rand zu lassen. Um diesen Rand nicht um-

ständig mit der Textverarbeitung durch Leerzeichen angeben zu müssen, können Sie links unten den linken und rechten Randsteller (Left und Right Margin) setzen. Hierbei steht jeweils die angegebene Zahl für die äußerste linke bzw. rechte Druckspalte. Wenn Sie, wie auf der Abbildung angegeben, den linken Rand auf 5 und den rechten auf 75 setzen, wird nur von Spalte 5 bis 75 gedruckt. Sie erhalten also einen linken Rand von 4 (nicht 5) Leerzeichen. Unter Paper Type geben Sie an, welchen Papiertyp Sie verwenden wollen. Wählen Sie »Fan-fold«, wenn Sie auf Endlospapier, und »Single«, wenn Sie auf Einzelblätter drucken oder einen automatischen Einzelblatt-einzug besitzen.

gegeben. Auf einem normalen DIN-A4-Blatt haben bei 10 Pica 80 Zeichen pro Zeile Platz, bei 12 Elite etwa 96 Zeichen und bei 15 Fine etwa 132 Zeichen. Für Briefe und ähnliche Dokumente sollte 10 Pica verwendet werden. Die kleineren Zeichendichten sind z.B. für Tabellen und Listen sinnvoll, um auf kleinem Raum viele Informationen unterzubringen.

Neben der Zeichendichte kann auch der Abstand der Zeilen variiert werden. Unter Spacing wählen Sie zwischen »6 lpi« (Zeilen pro Zoll) und »8 lpi«. Hierbei sollte für Briefe wiederum die Einstellung 6 lpi bevorzugt werden.

Alle Einstellungen für den Drucker sind nur Voreinstellungen. Dies heißt aber nicht, daß

Bild 5. Mit oder ohne Farbe, das ist die Frage

Sie nach dieser Festlegung nur diese Werte verwenden können. Es ist weiterhin möglich, über das Programm durch Steuersequenzen oder direkt am Drucker Veränderungen vorzunehmen. Da erfahrungsgemäß aber ein Großteil aller Ausdrucke immer mit den gleichen Druckereinstellungen erstellt werden, ist diese Voreinstellung sehr hilfreich.

Sie werden festgestellt haben, daß wir auf dem jetzigen Fenster vom Preferences neben dem OK- und Cancel-Schalter zwei Schalter noch nicht besprochen haben. Diese befinden sich rechts in der Mitte mit den Bezeichnungen »Graphic 1« und »Graphic 2« (die ältere Version des Preferences hat hier nur einen Schalter mit der Bezeichnung »Graphic Select«).

Klicken Sie also zunächst einmal den Schalter »Graphic 1« bzw. »Graphic Select« an. Es erscheint wiederum ein neues Fenster (Bild 5) mit einer Reihe von Voreinstellungen, die sich jetzt speziell auf den Ausdruck von Hardcopies beziehen. Sollten Sie einen Typenraddrucker besitzen, können Sie die Be-

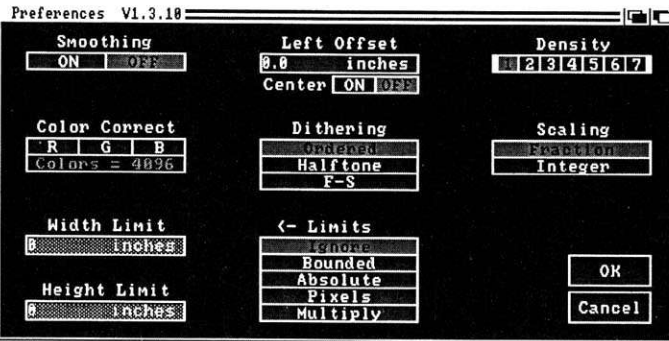


Bild 6. Mit diesen Funktionen haben Sie Ihre Bilder im Griff

schreibungen hierzu überspringen, da nur mit Matrix-, Tintenstrahl- und Laserdruckern Hardcopies erstellt werden können.

Als erstes fällt im oberen Teil die Zahlenleiste von 1 bis 15 auf. Der Amiga kann eine ganze Reihe verschiedener Farben auf dem Bildschirm darstellen. Es stellt sich nun aber die Frage, wie diese Bilder auf einem normalen Schwarzweiß-Drucker ausgegeben werden, der nur eine Farbe drucken kann. Des Rätsels Lösung ist folgendes: Der Schieberegler ganz oben stellt einen Schwellwertregler dar. Mit diesem Schalter stellen Sie die Schwelle ein, ab welchem Lichtwert eine Farbe als Schwarz interpretiert wird. Steht dieser Schalter auf den Wert 1, wird nur reines Schwarz vom Drucker auch schwarz ausgegeben. Je höher Sie diesen Wert wählen, um so hellere Farben werden auf dem Ausdruck schwarz gedruckt. Dieser Regler wird nur dann beachtet, wenn Sie bei »Shade Black and White« eingestellt haben. Wie in der Abbildung

Drucken à la Gutenberg

(Bild 5) ist ein Wert von 8 für viele Ausdrücke ausreichend. Probieren Sie aber ruhig mit verschiedenen Werten Ihre Hardcopies zu erstellen, denn oft verändert dieser Wert die Aussagekraft des Bildes. Auf diese Weise finden Sie am schnellsten Ihre optimalen Werte.

Links unter diesem Schieberegler befindet sich die Einstellung, wie die Grafik auf das Papier gebracht werden soll. Sie haben die Wahl zwischen horizontalem und vertikalem Ausdruck. Bei vertikalem Ausdruck wird die volle DIN-A4-Größe ausgenutzt, die Hardcopy wird etwa doppelt so groß, wie bei horizontalem Ausdruck.

Wiederum darunter können Sie einstellen, wie die Farben

auf dem Bildschirm beim Ausdruck übersetzt werden sollen. In den Einstellungen »Gray Scale« und »Gray Scale 2« (letzteres nur bei Preferences 1.3) werden die Farben in verschiedene Graustufen umgewandelt. Wobei der Schalter »Gray Scale 2« nur für User interessant ist, welche einen A2024-Monitor verwenden. Für Ihre Ausdrücke auf einem Matrixdrucker sollten Sie diese Einstellung wählen, da auf diese Weise die besten Ergebnisse erzielt werden. Wenn Sie jedoch mehr Wert auf scharfe Konturen und Kontraste legen, ist eher die Einstellung »Black and White« zu bevorzugen, wobei hier der zuvor beschriebene Threshold-Wert berücksichtigt wird. Sollten Sie der glückliche Besitzer eines Farbdruckers sein, wählen Sie natürlich Color, so daß Sie auch beim Ausdruck nicht auf die hervorragenden Grafikeigenschaften des Amiga verzichten müssen.

Bei den bisher beschriebenen Ausdrucken in Schwarz bzw. Grautönen wurde davon ausgegangen, daß dunkle Farbtöne als Schwarz und hellere in Grau, Hellgrau usw. ausgedruckt werden. Bei manchen Bildern kann es aber vorteilhaft sein, wenn dies genau umgedreht behandelt wird. Hierzu brauchen Sie nur auf der rechten Seite die Einstellung Image von »Positiv« auf »Negativ« umzustellen.

Wenn Sie auf diesem Bildschirm Ihre Eingaben beendet haben, wählen Sie »OK« zum Übernehmen oder »Cancel«, um die neuen Eingaben zu ignorieren. Sie sind nun wieder auf dem Hauptbildschirm der Preferences. Für all diejenigen unter Ihnen, welche Workbench 1.3 besitzen, finden Sie noch einen zweiten Unterpunkt zum Grafikausdruck. Bitte klicken Sie dazu »Graphic 2« an.

Sie finden hier eine ganze Anzahl weiterer Voreinstellungsmöglichkeiten (Bild 6) be-

züglich des Ausdrucks von Hardcopies. Diese Angaben sind für einen Ausdruck nicht zwingend erforderlich, sie können jedoch die ausgedruckte Grafik entscheidend verbessern.

Links oben können Sie »Smoothing« ein- (ON) oder ausschalten (OFF). Unter »Smoothing«, auch besser bekannt unter »Antialiasing«, versteht man das Glätten von Konturen, um die störende Treppenbildung bei Kanten zu vermeiden. Leider verlängert sich die Ausdruckzeit durch diese Funktion deutlich. Darunter finden Sie die Farbkorrekturschalter, mit ihnen kann man die Farben des Ausdrucks den Bildschirmfarben angleichen. Die passende Einstellung kann nur durch einen Probeausdruck gefunden werden. Die Anzahl der nach der Korrektur noch darstellbaren Farben wird eine Zeile tiefer angezeigt.

Sehr wichtig sind die beiden Angaben, die links unten eingegeben werden. Hier wird die Größe des Ausdrucks genau festgelegt. In direkter Beziehung zu den Werten, die Sie hier eintragen, steht die Auswahl, welche Sie rechts daneben unter »Limits« getroffen haben. Die fünf verschiedenen Angaben darunter haben folgende Eigenschaften:

- Ignore: Die links gemachten Werte werden ignoriert.
- Bounded: Die links gemachten Werte werden als Größen in Zoll interpretiert und die maximale Größe der Grafik wird von den links eingetragenen Werten begrenzt.
- Absolute: Die links gemachten Werte werden als Größen in Zoll interpretiert und die Hardcopy in genau dieser Größe erstellt.
- Pixel: Wie oben, jedoch werden die Werte als Pixelangaben interpretiert.
- Multiply: Die links gemachten Werte werden als Multiplikator interpretiert.

Eine weitere nützliche Einstellung können Sie oben in der Mitte unter »Left Offset« vornehmen. Hier wird der Abstand vom linken Rand in Zoll angegeben (1 Zoll = 2,54 cm). Wenn Sie darunter »Center« auf (ON) stellen, wird die Grafik automatisch zentriert. Hierbei wird dann aber die Angabe des linken Randes ignoriert. Außerdem richtet sich die Zentrierung nicht nach dem tatsächlich eingelegten Blatt, sondern nach

der Blattgröße, die Sie vorher gewählt haben.

Wenn Sie Probleme mit zu hoher Farbsättigung bei hoher Druckdichte (über 180 Punkte pro Inch) haben, ist die Funktion »Dithering« genau die richtige für Sie. Hier wird durch drei verschiedene Schattierungsverfahren versucht, dieses Manko zu beheben.

Ordered entspricht der normalen Schattierung.

Halftone - hier werden die Schattierungen mit Hilfe von Halbtönen erzeugt.

F-S - hier wird die Fehlerverteilungsmethode von Floyd-Steinberg angewandt, um die Schattierung zu erzeugen.

Welches Verfahren für Ihren Drucker das geeignetste ist, müssen Sie schon selbst herausfinden.

Eine weitere Option ist »Scaling«. Sie muß auf Integer gestellt werden, wenn Verzerrungen im Ausdruck erscheinen. Diese werden dann hoffentlich behoben.

Eine letzte wichtigere Einstellung kann schließlich noch rechts oben unter »Density« gemacht werden. Je größer Sie hier den Wert wählen, desto feiner wird der Ausdruck, d.h. desto höher wird die Auflösung. Welcher Druckdichte die einzelnen Werte entsprechen hängt von Ihrem Drucker ab, und kann für die einzelnen Druckertypen im Handbuch nachgeschlagen werden.

Nach dieser umfangreichen Beschreibung sollte die für Sie ideale Einstellung gefunden sein. Nachdem Sie auf das Hauptbildmenü (Bild 1) der Preferences zurückgekehrt sind, müssen Sie aber noch eine wichtige Sache erledigen. Um Ihre gewonnenen Voreinstellungen nicht nach dem Ausschalten des Amiga zu verlieren, müssen Sie diese Daten mit dem Schalter »Save« auf Ihre Workbench-Diskette schreiben. Danach brauchen Sie sich nicht mehr um die richtige Einstellung zu kümmern. Beim nächsten Booten Ihres Amiga wird sich dieser in Ihrer individuell gestalteten Workbench und mit komplett installierten Druckerparametern zeigen. *pe*

Andreas Regul beschäftigt sich seit Anfang 1988 intensiv mit dem Amiga, und kennt diesen Computer, nicht zuletzt durch seine exzellenten Programmierkenntnisse, durch und durch. Sie erreichen ihn über unsere Verlagsanschrift (Markt & Technik Verlag AG, Redaktion Sonderhefte, z.Hd. Herrn Regul, Hans-Pinsel-Straße 2, 8013 Haar bei München).

Eine Riesen-Menge nützlicher und interessanter Programme finden Sie im PD-Pool. Wir schaffen Überblick und sagen Ihnen, was Sie brauchen.

von Martin Jobst

Zum Unkostenpreis hat der Amiga-Anwender Zugriff auf einen riesigen Software-Teich - den PD-Pool. PD- oder Public-Domain-Software, das sind Programme, die jeder frei kopieren kann: Software zum Diskettenpreis.

siert fehlerhafte Sektoren auf Disketten, macht aber mehrere Versuche, diese zu kopieren. Somit wird der Anwender sofort auf schadhafte Disketten aufmerksam, noch bevor diese unlesbar sind. »TurboBackup« ist multitaskingfähig, läuft also gleichzeitig mit anderen Programmen. Dieses Programm gehört in jede Softwaresammlung. Sie finden »TurboBackup« auf der Fish 139.

Das CLI oder Command Line Interpreter ist eine große Hürde für Neulinge auf dem Amiga. Im Gegensatz zur Workbench ist es äußerst umständlich zu bedienen. Ganz abgesehen davon müssen Sie sich als Anwender viele Befehle merken. Trotzdem ist das CLI zur Arbeit mit Dateien unerlässlich. Wesentlich einfacher wird die Sa-

che mit »Diskman« (Bild 2). Das Programm, zu finden auf Fish 82, setzt dem CLI eine Benutzeroberfläche ähnlich der Workbench auf, und macht die Arbeit leicht. Durch Anklicken der einzelnen Schalter lassen sich Programme löschen, kopieren, umbenennen oder auch Bilder anzeigen. Wer Komfort sucht, sollte auf »Diskman« nicht verzichten.

Viele Programme verwenden die in den Preferences voreingestellten Farben. Oft ist es jedoch nützlich, die Farben von einem Programm aus verändern zu können. Mit »PopColours« von RPD 83 ist das möglich. Das Programm läuft im Hintergrund. Um die Farben Ihren Wünschen anzupassen, brauchen Sie diese mit den Schieberegeln (Bild 3) nur ent-

Nützlich

Fast

sprechend einzustellen. Auf diese Weise ersparen Sie sich das Laden der Preferences. Ein weiterer Meilenstein auf unserer Reise.

Ob aus Langeweile, böser Absicht oder Faszination programmiert, lästig - und mitunter gefährlich - sind sie allemal: Computer-Viren auf dem Amiga. Analog zu ihren biologischen Artgenossen vermehren und verbreiten sich die kleinen Schädlinge wann und wo sie nur können. Da Computer-Viren - einmal im Speicher - die unangenehme Fähigkeit besitzen, sich auf jede im Laufe der Zeit (bis zum Abschalten des Computers) eingelegte Diskette zu kopieren, entsteht durch Raubkopien natürlich ein geradezu idealer »Umschlagplatz« für Viren. Aber auch PD-Software stellt einen idealen Nährboden für Viren dar.

Wie bei ihren »natürlichen« Vorbildern gibt es auch bei Amiga-Viren gut- und böserartige. Dabei finden sich glücklicherweise nur einige wirklich gefährliche unter einer Vielzahl relativ harmloser Viren. Während etwa der »SCA«-Virus nur von Zeit zu Zeit seine Anwesenheit durch flockige Sprüche kundtut und schlimmstenfalls den (manchmal) benötigten Bootblock einer Diskette zerstört, finden sich auch widerli-

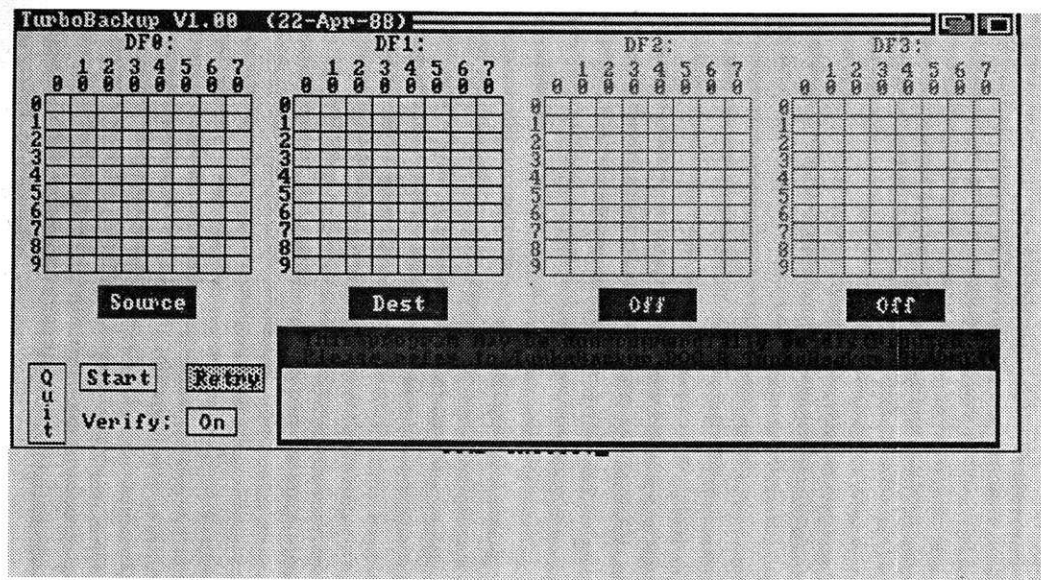


Bild 1. Das Kopierprogramm »TurboBackup« ist schnell und zuverlässig

»Ja, das kann ja wohl nichts sein!« Doch. PD-Programme können in einigen Bereichen qualitativ mit kommerzieller Software durchaus mithalten. Gerade im Bereich kleiner, nützlicher Hilfsprogramme sind sie unübertroffen. Begeben Sie sich mit uns auf eine kleine Reise in das Land der freikopierbaren Software und überzeugen Sie sich von deren Qualitäten.

Das Kopierprogramm »TurboBackup« (Bild 1) ist unsere erste Station im Software-Schlaraffenland. Es bietet nicht nur wesentlich mehr Komfort gegenüber dem Kopieren von der Workbench, sondern ist auch schneller. »TurboBackup« bietet aber noch mehr. So zählt es zu den sichersten Kopierprogrammen und überprüft alle kopierten Daten sofort. Es signalisiert

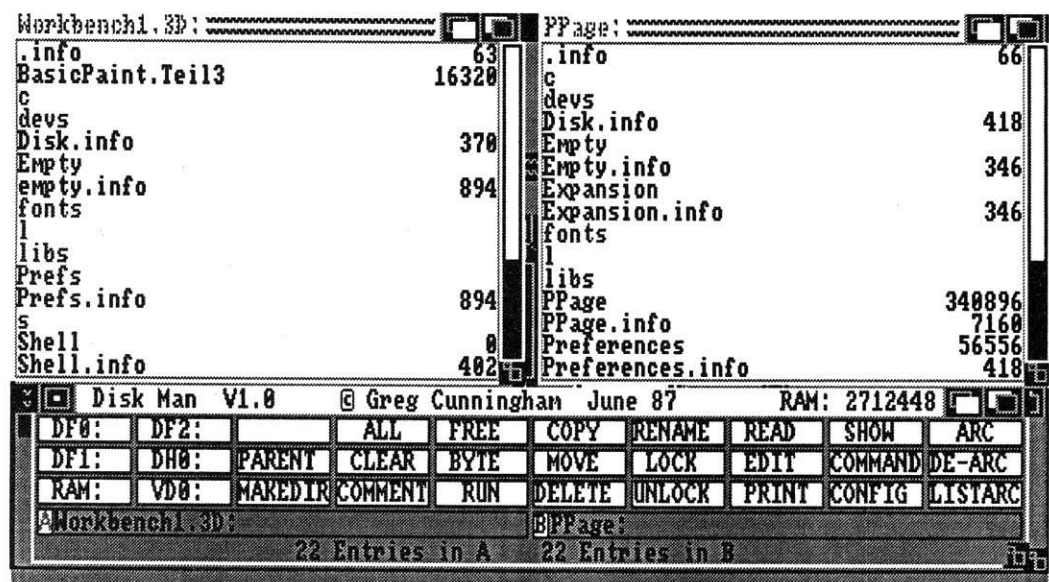


Bild 2. »Diskman« erleichtert die Arbeit mit Files und Dateien ungemein

PD-Software im Überblick

umsonst

chere Gesellen unter den Amiga-Bazillen.

Einer von ihnen ist der »Byte Bandit«-Virus. Unangenehmerweise schaltet er nämlich – nach Lust und Laune – nach einer gewissen Zeitspanne den Bildschirm dunkel. Weg sind sie, der mühevoll geschriebene Text, die sorgfältig eingegebenen Daten, der ausdauernd erreichte High-Score. Oder? Noch nicht ganz! Mit folgender Tastenkombination erblickt Ihr Amiga wieder das Licht der Welt: linke Alt-Taste, linke Amiga-Taste, Leerraum-Taste, rechte Amiga-Taste, Rechte Alt-Taste. Betätigen Sie diese Tasten nacheinander und halten Sie sie gedrückt, erwecken Sie den Amiga wieder zum Leben. Eine weitere Facette in den Untaten dieses Virus ist die fallweise Zerstörung der Struktur einer Diskette, was Datenverlust gleichkommt.

Alle bisher erwähnten Viren verbreiten sich über den Bootblock einer Diskette. Sie sind dadurch leicht aufzufinden und zu entfernen. Schwieriger wird es hingegen mit sogenannten Link-Viren (link, engl.: verbinden, anhängen). Wie der Name schon sagt, hängt sich ein derartiger Virus an den Code eines Programms und ist damit sehr schwer zu erkennen. Glücklicherweise haben Link-Viren auf dem Amiga – im Gegensatz zum Atari ST oder IBM-PC – noch keine große Verbreitung gefunden. Der einzige uns bekannte ist bislang der »IRQ«-Virus.

Während Sie nun vielleicht zitternd vor Ihrem Computer sitzen und um die friedvolle Existenz Ihrer Daten und Programme bangen, können wir Sie aber gleich beruhigen. Es wären keine Viren, wenn nicht bereits auch Gegenmittel dafür gefunden worden wären. In der Medizin sind es Arzneien, bei Computern sind es schlicht und einfach Virus-Killer. Viele gibt es bereits auf dem Markt, einige davon erzeugen netterweise sogar neue Viren (siehe Nebenwirkungen). Einem »Killer«, dem Sie Vertrauen schenken können und der durch ständige Updates möglichst alle Viren erkennt und eliminiert ist »Vi-

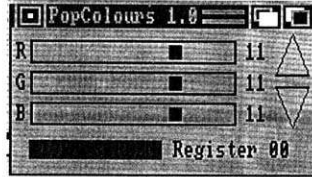


Bild 3. Mit »PopColours« verändern Sie die Bildschirmfarben

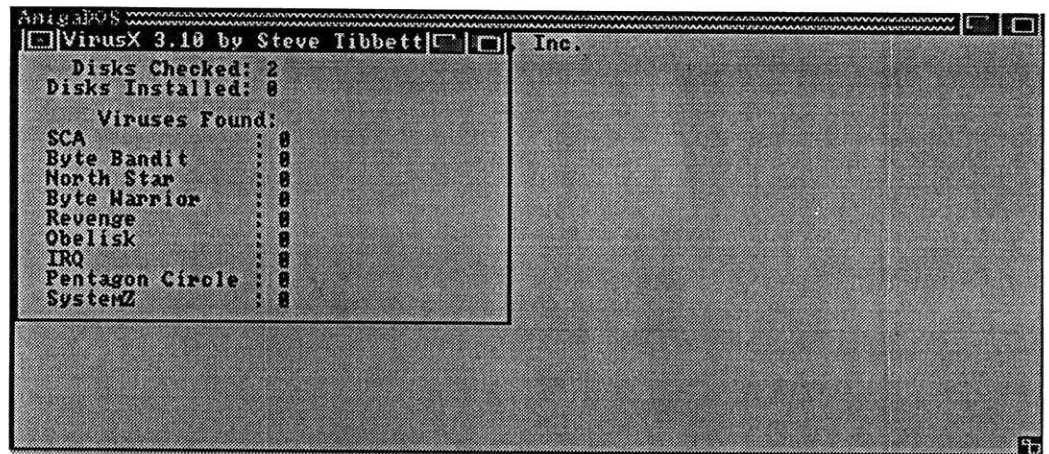


Bild 4. »VirusX« gibt Viren keine Chance und läßt Sie ruhig schlafen

rusX« (Bild 4). Das Programm stammt vom amerikanischen Autor Steve Tibbett und ist der nächste Zwischenstop auf unserer Reise. Die neueste Version 3.20 befindet sich auf der Fish 216. VirusX läuft im Hintergrund und untersucht sowohl den Speicher als auch eingelegte Disketten nach Viren, warnt beim Auffinden solcher und entfernt sie auf Wunsch.

VirusX ist zur Zeit eines der besten Mittel gegen den Übergriff von Viren auf Ihre Daten. Perfekt kann es dennoch nicht sein: Ständig finden neue Viren den Weg in die Lebensadern des Amiga, vergeuden Programmierer ihre Energien zum Schaden nichtsahnender Anwender. Trotz ständiger Upda-

tes ist es nicht möglich, daß VirusX auf dem aktuellsten Stand ist: kommt eine neue Version auf den Markt, gibt es schon zwei neue Viren. Trotzdem können sich Ihre Daten mit VirusX einigermaßen sicher fühlen, zumal das Programm auch meldet, wenn der Bootblock einer Diskette zwar kein bekannter Virus ist, aber vom Standardformat abweicht. Dann handelt es sich entweder um einen neuen Virus oder (Vorsicht! Nicht löschen!) um einen speziellen Startblock für ein Programm.

Wie ärgerlich! Aus Versehen haben Sie eine wichtige Datei von Diskette gelöscht. C'est la

se Zeit ruhen lassen und keine Taste betätigen – den Bildschirm dunkel. Dies hat den Sinn, daß sich die Farben nicht in die Leuchtschicht einbrennen und Ihren Bildschirm beschädigen. Die Gefahr besteht dann, wenn sich über einen längeren Zeitraum ein gleichbleibendes Bild auf dem Monitor befindet.

Ein beliebiger Tastendruck läßt das verschwundene Bild wieder erscheinen. Auf Wunsch können Sie sich von Mackie in der »Dunkelphase« auch durch Kaleidoskop-Effekte verwöhnen lassen. Als Bonbon hat Mackie noch eine Hot-Key-Funktion. Diese ermöglicht per Tasten-

komination das Öffnen eines CLI-Fensters aus laufenden Programmen heraus. Die letzte Station auf unserem Trip ist IFF Image-Filter. Wie sein Vorbild »Butcher« (siehe Übersicht Software, ab Seite 18) ermöglicht es mit einer Reihe von Werkzeugen das Nachbearbeiten von IFF-Bildern. So beherrscht es das mosaikartige Auflösen eines Bildes oder läßt mit Hilfe der Funktion »Smooth« die Konturen eines Bildes weicher werden.

Hat Sie diese kleine Auswahl aus dem Reich der PD-Programme überzeugen können? Wenn unsere kleine Reise für Sie zu kurz war und Sie noch mehr PD-Luft schnuppern wollen, dürfen wir Sie auf unser Sonderheft 5 verweisen. Das Heft befaßt sich mit dem Thema Public Domain und bringt eine Übersicht über alle verfügbaren PD-Programme.

so

so

so

PD-Programme

Programm	Diskette
TurboBackup	Fish 139
Diskman	Fish 82
PopColours	RPD 83
VirusX	Fish 216
Undelete	RPD 103
Mackie	Fish 161
IFF Image Filter	Tornado 25

Martin Jobst studiert Publizistik und Kommunikationswissenschaft in Salzburg und arbeitet seit drei Jahren als freier Mitarbeiter für Zeitschriften des Markt & Technik Verlages. Sie können ihn erreichen über unsere Verlagsanschrift (Markt & Technik Verlag AG, Redaktion Sonderhefte, z. Hd. Martin Jobst, Hans-Pinsel-Straße 2, 8013 Haar bei München).

Dieser Artikel soll Sie vor großem Schaden bewahren, der Ihnen den Spaß an Ihrem Amiga verderben könnte.

von Brigitte Bobenstetter

Viel Spaß mit Ihrem Amiga und beehren Sie uns bald wieder.« Mit diesen Worten beendet sich ein fast dreistündiges Verkaufsgespräch.

platz ein weiteres Problem verbunden. Wohin mit dem computerfeindlichen Ehepartner? Natürlich kann nicht jeder User ein Arbeitszimmer sein eigen nennen. Aber es hat sich für den Familienfrieden bewährt, wenn man sich mit seinem ganzen Computerzubehör an ein ruhiges Plätzchen zurückzieht. Eine Patentlösung bei dieser schwierigen Aufgabe gibt es allerdings nicht. Hier muß sich jeder eine individuelle Lösung zurechtzimmern.

Hat sich für den Computer nebst allerlei Zubehör ein Platz gefunden, geht es daran, die-

Der optimale Standort für Computer

Kein Platz an der Sonne

sen funktionell und bequem einzurichten. Egal, ob Anwender oder Programmierer – für beide Gruppen gelten bezüglich des Standorts bestimmte Faustregeln:

Der Arbeitsplatz sollte nie zu nahe an der Heizung stehen. Ein Computer besteht bekanntlich zum größten Teil aus einer Anhäufung von sogenannten Cips, welche nur dann gewillt sind einwandfrei zu arbeiten, wenn sich die Temperatur auf dem Motherboard (Hauptplatine) im Gehäuseinneren in Grenzen hält. Der maximale Wert beträgt ca. 70 Grad.

Ebenso schädlich für die elektronischen Bauteile ist extreme Feuchtigkeit. Insbesondere der Monitor, der ständig unter Hochspannung steht, würde das mit Arbeitsverweigerung, sprich Kurzschluß, quittieren. Also, der gewiß sehr ruhige Arbeitsplatz Keller ist möglichst zu vermeiden!

Ein sehr diffiziles und heikles Thema ist die Wahl des Bildschirmstandortes. Gerade an diesen Standort werden besondere Grundvoraussetzungen gestellt:

– Der Monitor sollte nie in unmittelbarer Nähe des Fensters

stehen (ist der Ausblick auch noch so schön), weil hier das Licht am grellsten ist und dadurch die Anpassung der Augen an die dunkle Bildschirmmaske erschwert wird.

– Der Monitor soll stets mit einer zur Fensterfront parallelen Blickrichtung aufgestellt sein: Falsch ist ein Aufstellen mit vom Fenster abgewandter Blickrichtung, da sich dann das Fenster nicht im Bildschirm spiegelt. Ebenfalls falsch ist ein Aufstellen mit Blickrichtung zum Fenster, da der Kontrast zwischen dunklem Bildschirm und hellem Tageslicht zu Sehbeschwerden führen kann.

– Die parallele Blickrichtung zur Lichtquelle sollte auch bei künstlicher Beleuchtung eingehalten werden.

Der Monitor als Spiegel

Wenn es die räumlichen Verhältnisse nicht zulassen, den Bildschirm wie beschrieben aufzustellen, gibt es (Gott sei Dank) noch eine Reihe von Möglichkeiten, die Blendwirkung, Spiegelungen und zu starke Kontraste zu verhindern: z. B. durch Drehen, Absenken oder Neigen des Monitors. Einfach lösen läßt sich das Problem durch Jalousien an den Fenstern oder Stellwände (Raumteiler). Der Fachhandel bietet auch spezielle Entspiegelungsfilter für Monitore an.

Ist es jedoch schon zu spät für meine Ratschläge, sprich Ihre Augen sind schon geschädigt, können Sie sich als Mitglied einer gesetzlichen Krankenkasse eine spezielle »Computerbrille« verschreiben lassen. Damit können Beschwerden wie »Augenflimmern« oder »Augentränen« am Bildschirm vermieden werden.

Auch beim Arbeitstisch gibt es eine Reihe von Anforderungen zu beachten. Zum Beispiel muß die Abmessung der Tischplatte eine flexible Anordnung der Arbeitsmittel erlauben (Drucker, Tastatur, Floppy). Die Anordnung der Arbeitsmittel, richtet sich nach der jeweiligen Tätigkeit. Denn, für die Daten-

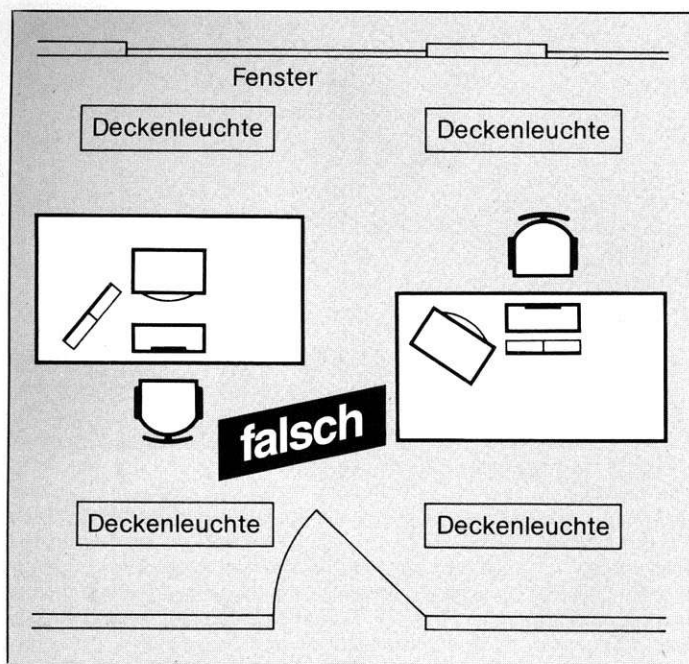


Bild 1. So sind Sehbeschwerden vorprogrammiert

sprach. Am Ende stand ich mit einem Amiga 500 strahlend vor der Kasse und zückte (weniger strahlend) meinen Geldbeutel. Nun gehöre ich also auch zu den über 300000 glücklichen Amiga-Besitzern in Deutschland. Doch wer denkt, mit dem Kauf sei alles paletti, hat sich gründlich getäuscht.

Meine Probleme begannen just in dem Augenblick, als es darum ging, für den Amiga ein kuscheliges Plätzchen zu finden, sprich einen funktionellen Computer-Arbeitsplatz. Wer mit dem Gedanken spielt, seinen Computer im Wohnzimmer einzuquartieren, der sei gewarnt. Zu leicht vergißt man bei der Standplatzwahl den Platz für das Zubehör, wie etwa die Maus, den Drucker, das externe Laufwerk oder die Festplatte, miteinzubeziehen.

Außerdem ist mit dem Wohnzimmer als Computer-Arbeits-

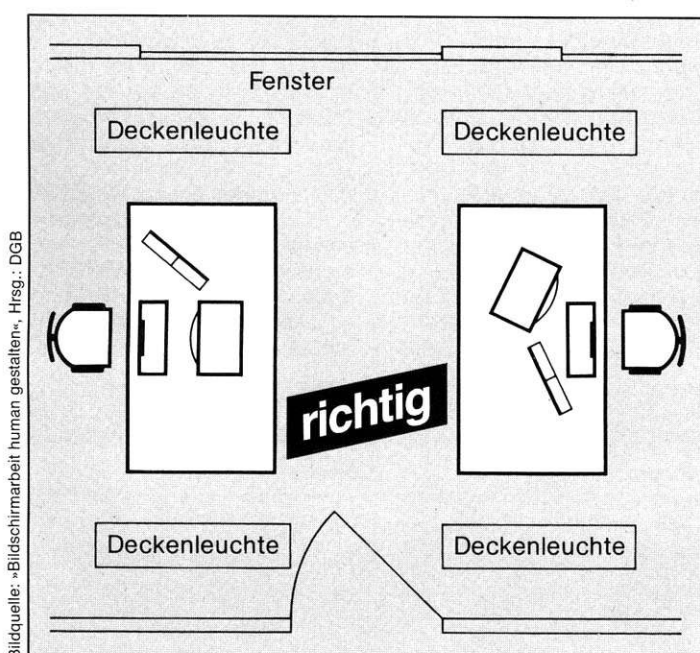


Bild 2. Hier ist der Standort durchdacht

Bildquelle: »Bildschirmarbeit human gestalten«, Hrsg.: DGB

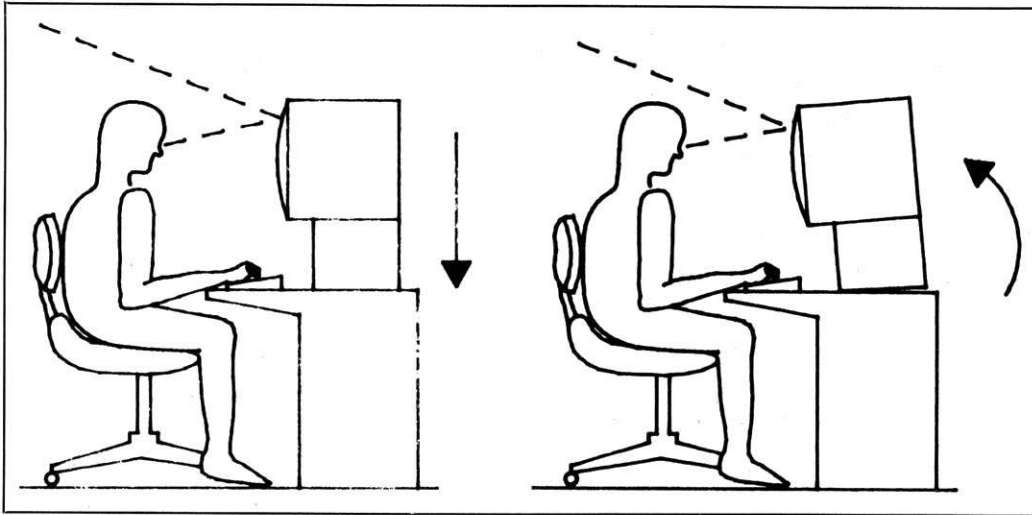


Bild 3. Bei Spiegelungen hilft meist schon leichtes Neigen oder Senken des Monitors

erfassung, Textverarbeitung oder das »Spielen« gelten verschiedene Merkmale.

Die Arbeitstischhöhe muß dem ergonomischen Erkenntnisstand entsprechen. Selbst hat man auf den Arbeitstisch wenig Einflußmöglichkeiten, es sei denn, man nennt einen höhenverstellbaren Tisch sein Eigen. Bei einem eventuellen Neukauf eines Arbeitstisches sollte man deshalb auf ergonomisch richtige Maße achten!

Steht nun endlich der Monitor auf dem Arbeitstisch an seinem richtigen Platz (was immerhin schon ein Riesenfortschritt ist), so stellt sich die Frage: Sitze ich an diesem Wunderwerk der Technik eigentlich richtig? Denn schließlich werde ich einige Stunden in dieser Körperstellung verbringen. Nach umfangreichen Recherchen kann ich erklären, wie eine gute Sitzhaltung aussieht:

Eine ergonomisch einwandfreie Haltung ist dann erreicht, wenn Ober- und Unterschenkel einen Winkel von 90 Grad oder etwas mehr bilden. Dabei sollen die Füße auf dem Boden oder einer evtl. vorhandenen Fußstütze (ist bei stetiger Arbeit zu empfehlen) stehen. Tunlichst vermieden werden sollte, die Oberschenkel zwischen Sitzfläche und Arbeitstisch einzuklemmen. Ist auch furchtbar unbequem. (Eigene schmerzliche Erfahrung!)

- Auch die Ober- und Unterarme sollen beim Arbeiten einen Winkel von ca. 90 Grad bilden und die Handhaltung möglichst gestreckt sein.

Diese entkrampfte Haltung läßt sich natürlich bei einem tobenenden Dauerfeuer gegen angreifende Zylonenkampfschiffe nicht durchhalten. (Dies wäre schlichtweg Selbstmord!)

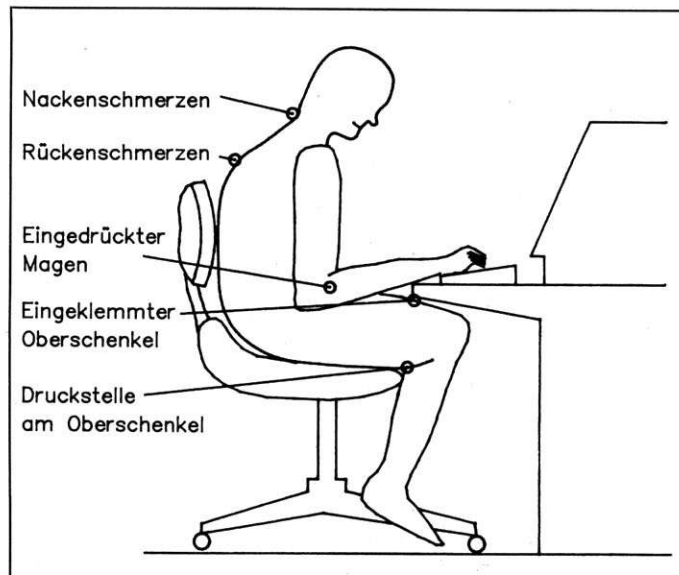


Bild 4. So werden Sie sicher ein Fall für die Krankenkasse

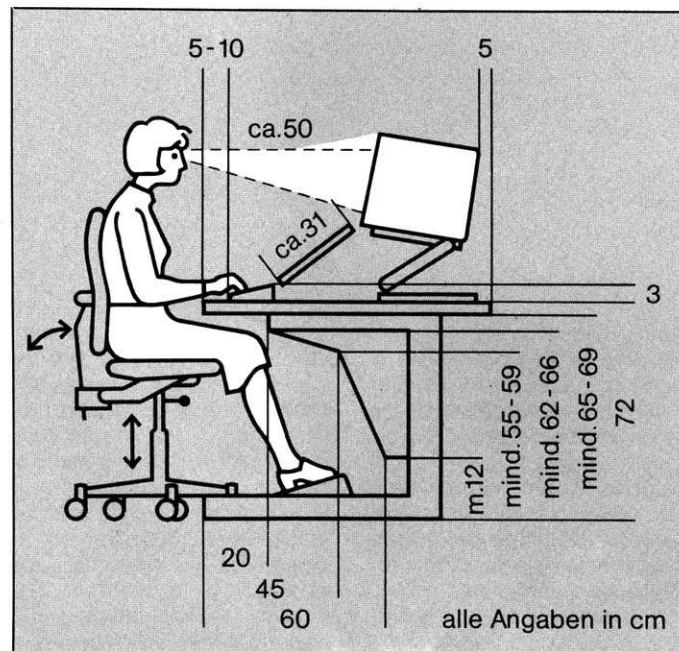


Bild 5. Ein nach neuen Erkenntnissen perfekter Arbeitsplatz

- Der Kopf sollte nur leicht nach unten geneigt sein und die Bildschirmoberkante bei dieser Körperhaltung in Augenhöhe oder etwas darunter liegen. Um diesen Anforderungen ohne größere Verrenkungen gerecht zu werden, ist ein individuell verstellbarer Bürostuhl unumgänglich. An diesem Arbeitsplatz soll nicht gelümmelt, sondern eine entspannte Sitzhaltung eingenommen werden.

Durch schmerzliche Erfahrungen zum »Spitzenergonomen« (selbst)ernannt, ist es mir ein leichtes, das Mysterium um die ergonomisch richtige Sitzhaltung weiter zu erläutern:

- Der Anwender soll die ganze Sitzfläche des Stuhles nutzen und niemals nur auf der Vorderkante sitzen.

- Die Sitztiefe so einstellen, daß zwischen Kniekehle und Sitzfläche eine Handbreit Luft bleibt.

- Die Oberkante der Rückenlehne soll bis zur Mitte des Schulterblattes reichen.

- Die Rückenstützenwölbung soll die Wirbelsäule an deren tiefster Einbuchtung abstützen.

Durch diese Haltung können Durchblutungsstörungen der Muskulatur vermieden werden. Nicht selten werden bewußt oder auch unbewußt schiefe - abnorme - Körperhaltungen gewählt, um beispielsweise Reflexionen im Blickfeld zu vermeiden. Schiefe Körperhaltungen erhöhen jedoch einseitig den Druck am Rand der elastischen Bandscheibe, ganz gleich, ob an der Hals-, Brust- oder Lendenwirbelsäule. Bei chronischer (anhaltender) Fehlbelastung degenerieren die Bandscheiben, die Beweglichkeit der Wirbelsäule wird gestört, muskuläre Verspannungen treten auf und Rückenschmerzen können die Folge sein.

Wie man sieht, ist das Hobby Computer nicht so einfach wie es aussieht. Auch ich mußte erst Erfahrungen sammeln, bevor ein gewisser Umdenkprozeß eingeleitet wurde. Heute sitze ich ergonomisch durchgestylt an einem ruhigen, bequemen und funktionellen Computerarbeitsplatz. Zwar meilenweit von jeglichen Errungenschaften der Zivilisation wie Kühlschrank, Radio und Steckdose entfernt, aber das läßt mich die Faszination des Amiga total vergessen. *pe*

Brigitte Bobenstetter ist schon seit einigen Jahren geschätztes Mitglied des M & T Verlags und leitet die Assistenz der Sonderhefte. Sie können sie erreichen über unsere Verlagsanschrift (Markt & Technik Verlag AG, Redaktion Sonderhefte, z.Hd. Frau Bobenstetter, Hans-Pinsel-Straße 2, 8013 Haar bei München).

Fast wie Weihnachten

von Brigitte Bobenstetter

Der Amiga 500 steht, aufs Vorzüglichste verpackt, zu Hause auf dem Teppichboden und harret der Dinge, die da kommen. Mit strahlender Miene und leicht unregelmäßigen Herzschlägen blicke ich voller Stolz auf meine allerneueste Errungenschaft. Endlich ist er da!

Schon seit längerem reife in mir der Wunsch, einen Computer in meinen leicht chaotischen 2-Personen-Haushalt mit aufzunehmen. Dieser technische Freund muß mir beim Bewältigen meiner umfangreichen Korrespondenz kräftig unter die Arme greifen. Außerdem werde ich sämtliche Ein- und Ausgaben meines Haushaltes über ihn laufen lassen, denn ich verliere leider langsam den Überblick. Deshalb setze ich große Hoffnungen auf dieses technische Wunderwerk. Der Verkäufer sagte, für Computerneulinge, die mit Ihrem Computer außer spielen noch mehr anfangen wollen, und das zu einem günstigen Preis, sei der Amiga 500 wie maßgeschneidert. Mit seinen Bits und Bytes und der passenden Software (Textverarbeitungs- und Dateiverwaltungsprogramme) will ich meinen Haushalt organisieren.

Momentan herrscht allerdings ein fast unbeschreibliches Auspackchaos. Eine Szenerie, die Ihnen bestimmt von Weihnachten bekannt vorkommt. Mit Taschenmesser und Schere versuche ich, meinen Helfer zu befreien, um ihm von Angesicht zu Angesicht gegenüberzustehen. Doch zuvor muß ich mich durch einen Berg von Styropor, Folien und Kartons durcharbeiten. Es blühe die Verpackungsindustrie! Natürlich kann ich eine gewisse Ungeduld nicht verleugnen, aber zugleich muß ich auch feststellen, daß mit übereilten Wegwerfaktionen auch leicht einmal ein dringend benötigtes Kabel verschwinden kann.

Wenn dann alles an Karton, Papier und Styropor entfernt ist, steht endlich alles vor mir, der Amiga, der dunkel glänzende Amiga-Monitor 1084S, diverse Kabel und zwei graue kleine Kästchen. Das größere graue

Schnell runter mit der Verpackung, die Kabel angestöpselt, Power an und los geht's. Doch oft geht dann nichts mehr, weil Sie irgendeine Kleinigkeit vergessen haben. Wir zeigen Ihnen, wie Sie es richtig machen.

Kästchen entpuppt sich als Transformator. Ohne seiner Spannung können die Bauteile im Amiga 500 nicht arbeiten. Das kleinere, elegante, graue Kästchen ist ein Eingabegerät, mit dem der Amiga bedient werden kann. Dieses Eingabegerät hat den merkwürdigen Namen Maus.

Somit ist die Grundausrüstung komplett. Nun stelle ich den Computer an den sorgfältig ausgewählten Platz und versuche, erste Kontakte zu knüpfen. Schritt für Schritt müssen jetzt

Amiga. Dazu wird das graue Kabel der Maus in den Steckeranschluß »Joystick 1« (Bild 1) an der Rückseite des Amiga gesteckt. Wunderbar, schon beschnuppern sich Maus und Amiga.

Als nächstes bin ich dafür, daß mein Amiga auch etwas sehen soll. Beim Kauf meiner Grundausrüstung habe ich mich zur Anschaffung eines

Amiga-1084S-RGB-Monitors entschlossen. Dieser Rot-Grün-Blau-Monitor ist ein Farbmonitor, der speziell für die Amiga-

»Inp. RGB« (Bild 2). Das Audio-Kabel ist schwarz und hat an beiden Enden jeweils zwei Stecker in den Farben Rot und Weiß. Drücken Sie den roten Stecker hinten am Amiga in die Buchse mit der Aufschrift »R Audio« und beim Monitor in »Audio-R«. Mit dem weißen Stecker verfahren Sie ebenso, nur daß die Buchse am Amiga hier »L Audio« heißt und am Monitor »Audio-L«. Zuletzt schließe ich den Monitor mit dem dünnen schwarzen Kabel an das Stromnetz an. Eine Verbindung zwischen Monitor, Amiga und Maus ist entstanden.

Momentan ist der Amiga noch vollkommen kraft- und saftlos und harret der Dinge, die da kommen. Als letztes und wichtigstes schließe ich nun den Amiga an das Stromnetz an. Dazu verbinde ich den

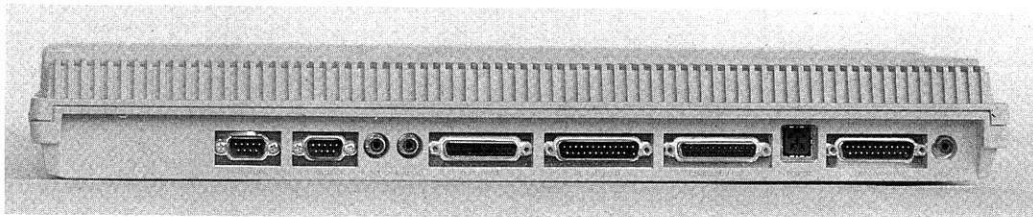


Bild 1. Die Schnittstelle zur Außenwelt des Amigas ist sehr umfangreich

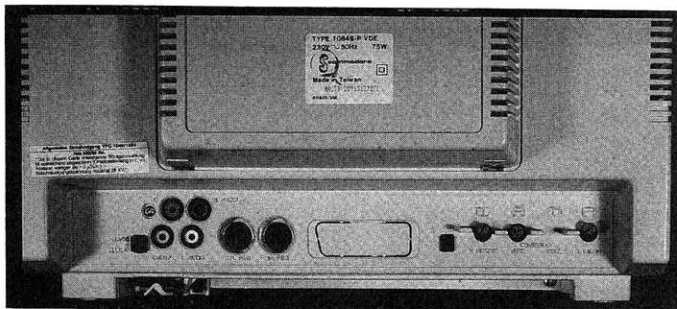


Bild 2. Die Rückseite des 1084S-Monitors

sämtliche Geräte miteinander verbunden werden. Allein der Amiga bietet hinreichend viele verschiedene Möglichkeiten, irgendein Kabel anzuschließen. Jetzt ist es an der Zeit, einmal einen etwas längeren Blick in das beiliegende Amiga-Handbuch zu werfen. Schon nach wenigen Einleitungssätzen geht es ausführlich auf das »Installieren« des Amigas nebst seiner »Systembestandteile« (Amiga 500, Monitor, Maus, Netzteil) ein. Nach kurzer Lektüre bin ich jetzt zu allem bereit. Zuerst verbinde ich die Maus mit dem

Serie entwickelt wurde. Er läßt sich aber auch an fast alle anderen Computer anschließen. Für den Amiga brauche ich zwei Kabel, jenes für die Bildübertragung ist am einen Ende länglich flach und am anderen Rund. Der längliche Anschluß wird in die Buchse auf der Rückseite des Amiga mit der Aufschrift »RGB-Video« (Bild 1) gesteckt und wenn er sitzt, mit den beiden Schrauben fest montiert. Das andere Ende kommt auf der Rückseite des Monitors in den runden Anschluß mit der Beschriftung

Netzstecker des Transformators mit dem Stromnetz und schiebe das andere Kabel in die Buchse mit der Aufschrift »Power« auf der Amiga-Rückseite. Fertig, alle Geräte sind nun miteinander verbunden und können Kraft aus der Steckdose tanken. Die Spannung steigt.

Der Monitor darf als erster ans Netz, dazu drücke ich rechts unten auf den Knopf. Der Monitor reagiert sofort darauf und ist betriebsbereit. Nun schlägt die große Stunde des Amiga 500. Der Kippschalter am Transformator wird gedrückt – und welch eine Freude, rechts am Amiga leuchtet mit kräftiger roter Farbe das POWER-Lämpchen auf.

Ich habe es wirklich ohne fremde Hilfe geschafft. Meine ersten Schritte auf dem Amiga können beginnen. *pe*

Brigitte Bobenstetter ist schon seit einigen Jahren geschätztes Mitglied des M & T Verlags und leitet die Assistenz der Sonderhefte. Sie erreichen sie über unsere Verlagsanschrift (Markt & Technik Verlag AG, Redaktion Sonderhefte, z.Hd. Frau Bobenstetter, Hans-Pinsel-Straße 2, 8013 Haar bei München).

Jetzt geht's los. Endlich will ich sehen, was ich für mein Geld bekommen habe und ob sich die Mühen der Platzwahl und des Anschließens gelohnt haben.

Wir sitzen uns endlich gegenüber. Hoffentlich verstehen wir uns gut. Einiges haben wir beide ja schon zusammen durchgestanden.

Wenn ich nur an das Chaos denke, das beim Auspacken herrschte, oder an meine Verzweiflung angesichts der mir unbekannten Geräte und des schier unüberwindbaren Kabelsalates. Doch für mich stellte das kein Problem dar. Souverän umschiffte ich all diese Klippen dank meiner Cleverneß und des Handbuchs.

Jetzt wird es aber ernst. Die Weichen sind gestellt, nichts kann mich abhalten, den Weg durch die geheimnisvolle Welt der Computerchips zu entdecken. Mit zwei Handgriffen steht der Amiga bei Fuß. Auffordernd blinzelt er mir mit seiner Power-LED-Anzeige zu. Nur Mut signalisiert er mir. Einmal tief eingeatmet gehe ich meine ersten Schritte auf dem Amiga entgegen. Mit dem Handbuch auf dem Schoß und genauen Anweisungen durch den Computer beginnt der Start eigentlich ganz locker.

Beim Einschalten erscheint auf dem Bildschirm eine Hand, welche die Workbench-Diskette hält. Ich sehe dies als Aufforderung an und greife meinerseits sofort zur Workbench-Diskette und schiebe sie ins Laufwerk hinein. Geht nicht! Logisch, die Diskette muß mit der metallenen Seite voran und dem Etikett nach oben in das Laufwerk geschoben werden. Und nicht verkehrt herum, wie ich das versucht habe.

Gut, es ist noch kein Meister vom Himmel gefallen. Weiter geht es in meiner Lektion: Erste Schritte auf dem Amiga. Wie ich im Handbuch nachlese, liest der Amiga die Informationen auf der eingelegten Workbench-Diskette. Deshalb leuchtet auch die grüne Kontrollleuchte rechts oben an der Tastatur auf. Ein ganz wichtiger Hinweis zu dieser Laufwerkskontrolllampe:

Computererlebnisse

Countdown

Niemals eine Diskette bei leuchtender Diode entfernen!

Nach etwa 40 Sekunden erscheint ein neues Bild auf dem Monitor. Viel ist auf dem Bildschirm nicht zu sehen, aber oben steht in einem weißen Balken »Workbench release xxxx free memory«, oh, rechts ist auch noch eine Diskette zu sehen. Nach kurzem Studium des Handbuchs erfahre ich, daß der Balken eine sogenannte Titel-Leiste ist. Die Zahl vor free memory ist der zu Verfügung stehende Speicherplatz.

Das Diskettensymbol zeigt an, daß sich im Laufwerk momentan die Diskette »Workbench 1.3D« befindet. Solche Zeichen nennt man Piktogramme. Überhaupt spricht man jetzt nicht mehr von dem Bild auf dem Monitor, sondern ganz profimäßig vom »Arbeits-tisch«.

halte. Danach ist dieses Problem erledigt. Der Pfeil kann aber auch über die Tastatur bewegt werden. Das geht so: Eine der beiden AMIGA-Tasten rechts

Piktogramme gefällig?

oder links an der Tastatur gedrückt halten und die Richtung mit den grauen Cursor-Tasten rechts bestimmen. Das ist sehr einfach, aber wesentlich langsamer und langweiliger. Mir macht es mit der Maus mehr Spaß. Jetzt interessiert mich aber, was es mit den beiden Tasten an der Maus auf sich hat. Wie ich im Handbuch nachlese, werden mit der rechten Taste Menüs ausgewählt, darum heißt diese Taste auch Menü-taste. Aha! Mit der linken Taste

schreibt: Für die meisten Hilfsmittel oder Programme einschließlich des Arbeitstisches stellt der Amiga Menüs zur Verfügung, in denen alle Funktionen aufgelistet werden, die ein bestimmtes Programm enthält. Alles kapiert? Nein!

Auch nicht schlimm, ich jedenfalls werde jetzt einmal so ein Menü anwählen. Durch die Maus bewegt, wandert mein Pfeil zum Workbench-Piktogramm, dort angekommen drücke ich die Auswahl-taste (linke Taste) »Klick«. Dann drücke ich die rechte Maustaste. Zu meiner Freude tut sich da etwas. Langsam dämmert es mir: Ich habe soeben einige Funktionen des Programms Workbench aufgelistet. Leider verschwindet diese Auflistung gleich wieder. Ein kurzes Nachlesen im Handbuch, und schon ist ein weiterer Fehler ausge-merzt. Wenn ich ein Menü öff-ne, muß ich die Menü-taste (rechte Taste) gedrückt halten. Das wird einige Male probiert und als erfolgreich abgehakt. Die Funktion »OPEN« hat es

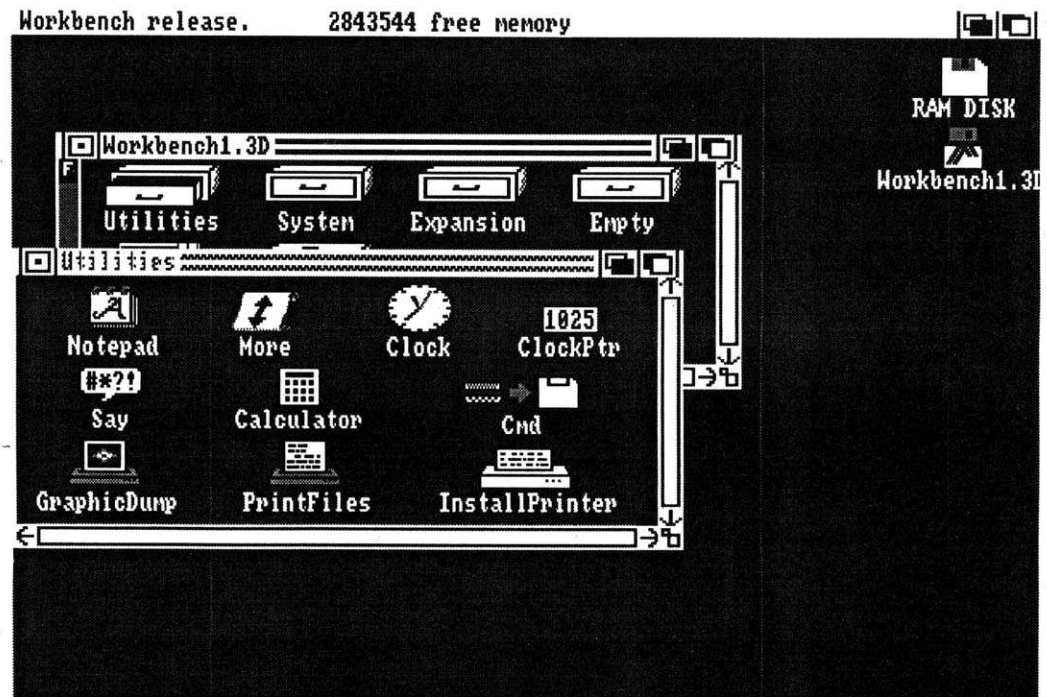


Bild 1. Die Workbench ist voller Überraschungen in ihren Fenstern

Auf diesem Arbeitstisch erkenne ich jetzt auch einen kleinen Pfeil. Wie putzig! Im Handbuch steht, daß ich den Pfeil mit der Maus bewegen kann. Der Pfeil bewegt sich wie von Zauberhand ganz nach meinem Willen. Zwar noch etwas holprig und ungelenkig, aber immer dahin, wo ich will.

Die Tasten an der Maus stören mich nur so lange, bis ich nachlese, daß ich sie falschem

können Piktogramme ange-wählt werden. Diese Taste heißt dann Auswahl-taste. Gut, noch einmal zusammengefaßt. Rechte Taste ist gleich Menü-taste und wählt Menü-Funktionen aus, die linke Taste heißt Aus-wahl-taste und wählt Piktogram-me an. Alles paletti. Piktogram-me kenne ich mittlerweile schon (Workbench-Piktogramm), aber was haben Menüs im Compu-ter zu suchen? Das Handbuch

mir angetan. Etwas zu öffnen ist immer mit einer gewissen Spannung und Erwartung verbunden. So auch hier. Auf dem Arbeitstisch erscheint ein völlig neues Bild. Im Handbuch steht, ein Arbeitstisch-Fenster wurde geöffnet. Auch gut. Dieser Kasten mit den verschiedenen Piktogrammen ist also ein Fenster. Herrlich anzuschauen, was sich in diesem Fenster alles an Verzierungen befindet.

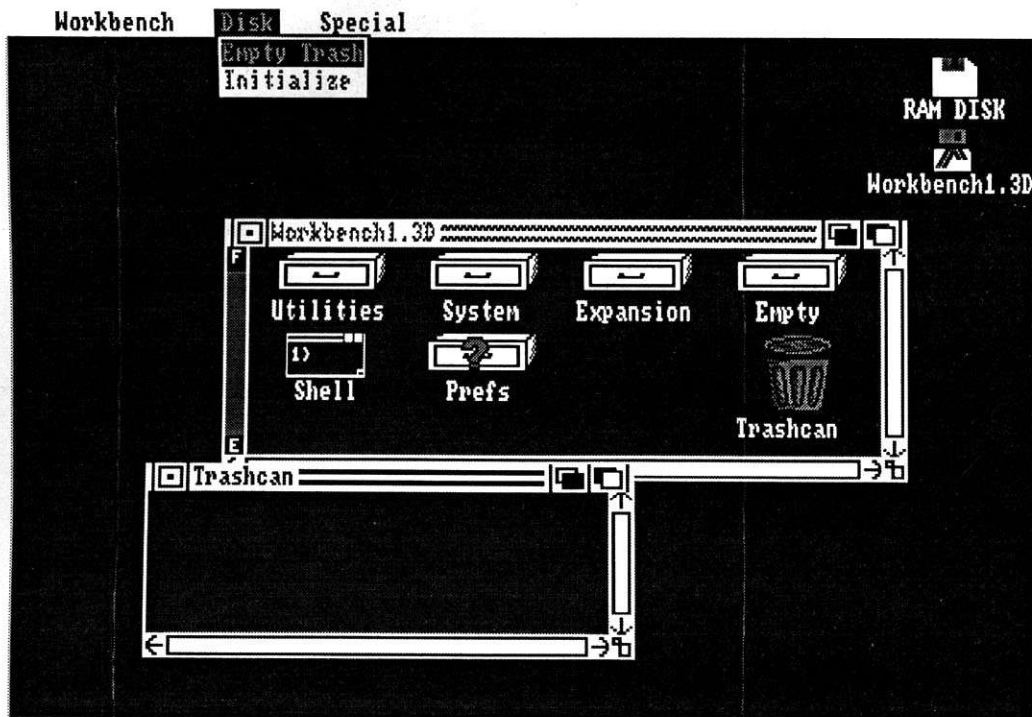


Bild 2. Die Mülltonne (Trashcan) muß von Hand geleert werden (Empty Trash)

Eine Uhr, ein Fragezeichen, verschiedene Schubladen und eine uralte Mülltonne. Ich sage mir: Nicht aufregen, nur wundern! Das Handbuch sagt, diese Piktogramme repräsentieren den Inhalt der Workbench-Diskette. Da kann ja einiges auf mich zukommen. Was bitte hat eine Mülltonne im Computer zu suchen? Dem muß auf den Grund gegangen werden.

Ich klicke das Symbol mit der Auswahlstaste an und warte. Ein

weiteres Fenster öffnet sich. Hilfe, jetzt habe ich plötzlich zwei Fenster übereinander auf dem Arbeitstisch. Im Hintergrund kann ich das erste Arbeitstisch-Fenster noch erkennen und darüber liegt nun das neue Fenster mit der Mülltonne. Trotzdem weiß ich mit der Mülltonne nichts anzufangen. Ein kurzes Studium des Handbuchs ist somit wieder einmal unumgänglich. Das Piktogramm ist keine Mülltonne, sondern ein

Papierkorb. In diesen Papierkorb kann ich z.B. einen Text verstauen, den ich momentan nicht benötige. Dort lagert der Text so lange unsichtbar, bis ich ihn wieder nutzen oder unwiderruflich aus meinem Computer werfen möchte. Momentan habe ich aber sowieso nichts, was ich in den Papierkorb werfen kann.

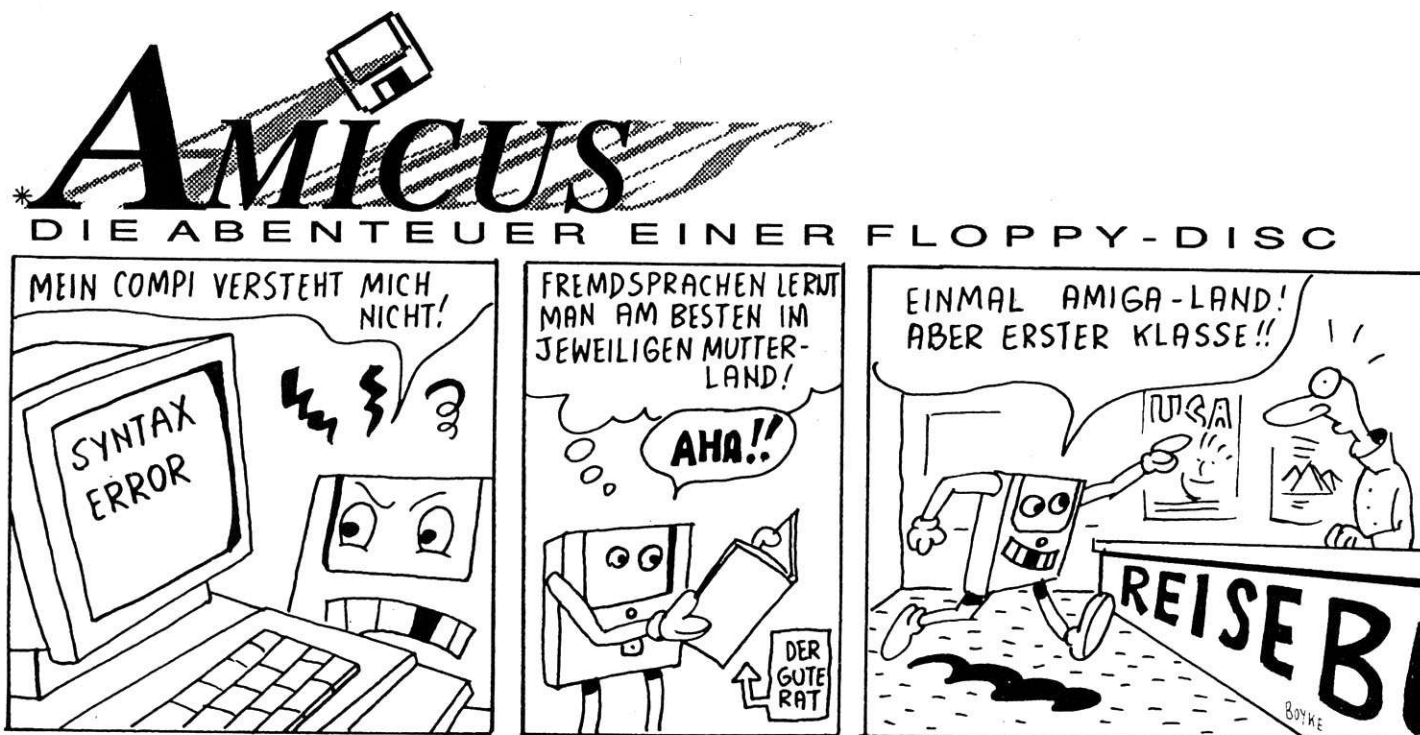
Aber nun stellt sich ein anderes Problem: Wie kann ich dieses Fenster wieder schließen.

Nach endlosem Blättern im Handbuch konnte ich auch dieses Problem lösen. Jedes Fenster besitzt links oben ein sogenanntes Schließ-Symbol, welches ein viereckiges Kästchen mit schwarzem Punkt darstellt. Ein kurzes Anklicken mit der Auswahlstaste und schon ist das Fenster verschwunden.

Und überhaupt will ich jetzt aus dieser Workbench raus. Gesagt, getan: Mit der Menüstaste wähle ich oben links die Workbench an und beende das Programm mit der Funktion »close« aus dem Project Menü. Schon habe ich den Weg nach »draußen« gefunden.

Als nächstes lege ich die Diskette »Erste Schritte auf dem AMIGA« ein. Diese Diskette befindet sich standardmäßig bei jeder Grundausstattung. Voller Spannung warte ich auf die ersten Lektionen. Zu meinem Erstaunen mußte ich feststellen, daß ich das alles schon beherrsche. Wer weiß denn nicht, wie man eine Maus hält? Auch sind mir die Funktionen der beiden Maustasten schon längst bekannt. Ich sehe schon, diese Diskette ist nur was für Anfänger, und das sind wir nun ja nicht mehr – oder? *pe*

Brigitte Bobenstetter ist schon seit einigen Jahren geschätztes Mitglied des M & T Verlags und leitet die Assistenz der Sonderhefte. Sie können sie über unsere Verlagsanschrift erreichen (Markt & Technik Verlag AG, Redaktion Sonderhefte, z.Hd. Frau Bobenstetter, Hans-Pinsel-Straße 2, 8013 Haar bei München).



Machen Sie mit!

~~A~~MIGA-SONDERHEFT 8

Diesen Beitrag im AMIGA-Sonderheft fand ich besonders gut:

Ausgabe: _____ / _____ Seite: _____

Artikel: _____

Ich wünsche mir für eine der nächsten Ausgaben folgende Themen:

**Ich möchte an der redaktionellen Gestaltung mitarbeiten.
Meine Vorschläge:**

Ich kann folgende(s) Programm(e) zur Veröffentlichung anbieten:

Dieses Problem habe ich:



Schreiben Sie uns!

Ich besitze einen

A500 ____ **A1000** ____ **A2000** ____ **A2500** ____

mit 1 ____ **2** ____ **3** ____ **mehr** ____ **Laufwerken**

mit einer Festplatte ____

Ich verwende einen Drucker ____

mit 9 Nadeln ____ **24 Nadeln** ____

Zusätzlich besitze ich einen

C 64 ____ **PC** ____ **Atari ST** ____ **andere** ____

Diese Note (1 bis 6, 1 am besten) gebe ich dem

AMIGA-Sonderheft: ____

Das sollte am AMIGA-Sonderheft besser werden:

Name: _____

Alter: _____ Jahre

Adresse: _____

Telefon: _____

Bitte schicken Sie die Mitmachkarte
in einem Briefumschlag
an folgende Adresse:

Markt & Technik Verlag AG
Redaktion Sonderhefte
Stichwort: Mitmachkarte Amiga
Hans-Pinsel-Straße 2
8013 Haar b. München



Das universelle Nachschlagewerk

Amiga von A bis Z

von David Twigg-Flesner

Abkürzungen und englische Fachbegriffe beherrschen die Welt der Computerexperten. Auch der Amiga mit seiner komplexen Funktionsweise wird oftmals nur in »Amiga-Sprache« beschrieben. Da ist von Animation, Libraries, Devices, Bitmaps, AbsExecBases oder IDCMP die Rede, was einem vor allem zu schaffen macht, wenn man der englischen Sprache nicht mächtig ist.

Der Grund für all diese merkwürdigen Fremdwörter und Abkürzungen, liegt an der Fachliteratur, die lange Zeit nur in Englisch erhältlich war. Notgedrungen bürgerten sich diese englischen Wörter im Sprachschatz der Amiga-Kenner ein. Zum Nachteil aller Einsteiger. Doch bevor Sie verzweifelt

Gerade als Einsteiger hat man es schwer, die Fachbegriffe, die man im Zusammenhang mit dem Amiga hört, zu verstehen, zumal es sich hier meist um englische Worte oder Wortneubildungen handelt. Unser Amiga-Lexikon wird Ihnen hier und da aus der Klemme helfen.

dicke Wörterbücher wälzen, verwenden Sie doch unser Lexikon. Es beschreibt in alphabetischer Reihenfolge nahezu alle wichtigen Begriffe zum Amiga. Selbstverständlich kann unsere Sammlung nicht so ausführlich wie ein Fachbuch sein. Für allgemeine Informationen ist es jedoch sehr nützlich. Obgleich wir uns um Vollständigkeit bemühten, kann es dennoch vorkommen, daß der eine oder andere Ausdruck nicht auffindbar ist. Handelt es sich

um zusammengesetzte Wörter, schlagen Sie in diesem Fall bitte die Einzelbegriffe nach. Sicherlich werden Sie dann fünfzig. Spezialausdrücke, die nur mit einem bestimmten Programm in Verbindung stehen oder etwa Programmnamen, konnten wir aus Platzgründen nicht in unser Lexikon aufnehmen. Sie würden angesichts des riesigen Softwareangebots für den Amiga ganze Bibliotheken füllen. Hier ist es sinnvoller, die entsprechenden Programm-

beschreibungen zu studieren. Wir haben versucht, thematische Zusammenhänge zwischen Begriffen mit Querverweisen (->) zu knüpfen, so daß Sie eine größtmögliche Informationsfülle zu dem nachgeschlagenen Thema oder Begriff erhalten.

Da derzeit zwei (beim A1000 drei) Betriebssysteme für den Amiga aktuell sind, wird gelegentlich zur Unterscheidung von Version 1.1, 1.2 oder 1.3 gesprochen. Da die drei Betriebssysteme weitgehend »aufwärtskompatibel« sind, das heißt Version 1.2 und 1.3 ähnlich funktionieren wie Version 1.1, und Version 1.2 konform mit Version 1.3 geht, gilt das für eine Version Beschriebene auch für die neueren Varianten des Amiga-Betriebssystems. Mögliche Unterschiede werden allerdings deutlich gemacht.

(Michael Thomas/so)

AbsExecBase

Die Speicherstelle 4, der Sprungvektor zur -> exec.library (also die Adresse der Library), ist eine der empfindlichen Speicheradressen des Amiga. Da Programme erst über sie den Standort der Systemroutinen erfahren, führt ein Überschreiben dieser Adresse zum Systemabsturz.

ACBM

(Amiga Contiguous Bitmap) Die »aneinandergrenzenden« Speicherbereiche der verschiedenen -> Bitplanes eines -> Bitmaps beim Amiga: Anders als beim -> IFF-Verfahren werden im Speicher des Amiga und bei den Dateien von Amiga-Basic die einzelnen Bitplanes getrennt der Reihe nach verwaltet bzw. gespeichert. Umwandlungsprogramme für die beiden Formate sind auf -> FishDisk zu finden.

AddMem

Ein Hilfsprogramm für die Einbindung von Speichererweiterungen unter Version 1.1; anders als -> auto.config-Erweiterungen überprüft das Programm nicht, ob der zugewiesene Speicherbereich bereits belegt ist.

AGNUS

Agnus koordiniert als Adreßgenerator-Chip den Speicherzugriff sowohl der CPU als auch der Sonderchips (-> Chip-RAM, -> Fast-RAM) und ist für die Steuerung der -> DMA-Kanäle verantwortlich. Der Chip enthält die beiden Hilfschips -> Blitter und -> Copper.

alert

Systemwarnung oder -> guru meditation Nummer.

aliasing

(auch »the jaggies«) Die stufenförmige Erscheinung bei nicht gerade senkrecht oder waagerecht verlaufenden Linien am Bildschirm.

aliasing distortion

-> low pass filter, -> quantization.

ALink

Ein -> Link, der zum Zusammenbinden einzelner Codemodule (-> object code) zu einem lauffähigen Programm (-> load file) verwendet wird; war Teil der frühen -> developers kits; ist mittlerweile von der verbesserten Version, BLink (FishDisk), abgelöst worden.

Amiga-DOS

(Disk Operating System) Die Bezeichnung der Laufwerk-s, Dateiverwaltungs- und Multitaskingroutinen (-> process) des Amiga; diese wurden von der englischen Firma -> Metacomco aus Bristol in Anlehnung an das System »Tripos« geschrieben und sind weitgehend Unix-kompatibel. Die meisten Routinen sind in der Programmiersprache -> BCPL geschrieben. Die Grundroutinen von Amiga-DOS sind über die entsprechenden Befehle des -> CLI dem Benutzer zugänglich (siehe operating system).

AmigaLibDisk

-> Fish-Disks.

Amplitude modulation

(AM) Die (fortlaufende) Veränderung der Höhe (Amplitude) eines Signals (einer Wellenform); das heißt die Lautstärke des Signals wird geändert. Dieses Verfahren wird beim Amiga benutzt, um beispielsweise die Intonationskurve bei der Sprachsynthese zu erzeugen; es kann auch für die Programmierungen von -> Hüllkurven eingesetzt werden. Das Verfahren wird beim Amiga mit -> attached channels realisiert (siehe fre-

quency modulation) und ist von Amiga-Basic aus nicht ohne weiteres möglich.

Animation

Eine besondere Stärke der Systemroutinen des Amiga ist die Animation, die in zwei verschiedenen Formen vorhanden ist: Sprite-Animation und Playfield-Animation (BOBs). Unter Animation versteht man die Bewegung eines sich selbst in unterschiedlichen Bewegungsphasen befindlichen Objektes am Bildschirm (Beispiel: gehende Figur).

Der Amiga verfügt hierzu über acht Hardware-Sprites und eine beliebige Anzahl BOBs; anstelle der Hardware-Sprites erscheinen aber in der Regel sogenannte VSprites (virtual sprites = Scheinsprites). Hierbei handelt es sich um verschiedene grafische Formen eines mehrfach verwendeten Hardware-Sprites. Diese Mehrfachverwendung wird durch die Zuweisung eines eigenen -> DMA-Kanals zu jedem Hardware-Sprite ermöglicht; einzige Voraussetzung für die Mehrfachverwendung ist, daß die verschiedenen Formen des Sprites mindestens eine Bild-

schirmzeile Abstand voneinander haben müssen.

Ein BOB (Blitter Object) ist ein Teil des normalen »Bildes«, das durch den -> Blitter sehr schnell von einer Position in eine andere »kopiert« werden kann.

Die Playfield-Animation liegt auch in zwei Formen vor; man kann mit den von Zeichentrickfilmen bekannten Verfahren der »cel animation« (-> sequenced drawing: Eine Folge von Einzelbildern, die ursprünglich auf Papier, später auf Zelluloid, gezeichnet wurden) arbeiten oder mit -> motion control die Geschwindigkeit und Beschleunigung eines BOBs oder Sprites in vertikaler und horizontaler Richtung steuern.

Die Animationsroutinen sind von Amiga-Basic über die entsprechenden Befehle erreichbar (siehe graphics primitives).

AnimComp / AnimObj -> GELS.

ANSI X3.64-1979

(American National Standards Institute) Eine normgebende Institution der Vereinigten Staaten, die überwiegend Hardware-Normen aufstellt. X3.64-1979 ist eine Normierung der SteuerCodes für -> Terminals und bildet die Grundlage der vom Amiga benutzten Tastatur- und SteuerCodes; ergänzt werden die in der -> ISO / DIS 6429.2-Version implementierten Codes durch -> DEC-ähnliche und Amiga-spezifische Steuersequenzen.

applications program

(Anwendungsprogramm) Jede Art von Programm, mit dem der Benutzer Daten verarbeiten kann (siehe utility).

argument template

(Befehlszeilenmuster) Eine Anzeige der bei einem (meistens -> CLI-) Befehl erforderlichen Parameter sowie der Befehlssyntax am Bildschirm; eine sogenannte »Help-Funktion«, die bei fast allen von Metacomco programmierten CLI-Befehlen durch Eingabe des Befehls mit einem Fragezeichen abgerufen werden kann. Bei Ergänzungsbefehlen aus anderen Quellen kann diese Funktion fehlen.

artifacting -> hue.

ASCII

(American Standard Code for Information Interchange) Eine standardisierte Codierung für Zeichensätze und Steuerzei-

chen (A=65,B=66...), die 1963 erstmals eingeführt wurde; mit diesem 7-Bit-Code ohne Paritätsvorschriften sind 128 Zeichen darstellbar und auch standardmäßig definiert. Das achte Bit wird in der Regel benutzt, um systemspezifische Zeichen zu codieren, wie das auch beim Amiga der Fall ist.

asynchronous

(nicht zeitgleich) Wörtlich verstanden müßte diese Bezeichnung bedeuten, daß der Computer Befehle, Routinen usw. »nacheinander« ausführt, das heißt eine laufende Routine muß zuerst beendet sein, bevor die nächste Routine beginnen darf (siehe synchronous).

Im -> RKM wird diese Bezeichnung anscheinend anders ausgelegt, wodurch die Bedeutung ins Gegenteil umgekehrt wird, wie bei der Erläuterung des Befehls SAY im Amiga-Basic-Handbuch.

ATOM

(ALink Temporary Object Modifier) Eine Hilfsroutine für Assembler, die dazu dient, -> Objectcode-Module neu zu linken, damit der erzeugte Code auf -> Chip-RAM (Daten) und -> Fast-RAM (Code) verteilt werden kann, was bei Erweiterungen über 512 KByte hinaus vor allem bei speicherintensiven Grafikanwendungen erforderlich sein kann. Um bereits gelinkte Programme zu modifizieren, müssen diese zuerst mit einem Hilfsprogramm wieder in nichtgelinkte Objectcode-Module zurückverwandelt und dann mit ATOM für einen neuen Linkvorgang vorbereitet werden.

attached channels

(gekoppelte Audiokanäle) Eine Besonderheit des -> audio.device besteht darin, die vier Kanäle paarweise (0/1, 1/2, 2/3) zu koppeln, um den höherwertigen (Daten-)Kanal durch den niederwertigen Kanal entweder in Amplitude oder Frequenz zu modulieren. Der Modulationsmodus wird durch Eingriff in die Hardware-Register bestimmt; das ADKCON-Register (&hDEF09E) enthält in den untersten 8 Bit die entsprechende Steuerung. Die Bits 0 bis 3 schalten die Amplituden-, 4 bis 7 die Frequenzmodulation des entsprechenden Kanals ein. Die Modulationsdaten werden für beide Modi als 16-Bit-Werte über den gesetzten Kanal verarbeitet, obwohl bei der Amplitudenmodulation nur die untersten 7 Bit gültig sind, weil die maximale

Lautstärke 64 beträgt. Diese Festlegung auf 16 Bit ist notwendig, weil beide Modi gleichzeitig aktiv sein dürfen; in diesem Fall werden die Daten abwechselnd für die Modulation herangezogen. Bei Amiga-Basic ist dieses Verfahren nur durch POKEs erreichbar; dabei müssen die oberen 8 Bit des ADKCON-Registers unverändert bleiben, weil diese für das Laufwerk verwendet werden.

audio.device

Eine Systemroutine (-> device), die vier Audiokanäle verwaltet und steuert. Zur Tonerzeugung ist eine Wellenform erforderlich, die als Datenwerte zu je 8 Bit in einem Integerfeld abgelegt wird. Digitalisierte Daten können auch verwendet werden, solange die Formatbestimmungen für die Daten eingehalten werden. Das audio.device wird auch in der Endstufe der -> Sprachsynthese benutzt (siehe attached channels). Über Amiga-Basic ist das audio.device mit den Befehlen SOUND und WAVE erreichbar (siehe sound synthesis).

auto.config

Automatic configuration / automatische Einbindung von Zusatzgeräten wie Festplatten-Laufwerke und Speichererweiterungen in das System: Solche Zusätze werden mit der -> Startup-Sequence, gegebenenfalls mit einem in der -> Expansion-Schublade abgelegten Treiber beim -> Booten des Systems angemeldet und erkannt. Für solche Geräte hat Commodore ein System von Erkennungsbytes eingeführt, um zu verhindern, daß ein gegebener Speicherbereich von mehreren Zusätzen gleichzeitig beansprucht werden kann. Zum Einbinden stehen insgesamt 128 Abschnitte (-> expansion slot) mit jeweils 4 KByte am oberen Ende des Speicherbereichs zur Verfügung; benötigt ein Zusatz mehr als diese 4 KByte, wie das beispielsweise bei einer über einen -> DMA-Kanal betriebenen Festplatte der Fall sein dürfte, so muß der für die normale Speichererweiterung vorgesehene Bereich benutzt werden (siehe Zorro).

backdrop

(Hintergrundfenster) (-> layers) Eine besondere Art -> Window, bzw. eine »Schicht« des Bildschirms, die stets die hinterste aller anderen »Schichten« bildet und die Funktion eines Hintergrunds bei verschiedenen anderen Programmen

übernimmt (Beispiel: Workbench-Fenster). Als einziges -> Gadget besitzt ein solches Fenster das Schließ-Gadget. Ein »backdrop« ist von Amiga-Basic aus nicht definierbar.

backup

Sicherheitskopie einer Diskette, Festplatte oder einer Datei.

backward referencing

-> forward referencing.

base

(Basisadresse) Die Einsprungsadresse einer -> Library, entweder im RAM-Speicher (bei diskettenresidenten) oder im ROM (-> WCS) (bei systemresidenten) Libraries (siehe AbsExecBase). In Amiga-Basic wird diese Adresse mit dem LIBRARY-Befehl ermittelt, damit die Libraryfunktionen abgerufen werden können.

batch file

(auch command file, script, sequence file - Befehlslistendatei) Eine Textdatei, die -> CLI-Befehle enthält und mit dem EXECUTE-Befehl wie ein normaler CLI-Befehl abgerufen und ausgeführt wird (Beispiel: -> startup-sequence). Wie diese Datei werden »batch files« am besten im s-Verzeichnis der Diskette abgelegt, weil EXECUTE sie dort zuerst, danach erst im aktuellen Verzeichnis sucht.

BCPL

(Binary Coded Programming Language) Ein Vorgänger der Programmiersprache C und daher auch mit C weitestgehend kompatibel; im Unterschied zu C kennt BCPL nur einen Datentyp, das binäre Wort und ist daher zur Programmierung von Computersystemen, beispielsweise des -> Amiga-DOS, gut geeignet. Viele Bereiche von AmigaDOS, einschließlich der CLI-Befehle, sind nicht in C, sondern BCPL geschrieben. Ein entsprechender Compiler von Metacomco ist bereits erschienen.

Bitmap

Bildschirmspeicher, in dem jedes Bit einem Punkt auf dem Bildschirm zugeordnet ist. Beim Amiga heißen die Bitmaps aber -> Bitplanes, weil das Bild »mehrschichtig« aufgebaut wird. In der Sprache C ist »Bitmap« ein Zeiger auf eine Bitplane-Struktur.

Bitmap Display

Das »Zeichenblatt« für den Bildschirm; kann aus bis zu acht -> Bitplanes bestehen. Davon werden aber derzeit ma-

ximal fünf (sechs im HAM-Modus) unterstützt.

Bitmap

Eine -> Bitmap für eine »Schicht« des Amiga-Bildschirms; die Anzahl der verwendeten Bitplanes bestimmt die Anzahl der verfügbaren Farben am Bildschirm (siehe colour registers). Die Bitplanes sind sehr speicherintensiv; eine Bitplane in niedrigauflösendem Modus braucht 8000 Byte; dieser Bedarf wird bei Hochauflösung oder -> Interlace verdoppelt, bei beiden gleichzeitig vervierfacht - ein Bildschirm mit Hochauflösung, Interlace und 16 Farben braucht also insgesamt 128 000 Byte und ist daher bei einem Amiga mit nur 256 KByte nicht möglich, denn es bleiben nur etwa 30 000 Byte für das Programm und die Daten. Bei dem PAL-Amiga mit 256/512 Bildschirmzeilen ist der Speicherbedarf entsprechend höher, und zwar um etwas über 25 Prozent (10 240 Byte pro Bitplane im niedrigauflösenden Modus).

Blitter

(auch Bimmer) Ein Hilfschip von -> Agnus, als BLock Image TransferER (Blitter) verantwortlich für Datenverschiebungen (Bildschirmbereiche wie Fenster, aber auch Daten und Programme) im -> Chip-RAM; als Bit Image Manipulator (Bimmer) kann Blitter über drei der eigenen -> DMA-Kanäle bis zu 256 logische Operationen (siehe minterm) an den Daten vornehmen, bevor sie über den vierten Kanal in den Speicher zurückgeschrieben werden. Außerdem zeichnet der Chip Linien mit einer Geschwindigkeit von 1 Million Punkte pro Sekunde.

Blitter Object

(BOB -> Animation) Teil eines -> Playfields, der von -> Blitter »animiert« wird; unterliegt in den Ausmaßen keinen Begrenzungen, wird aber möglicherweise im 32-Farben-Modus, bei gleichzeitiger Verwendung von -> Sprites, in der Farbenzahl eingeschränkt (siehe colour registers).

.bmap files

Besondere Dateien, die von Amiga-Basic für den LIBRARY-Befehl benötigt werden; diese Dateien werden mit dem Hilfsprogramm ConvertFD auf der Extras-Diskette aus den -> .fd-files erzeugt, damit Amiga-Basic die übergebenen Parameter bei Libraryaufrufen auf

ihre Richtigkeit überprüfen kann. Das Hilfsprogramm läuft, zumindest unter Version 1.1, nicht völlig richtig; enthält ein .fd-file Funktionsnamen oder Wörter, die mit Schlüsselwörtern (FOR, PRINT...) des Amiga-Basic identisch sind, werden diese als -> Token statt als ASCII-Kette verschlüsselt. Eine korrigierte Version des Programms befindet sich auf Fish-Disk #27. Zum Glück sind die meist benötigten .fd-files bereits in .bmap-Format auf der Extras 1.2-Diskette enthalten.

boot disk

Die Diskette, mit der das System initialisiert wird, also diejenige, die nach Anforderung der -> Workbench-Diskette eingelegt wird.

Boot-ROM

(Start-ROM) Beim Einschalten des Amiga 1000 übernimmt dieser ROM-Baustein die Überprüfung des verfügbaren Speichers oder anderer Bausteine und fordert die Eingabe der -> Kickstart-Diskette; beim A500 und A2000 sind die Kickstart-Routinen auch im ROM untergebracht und werden nach der Systemüberprüfung automatisch aktiviert.

boot(strap)

(wörtlich: Schnürsenkel) Die Initialisierungsroutinen des Computersystems; abgeleitet von der Wendung »to pull oneself up by one's bootstraps«, die am besten mit der Geschichte des Barons von Münchhausen verglichen werden kann, der sich am eigenen Zopf aus dem Sumpf zog. Beim Computer bedeutet das, daß der Computer mit einem Ladeprogramm sich selbst für die eigentliche Datenverarbeitung »initialisieren« muß; dies geschieht beim Amiga mit der Kickstart-Diskette (nur A1000) für das Betriebssystem selbst und mit einer »Workbench-Diskette« (-> system disk) für die Workbench. Da A500/A2000 den Kickstart im ROM haben, läuft die Initialisierung nach der Überprüfung des -> Boot-ROM ab.

boot sectors

(Startsektoren einer Diskette) Jede »selbststartende« Amiga-Diskette (Beispiel: Workbench) enthält diese beiden Sektoren, die das Einlesen der Disketten-daten (des Disk-Icons) ohne vorher eingelegte Workbench-Diskette erlauben. Bei frisch formatierten Disketten werden diese Sektoren mit dem -> CLI-Befehl INSTALL auf die Diskette geschrieben.

bridge board

(auch bridge card) Die Bezeichnung für Zusatzkarten zum Amiga 2000, mit denen der Computer IBM-kompatibel gemacht wird; so genannt, weil die Karte wie eine Brücke einen Amiga-Slot mit einem PC-Slot verbindet (siehe Janus, Zorro).

buffer

(Pufferspeicher) Ein Speicherbereich, der zwischen einem Programm und einem -> device oder einer Datei geschaltet wird und als Übergangsspeicher für den Datenaustausch benutzt wird. Die übliche Größe der vom System zugewiesenen Puffer beträgt 512 Byte. Jedes angeschlossene Laufwerk hat einen besonders großen Puffer, weil das -> trackdisk.device das sogenannte »track i/o-Verfahren« benutzt, also eine ganze Spur der Diskette in den Puffer liest. Da eine Spur 11 Sektoren zu je 512 Byte hat, vielleicht noch zusätzlich eine Menge Byte Müll enthält, werden etwa 17 KByte für jeden Diskpuffer verbraucht. Die Vergrößerung des Diskpuffers ist mit dem CLI-Befehl ADDBUFFERS möglich. Ein Puffer kann dynamisch oder statisch sein; ersterer wird während eines Programmlaufs zugewiesen und nach Beendigung des Programms dem -> heap wieder zugefügt; letzterer bleibt bis zum Ausschalten des Computers zugewiesen und kann auch bei Nicht-Benutzung nicht mehr von anderen Routinen belegt werden (siehe dynamic allocation).

bug

(bug: Käfer, Wanze) Ein in der Regel schwer auszumachender Fehler in einem Computerprogramm (siehe debug, troubleshooting).

C

Eine Programmiersprache, die aus -> BCPL über B hervorgegangen ist. Mit ihr - sie enthält Aspekte sowohl einer höheren Programmiersprache als auch der Assembler-Sprache - wurde das Betriebssystem Unix entwickelt. Die Standardbeschreibung der Sprache ist in dem Buch »Die Programmiersprache C« von Kernighan und Ritchie zu finden.

C:Directory

(commands: Systemverzeichnis) Das C-Verzeichnis des -> Amiga-DOS; enthält die Befehle und Befehlsroutinen des -> CLI, die zuerst im Verzeichnis -> SYS:, dann hier gesucht werden (siehe logical device).

cache

(auch cache-memory: Ablagespeicher) Ein (zusätzlicher) Speicherbereich, in dem Daten zwecks schnelleren Zugriffs vorübergehend abgelegt (wörtlich: »versteckt«) werden (siehe buffer).

CAD

(computer aided design) Das technische Zeichnen am Computerbildschirm anstelle des Reißbretts.

CAM

(computer aided manufacturing) Wie CAD, aber mit ausgemalten, dreidimensional wirkenden Darstellungen.

CAI

(computer aided instruction) computerunterstützter Unterricht (nicht Informatik).

CALL

(computer aided language learning) computerunterstützter (Fremd-)Sprachenunterricht.

computer science

Informatikunterricht.

case sensitivity

Unterscheidung von Groß- oder Kleinschreibung in Wörtern. Vor allem bei Systemroutinen, die in der Assembler-Sprache aufgerufen werden, ist es wichtig, die richtige Schreibweise zu beachten. Die Routinen unterscheiden zwischen Groß- und Kleinschreibung und so werden sie beim Aufruf eventuell nicht richtig identifiziert.

checkmark

(Kontrollzeichen, »Haken«) Das Zeichen zur Markierung einer aktiven Attributoption in einem -> Menü.

Chip-RAM

Wird als Bildschirmspeicher, Stapelspeicher und Programmspeicher benutzt und enthält auch sämtliche Pufferspeicher. Puffer für -> DMA-Kanäle müssen hier liegen, weil dieser Bereich zur Zeit als einziger im System für die Sonderchips (daher CHIP-RAM) und insbesondere für den -> Blitter zugänglich ist (siehe Fast-RAM). Bei den bisher erhältlichen Chips für Amiga-Computer ist das Chip-RAM die untersten 512 KByte des Speichers. Neuere Versionen der Sonderchips (-> ECS, Enhanced Chip Set) können allerdings auf 1 MByte Chip-RAM zugreifen.

chromaticity

(die »Farbigkeit« eines bestimmten Farbtons) Die Qualität der Farbe als Ergebnis der verschiedenen Anteile der Farben

Rot, Grün und Blau. Je niedriger der Wert einer Farbe, um so dunkler der Anteil an der Farbigkeit und somit die Farbe selbst. Für die voreingestellten Farben des Amiga-Basic (siehe colour registers) sehen die entsprechenden Anteile folgendermaßen (Reihenfolge R-G-B) aus:

0	0.0, 0.3, 0.6	16	0.0, 0.0, 0.0
1	1.0, 1.0, 1.0	17	0.8, 0.1, 0.1
2	0.0, 0.0, 0.0	18	0.0, 0.0, 0.0
3	1.0, 0.5, 0.0	19	1.0, 0.8, 0.6
4	0.0, 0.0, 1.0	20	0.2, 0.2, 0.2
5	1.0, 0.0, 1.0	21	0.3, 0.3, 0.3
6	0.0, 1.0, 1.0	22	0.4, 0.4, 0.4
7	1.0, 1.0, 1.0	23	0.4, 0.5, 0.5
8	0.4, 0.1, 0.0	24	0.5, 0.5, 0.5
9	0.900.3, 0.0	25	0.5, 0.6, 0.5
10	0.6, 1.0, 0.1	26	0.6, 0.6, 0.6
11	0.9, 0.7, 0.0	27	0.7, 0.7, 0.7
12	0.3, 0.3, 1.0	28	0.8, 0.7, 0.8
13	0.6, 0.1, 1.0	29	0.8, 0.8, 0.8
14	0.0, 1.0, 0.5	30	0.9, 0.9, 0.9
15	0.8, 0.8, 0.8	31	1.0, 1.0, 1.0

chunk

Ein Datenblock in einer -> IFF-Datei, in dem eine bestimmte Art von Daten enthalten ist; zur Zeit sind die vier Haupttypen FORM, LIST, PROP und CAT sowie ein Leerblock definiert. Als Untertypen sind je nach Verwendungszwecken definiert: ILBM (Interleaved Bitmap) für Grafikdateien, FTXT (formatted text), 8SVX (8-Bit Sound Samples) und SMUS (Simple Music Score). Diese sind wiederum in untergeordnete chunk-Typen aufgeteilt:

ILBM: BMHD (bitmap header), CMAP (colour map), GRAB (hotspot, der aktive Punkt des Mauszeigers oder eines »brush«), T (destination merge data), SPRT (Sprite data), CAMG (Amiga Viewport mode) und BODY (standard data chunk - enthält die eigentlichen Daten); hierzu kommen einige »privat« definierte chunks, wie CRNG (colour range DPaint) und CCRT (colour cycling range / timing Graphics)

FTXT:CHRS (characters nach -> ANSI/ISO), FONS (font specifiers), BODY

SMUS:SHDR (score header), NAME (des Stücks), (c), AUTH (Autor), ANNO (Annotation zum Stück), I (instrument specifier), TRAK (Daten), BODY; IREV ist ein »privater« chunk bei Instant Music

8SVX:VHDR (header), NAME bis ANNO wie bei SMUS, ATAK und RLSE (Attack / Release der Hüllkurve) und BODY.

Diese chunk-Typen werden durch weitere, privat definierte Typen, ständig ergänzt.

cleanup

(Speicheraufräumung) Die von einem -> process zu erledigenden »Aufräumungsarbeiten«, bevor alle zum process zugehörigen -> Tasks von den verschiedenen Systemlisten »abgekoppelt« und dadurch effektiv gelöscht werden; hierzu gehört beispielsweise die Beantwortung aller noch nicht verarbeiteten -> messages, die Freigabe von Puffern und statischen Speichern, der Aufruf der Systemroutinen für die Freigabe zugewiesenen Speicherplatzes, kurzum die Rückgabe aller von dem Programm beanspruchten -> resources.

CLI

(Command Line Interface: Befehlszeilen-Eingabe von -> Amiga-DOS) Ein besonderes -> Window (Fenster), mit dem der Programmierer Zugang zu den Amiga-DOS-Routinen erhält; diesen Zugang kann man auf verschiedene Weise erreichen: a) mit -> Preferences das CLI einschalten, die SYSTEM-Schublade öffnen und das CLI-Icon doppelt anklicken; b) während des Einlesens der Workbench-Diskette durch Unterbrechung der Ausführung der »startup-sequence« mit der Tastenkombination <CTRL D> nach Erscheinen des großen CLI-Windows; c) das Löschen oder Umbenennen der Datei »startup-sequence« (beispielsweise in »no-startup-sequence«; d) die »startup-sequence« dahingehend editieren, daß neben dem Workbench-Fenster ein CLI-Window gleich mitaktiviert wird. Jedes CLI-Window entspricht einer getrennten Task, effektiv einem weiteren -> Terminal, das im Vorder- oder Hintergrund laufen kann. Bis zu maximal 20 solche CLIs dürfen gleichzeitig in Betrieb sein, vorausgesetzt, genügend Speicherplatz ist dafür verfügbar. Ein CLI kann entweder »interaktiv« oder »nicht-interaktiv« sein; im ersten Fall werden eingegebene Befehle direkt ausgeführt, die Ausgabe erscheint im selben Fenster; im zweiten Fall läuft das CLI »im Hintergrund«, weil mit RUN oder RUN EXECUTE gestartet und hat nur ein Ausgabe-Fenster. Die Eingabe erfolgt dann entweder über die Parameterangabe nach RUN oder aus dem mit EXECUTE aufgerufenen -> batch file. Da jeweils nur ein CLI-Window »aktuell« sein kann, stellt Amiga-DOS auch Befehle für die Überwachung und den Abbruch der laufen-

den CLIs (STATUS, BREAK) zur Verfügung (siehe TCB).

Programme, die vom CLI aus gestartet werden (einschließlich der CLI-Befehle selbst), laufen alle als solche »Tochterprozesse« ab; während der Ausführung »schläft« das aufrufende CLI, bis es durch Beenden des aufgerufenen Programms wieder »geweckt« wird. Bei der neuen Workbench-Version 1.3 steht ein zusätzlicher Eingabe-Interpreter namens Amiga-Shell zur Verfügung.

clipboard.device

Eine Systemroutine, die für die Verwaltung und Ablage von »clips« (Beispiel: mit der Kamera bei Textcraft kopierte Textstellen) zuständig ist. Solche »clips« können entweder im Arbeitsspeicher des Programms, im clipboard-Speicher oder in einer Datei im clipboard-Verzeichnis einer Diskette aufbewahrt werden. Clips müssen in IFF-Format (FTXT oder ILBM) abgelegt werden. Obwohl die bisherigen Programme in der Regel nur ein Clip auf einmal verwalten, können vom System durchaus mehrere benutzt werden.

clipping

(abschneiden) Beschränkung des Bereiches der Bildschirmausgabe auf ein -> ViewPort. Clipping ist besonders bei der Verwendung von -> Bitmaps erforderlich, um zu verhindern, daß Daten (versehentlich) außerhalb des entsprechenden Bitmap-Speichers geschrieben werden, was in den meisten Fällen zu einem Systemabsturz führen dürfte.

CList.library

Systemlibrary für die Verwaltung von Zeicheneingaben.

clone

(Zweig, Ableger) Die (gängige) Bezeichnung für einen Computer, der dem IBM-PC angelehnt ist; der -> Sidecar-Zusatz des Amiga entspricht einem solchen »Ableger«.

collision stack

(Speicher des Amiga-Basic für die Ablage von Kollisionsmeldungen): Maximal 16 Einträge sind möglich, alle weiteren werden ignoriert.

colour descriptor words

Zwei (16-Bit-)Datenwörter, die eine Zeile eines Sprites definieren; die gesetzten Pixel bestimmen die Gestalt und die Farbe.

colour indirection

(indirekte Farbbestimmung) Die Bildschirmdarstellung des

Amiga besteht aus -> Bitplanes. Für jeden Punkt der jeweiligen Darstellung (Hintergrund, Fenster, Sprites,...) ist in den Bitplanes ein Bit reserviert. Aus den Informationen »Bit gesetzt (1) oder nicht gesetzt (0)« und der binären Wertigkeit der Plane (1,2,4,8,16) ermittelt der Amiga die Nummer eines Farbgisters, das dann die entsprechende Farbe enthält (Beispiel mit drei Bitplanes: 0*1 (Bit in Plane mit Wertigkeit 1 nicht gesetzt) +1*2+1*4 = Farbgister 6). Mit dieser Technik kann es bei der gleichzeitigen Verwendung von BOBs und Sprites mit mehr als 16 Farben zu einer Änderung der BOB-Farben kommen, weil die Farbgister 16 bis 31 von den Sprites genutzt werden.

colour map

Der Teil einer -> ViewPort-Definition, der die Rot/Grün/Blau-Werte der Farbgister enthält. In Amiga-Basic wird diese Liste mit dem PALETTE-Befehl erzeugt.

colour register

Der Amiga hat insgesamt 32 Farbgister mit jeweils 16 Bit. Davon werden insgesamt 12 Bit für die Definition der Rot/Grün/Blau-Anteile der jeweiligen Farbe benutzt, wodurch sich insgesamt 4096 Farben definieren lassen. Beim Aufruf von Amiga-Basic werden diese Register mit den folgenden Farben voreingestellt:

0	dunkelblau	16	schwarz
1	weiß	17	rot
2	schwarz	18	schwarz
3	orange	19	hellbraun
4	blau	20	dunkelgrau
5	purpur	21	dunkelgrau
6	zyan	22	dunkelgrau
7	weiß	23	dunkelgrau
8	blau	24	mittelgrau
9	orange	25	mittelgrau
10	hellgrün	26	mittelgrau
11	goldgelb	27	mittelgrau
12	blau	28	hellgrau
13	lila	29	hellgrau
14	grünblau	30	hellgrau
15	mittelgrau	31	weiß

Die letzten 16 Register sind ebenfalls, jeweils in Vierergruppen, paarweise den acht Sprites zugewiesen. Als Sprite-Farbgister ist das erste Register der Gruppe stets mit der Hintergrundfarbe belegt und daher »transparent«. Die voreingestellten Farben der Register 20 bis 31 stellen eine Abstufung der Farben und nicht identische Grautöne dar (siehe colour indirection, chromaticity).

command template

-> argument template.

compiler

(compile: (Daten) zusammenstellen, verfassen) Ein Programm zur Übersetzung von in einer höheren Programmiersprache (Beispiel C) verfaßtem Quellcode in maschinennahen, aber nicht unbedingt lauffähigen, binären Code. Ein solches Programm besteht in der Regel aus drei Modulen, die der Reihe nach abgearbeitet werden. Der »preprocessor« (nur bei C) leistet die Vorbereitungsarbeit, erweitert Makrobefehle, ersetzt Variablendefinitionen durch Variablenwerte oder bindet anhand der »compiler directives« (auch »preprocessor commands«) die entsprechenden -> header files in den Code ein. Der »parser« zergliedert dann Programmanweisungen gemäß den »Grammatikregeln« der benutzten Sprache und überprüft sie auf ihre Richtigkeit. Der eigentliche »compiler« übernimmt dann die Übersetzung der Anweisung in prozessorgerechte Codewerte oder in Assembler-Anweisungen (Assembler-Quellcode), die dann von einem »assembler« endgültig in Maschinensprache umgesetzt wird. Nach einer eventuellen »Optimierung« des Codes ist meist noch ein Linkdurchlauf (-> link editor) notwendig (siehe source code, object code, load file).

CON:

(-> logical device) Im -> CLI die Bezeichnung eines -> virtuellen (scheinbaren) Terminals, in der Praxis ist CON: ein (neues) -> Window für die Ein- und Ausgabe eines CLI (siehe CSI und NEWCON).

console.device

Eine Systemroutine (-> Device), die Eingaben eines -> Windows entgegennimmt und am Bildschirm anzeigt. Auf diese Weise verhält sich jedes Window wie ein erweitertes -> ASCII-Terminal. Die Steuerung der Eingabe erfolgt weitgehend über -> ANSI/ISO-Sequenzen, die von Amiga-eigenen Sequenzen ergänzt werden. Im »normalen« Betrieb werden diese Sequenzen mit Hilfe der aktuellen -> Keymaps in ASCII-Zeichen und Steuerzeichen umgewandelt. Kenntnisse der benutzten »Rohsequenzen« ist für die Steuerung der logischen Geräte -> PRT:, -> PAR: und -> SER: unter Amiga-Basic empfehlenswert; die Liste ist im -> RKM Libraries and Devices zu finden (siehe RAW:, virtual terminal).

consoledevice.library

Systemlibrary für die Verwaltung der Eingabe über das -> console.device.

container

(Behälter) Der äußere Rahmen einer -> Gadget-Darstellung (insbesondere bei -> proportional gadgets).

Copper

(co-processor) Der zweite Hilfschip von -> Agnus. Copper steuert das Videosignal und ist daher für die Darstellung am Bildschirm zuständig. Der Chip überwacht ein Register, dessen Inhalt der Nummer der gerade darzustellenden Bildschirmzeile entspricht, und kann so praktisch »im Vorübergehen« die Hardware-Register mit neuen Werten versehen, -> Sprites positionieren, Farben wechseln, -> Windows verschieben und die Daten für den -> Blitter übermitteln. Copper ermöglicht die softwaremäßig gesteuerte Einteilung des sichtbaren Bildschirms in verschiedene Windows mit eventuell unterschiedlicher Auflösung sowie die Animation.

Copper list

(Copper-Liste) Liste für den -> Copper, in der alle fortlaufenden Daten für die Bildschirmdarstellung, wie etwa die Farben, -> screen - Daten enthalten sind. Die Copper-Liste kann vom Programmierer auch manipuliert werden, um interessante Grafikeffekte zu erzeugen.

CPU

(Central Processing Unit) Hauptchip im Computer (Hauptprozessor). Die CPU des Amiga ist der MC68000, ein leistungsfähiger 16-Bit-Prozessor, dessen Nachfolger MC68010, MC68020 und MC68030 bereits auf dem Markt sind. Die CPU übernimmt die wichtigen Ausgaben in einem Computer, wie das Abarbeiten von Programmen und das Durchführen von Berechnungen. Die CPU wird im Amiga von Hilfsprozessoren, den -> Customchips unterstützt, um schneller und effizienter zu arbeiten.

critical resource

Eine -> resource, die zu einem gegebenen Zeitpunkt nur von einer einzigen Routine, beispielsweise der CPU, beansprucht werden darf (siehe deadly embrace).

critical section

Eine Operation innerhalb eines -> Tasks, die von anderen

Tasks nicht unterbrochen werden darf (siehe mutual exclusion, deadly embrace). Bei -> Exec wird ein Codeabschnitt, der auf eine globale Datenstruktur zugreifen soll, auch als »critical section« bezeichnet, weil die globalen Daten während des Zugriffs von anderen Routinen nicht verändert werden dürfen.

CSI

(Control Sequence Introducer) Steuersequenz zur Einleitung besonderer Formatierungsangaben im CLI. Ein CLI-Fenster kennt neben den normalen Eingaben auch besondere Steuersequenzen, die durch die Tastenfolge ALT+ESC plus linke eckige Klammer plus Zahl eingeleitet und mit »m« beendet werden. Mit diesen CSI-Sequenzen lassen sich Textdarstellung:

0 = normal; 1 = Fettdruck; 3 = Schrägschrift; 4 = Unterstrichen; 7 = Inversdarstellung, Vordergrundfarbe:
30 = blau; 31 = weiß (normal); 32 = schwarz; 33 = orange und Hintergrundfarbe:
40 = blau (normal); 41 = weiß; 42 = schwarz; 43 = orange bestimmen. Diese Sequenzen dürfen in einer Folge hinter einem CSI und durch Semikolon getrennt zu einer einzigen Befehlskette zusammengesetzt werden. Bei den CLI-Befehlen PROMPT und ECHO wird auch die Standardsequenz durch »*e« anstelle von ALT+ESC ersetzt (Beispiel: ECHO " *e[32; 43m Amiga-Magazin". Darüber hinaus gibt es etliche andere CSI-Sequenzen, die man auf Seite 222 des Amiga-DOS Manuals oder, für die Druckersteuerung, in der Liste auf Seite 420/421 des -> RKM Libraries and Devices finden kann.

CString.library

Systemlibrary mit den Routinen für die Zeichenkettenverarbeitung.

current directory

Das derzeit »eingeschaltete« Verzeichnis, das vom CLI praktisch als Hauptverzeichnis betrachtet wird. Die Bestimmung des aktuellen Verzeichnisses erfolgt mit dem Befehl CD (»current directory« und nicht, wie häufig irrtümlich behauptet, »change directory«) (siehe directory, filing system).

current window

Das zu einem gegebenen Zeitpunkt einzige aktive -> Window; ein laufendes Amiga-Basic-Programm hat zwar ein

Ausgabe-Window, aber kein »current window«, denn dies ist das, beim Programmlauf nicht sichtbare, Fenster der Workbench selbst. Das »current window«, das Eingabe und Ausgabe verarbeitet, wird durch die durchgezogenen Linien (1.1) bzw. Balken (1.2 und 1.3) in der Titelleiste gekennzeichnet (siehe ghosting).

Customchips

(custommade: maßgeschneidert) Die Hilfschips im Amiga, die der -> CPU wesentliche Arbeiten abnehmen, wie etwa den Bildschirmaufbau, die Anzeige und Verwaltung der Sprites, das Erzeugen von Sounds. Dadurch bleibt der CPU mehr Zeit für Berechnungen.

cut, copy, paste

(löschen, kopieren, einfügen) Diese drei Funktionen sind Standardauswahlmöglichkeiten eines jeden Edit-Menüs. Mit »cut« wird ein Bereich des Bildschirms gelöscht und im -> clipboard abgelegt; dieser Vorgang kann also, solange »cut« nicht erneut gewählt wurde, wieder rückgängig gemacht werden. Mit »copy« wird der markierte Bildschirmbereich ohne ein Löschen im »Clipboard« abgelegt. Durch »paste« kann dieser Bereich dann an beliebige Stellen wieder ins Bild »eingeklebt« werden.

cylinder

Die übereinanderliegenden Spuren einer Diskette oder einer Festplatte (Hard-Disk) bei mehrseitigen Disketten oder Festplatten. Der »cylinder« ist die Summe aller Spuren, die gelesen werden können, wenn der Schreib-/Lesekopf sich in einer gegebenen Position, also auf einer bestimmten Spur, befindet (siehe trackdisk.device).

D/A converter

(Digital/Analog-Wandler) Ein Gerät oder Schaltkreis zur Umwandlung von digitalen (binären) Werten in analoge (fortlaufende) Signale; die »Auflösung« (Genauigkeit) eines solchen Umwandlers wird von der Anzahl der für einen einzelnen Wert verwendeten Bits angegeben. Der Amiga kann außerdem über die Lautstärke etwa 2 Bit »dynamische Auflösung« pro Audiokanal zusätzlich erzeugen; ein über alle vier Kanäle ausgegebener Ton hat somit effektiv eine Auflösung von 14 Bit.

daisy chain

(wörtlich: Gänseblümchenkette) Das Aneinanderreihen

von Geräten oder Erweiterungen, wie etwa der externen Laufwerke, so daß die Datenübertragung entlang der Kette vom ersten bis zum letzten Gerät durchgegeben wird.

deadlock situation

(Pattzustand) Ein Programm kann auf ein -> resource nicht zugreifen, weil dieses von einem anderen Programm oder einer anderen Routine nicht ordnungsgemäß freigegeben wurde (siehe lockup).

deadly embrace

(tödliche Umarmung) Eine Pattsituation, die dann entsteht, wenn zwei Routinen gleichzeitig versuchen, auf ein -> resource zuzugreifen und sich dabei gegenseitig an der Übernahme der resource-Steuerung hindern. Dieser Zustand führt meistens zu einem Absturz, bei dem der Computer »paralisiert« wird (-> lockup), weil die Steuerung durch das Hin und Her zwischen den konkurrierenden Routinen »eingefroren« wird (siehe deadlock).

debug()

Die systeminterne Fehlerverarbeitungsroutine des Amigas; Daten (Registerinhalte, Speicherauszüge...) werden mit 9600 Bit/s über den seriellen Port an ein angeschlossenes -> Terminal ausgegeben. Ab Version 1.2 kann man mit dem CLI-Befehl »LoadWb -debug« das Menü für diese Routine auf der Workbench sichtbar machen; es kann aber nicht nach erfolgtem Absturz, sondern nur zum Testen oder Überprüfen bereits laufender Routinen benutzt werden (siehe bug, troubleshooting, WACK).

declaration

Die Deklaration legt die Eigenschaften einer Variablen (Datentyp, Speicherbedarf) oder den Rückgabewert einer Funktion fest. Im Gegensatz zur Definition wird kein Speicherplatz bereitgestellt.

default tool

Das Programm (die Routine), das aufgerufen werden muß, wenn ein -> project direkt gestartet werden soll. Die Angabe des »default tools« befindet sich in dem -> .info-file des Projektes und kann nach Aufruf des Menüpunktes »Info« aus dem Workbench-Menü auch geändert werden. Daher kann durch Anklicken einer Notepad-Datei diese auch von einem Textverarbeitungsprogramm eingelesen werden.

(value/parameter) Der »vor-eingestellte« - also ohne weitere Angabe gültige - Wert eines Parameters, wie etwa die Einstellung df0: für den CLI-Befehl dir.

DEC

(Digital Equipment Corporation) Hersteller der »VAX«-Computer; hat Steuersequenzen definiert, die teilweise auch beim Amiga intern verwendet werden (siehe ANSI, ASCII, ISO).

delimiter

(Abgrenzungszeichen) Das Zeichen, das den Anfang oder das Ende eines Befehls (meist Leerzeichen, auch Doppelpunkt) oder einer Zeichenkette (meist Hochkommata) markiert (siehe special characters).

Denise

Der »display encoder«-Chip, der die -> Bitplane-Daten in ein für -> Agnus (-> Copper) brauchbares Format umwandelt; dieser Chip enthält die Farb-, Sprite- und Scrollregister und überwacht die Eingaben über den -> mouseport.

depth

(Bildschirmtiefe) Die Anzahl der verwendeten -> Bitplanes, durch die festgelegt wird, wieviel Farben zur Darstellung verfügbar sind.

developer's kit

Die von Commodore/Amiga zusammengestellten Unterlagen und Hilfen für registrierte Software-Entwickler; derzeit besteht diese »Ausrüstung« aus:

Amiga technical manuals (-> RKM); Amiga manual; Schaltpläne; -> Zorro Schaltpläne; Lattice C v3.10; Macro Assembler (v.1987); SHELL und TOOLKIT von Metacomco; v.1.2 enhancer upgrade; die neuen Disketten (Kickstart, Workbench und Extras) plus Errata; v.1.2 Developer's toolkit (mit verschiedenen Hilfsprogrammen); verschiedene Programmbeispiele als Listings im »ASL Developer's Guide« und als ausgewählte Programme aus der Amiga-LibDisk-Serie; dazu der neue Linker BLink (siehe ALink). In Europa ist diese Ausrüstung zum Preis von 600 Pfund bei Ariadne Software, London, erhältlich.

.device

(sprich: dot device) Eine Systemroutine, die als softwaremäßige Schnittstelle die Datenübertragung zwischen dem Amiga und einer »device unit«

verwaltet; auch als »device driver« (Gerätetreiber) bezeichnet. Diese Routinen laufen stets als »Hintergrundtasks« und werden daher vom Benutzer nicht wahrgenommen. Durch sie können Peripheriegeräte genau wie Dateien auf der Diskette behandelt werden (device independent i/o) (siehe logical device, physical device, filing system). Zu einem .device gehören eine von -> Exec verwaltete Datenstruktur und ein Codemodul für die auszuführenden Operationen. Der Amiga verfügt über zwölf solcher »system devices«: -> audio, clipboard, console, gameport, input, keyboard, narrator, parallel, printer, serial, timer und trackdisk.

DEVS

Das »devices«-Systemverzeichnis des -> Amiga-DOS; alle Befehle zum Öffnen eines »device« suchen die nicht ROM-residenten Gerätetreiber in diesem Verzeichnis.

digitizer

(auch A/D converter) Ein Digitalisiergerät, ein Analog/Digital-Wandler; das Gegenstück zum -> D/A converter. Das zu erfassende analoge Signal wird in gleichmäßige Zeitabschnitte zerlegt, der Abschnitt gemessen und in digitale Werte umgesetzt. Es gibt zwei Hauptanwendungen: »audio digitizer« für Tonquellen und »video digitizer« (siehe Frame Grabber) für grafische Vorlagen.

directory

Das Inhaltsverzeichnis des -> filing system; eine Schublade auf der Workbench.

Als Inhaltsverzeichnis fungiert ein directory bei der Anzeige des Disketteninhaltes am Bildschirm; darüber hinaus enthält ein directory weitere Daten, wie etwa die Position und den Umfang, den PROTECT-Status, das Datum des Anlegens und eventuell auch ein Kommentar (filenote) jeder Datei auf der Diskette. Das von -> Amiga-DOS verwaltete -> filing system kennt zwei verschiedenartige directory-Typen: root directory und user directory. Beide Typen können bis zu 72 Einträge enthalten.

Das »root directory« (wörtlich: Wurzel, besser: Stammverzeichnis) ist das Hauptverzeichnis einer Diskette und wird immer in Block 880 (-> cylinder 0, -> head 1) angelegt. In diesem Verzeichnis sind die Blocknummern der in keinem anderen Verzeichnis abgelegten Dateien eingetragen.

Ein »user directory« ist ein untergeordnetes Verzeichnis, entweder ein »system directory« (siehe logical device) oder eine vom Benutzer eingerichtete »Schublade«. Im Gegensatz zum »root directory« darf ein »user directory« verschachtelt, also in einem anderen Verzeichnis eingerichtet werden. Diese Verschachtelung darf aber maximal (einschließlich root directory) sechs Ebenen tief (siehe stack) und nicht rekursiv sein (das heißt, ein Verzeichnis darf sich selbst nicht enthalten).

diskfont.library

(-> library) Eine Sammlung von Systemroutinen, die das Einlesen neuer Zeichensätze aus dem Verzeichnis »fonts« der Diskette ermöglichen und geladene Zeichensätze verwalten. Ein »geschlossener« Zeichensatz wird erst vom System abgebunden, wenn ein anderer Zeichensatz geladen wird; der von dem Zeichensatz belegte Speicherplatz bleibt also belegt, solange kein neuer Zeichensatz den Speicherplatz beansprucht.

disk gauge

(Diskettenfüllanzeige) Am linken Rand des ersten -> Window einer Diskette (dem -> root directory entsprechend) erscheint ein senkrecht verlaufender Streifen, der grob anzeigt, wieviel freier Speicherplatz auf der Diskette vorhanden ist.

Disk-Validator

Eine Datei im Verzeichnis »!«, die für das richtige Funktionieren des Systems unentbehrlich ist. Bei jedem Diskettenwechsel wird diese Routine aufgerufen, um die neu eingelegte Diskette zu überprüfen (validieren). Solange diese Validierung nicht abgeschlossen ist, kann kein Schreibzugriff (wohl aber Lesezugriffe) auf die Diskette erfolgen; es erscheint die Meldung »disk not validated«. Das Ergebnis dieser Überprüfung findet man in den -> INFO-Daten des -> CLI unter der Überschrift »soft errors«.

display field

(der Darstellungsbereich des Bildschirms) Im amerikanischen NTSC-System besteht ein display field aus 262,5 Bildschirmzeilen, beim PAL-Amiga aus 313. Davon werden 200/256 Zeilen für die Bildschirmanzeige verwendet. Die restlichen gehören zur -> vertical blanking area, in der keine Anzeige

erfolgt. In Amerika wird der vollständige Zeilensatz beim Aufbau des Bildes in $\frac{1}{60}$, in Europa in $\frac{1}{50}$ Sekunde vom Elektronenstrahl durchlaufen (siehe interlace mode).

display modes

Die verschiedenen Darstellungsmöglichkeiten des Amiga: niedrigauflösend: 320 Punkte x 200/256 Zeilen mit maximal 32 Farben (normal)

hochauflösend: 640 Punkte x 200/256 Zeilen mit maximal 16 Farben

interlace: 320/640 x 400/512 Zeilen mit 32/16 Farben

hold & modify (HAM): 320 Punkte x 200/256 Zeilen mit 4096 Farben

extra halfbrite: normal mit 64 Farbtönen

single playfield: mit nur einer Darstellungsebene

dual playfield: mit zwei Darstellungsebenen

sprite mode: mit aktiven Sprites

display processor

Ein Baustein, der nur für den Aufbau und die Verwaltung der Bildschirmanzeige zuständig ist; diese Aufgabe teilen sich beim Amiga -> Blitter, -> Copper und -> Denise.

DMA

(Direct Memory Access - direkter Speicherzugriff) Daten werden nicht von der CPU, sondern von Hilfsprozessoren (-> Blitter, Copper) im Speicher hin- und hergeschoben; dadurch wird der Hauptprozessor weitgehend für reine Rechenaufgaben frei und kann wesentlich schneller - da fast ungebremst - arbeiten.

Der Amiga verfügt über 25 DMA-Datenkanäle, von denen die meisten bereits Subsystemen zugewiesen sind: Spritemaschine acht, audio.device und Blitter je vier und die Laufwerke (maximal vier) sowie Copper je einen. Angaben im -> RKM zufolge, läuft die CPU bei gleichzeitigem Betrieb von allen vier Audiokanälen und Blitter mit einer 16farbigen niedrigauflösenden Bildschirmdarstellung (vier Bitplanes) mit 7,1 MHz, was einer Geschwindigkeitseinbuße von etwa 0,85 Prozent entspricht.

Der Hauptnachteil der DMA besteht darin, daß die erforderlichen Datenpuffer alle im -> Chip-RAM liegen müssen, was eventuell trotz freiem -> Fast-RAM zu Speicherengpässen (besonders bei Grafikprogrammen, wenn Sie im Hires-Modus oder Overscan-Modus arbeiten) führen könnte (siehe time slot allocation).

dos.library

(-> library) Diese Sammlung von Routinen bildet die Grundlage des -> Amiga-DOS und unterstützt das -> filing system sowie -> multitasking (siehe NuDOS.library)

double buffering

(Doppelpufferung, siehe buffer) Die Verwendung von zwei Puffern gleichzeitig: Während Daten des ersten Puffers verarbeitet (ausgegeben) werden, erfolgt bereits die Aufbereitung neuer Daten im zweiten Puffer und umgekehrt. Das Verfahren wird besonders bei Grafik und Musik eingesetzt, um so etwa BOBs schnell und flackerfrei zu bewegen oder längere Tonfolgen ohne Unterbrechung zu spielen.

drag

(ziehen, neu positionieren) (siehe Gadget: drag bar).

drawer

(Schublade der Workbench) Die optische Darstellung eines -> user directory. Das doppelte Anklicken eines solchen -> Icon entspricht dem CLI-Befehl »DIR Schubladename«; das Ziehen des Icon in eine andere Schublade entspricht dem Befehl »COPY Schubladename TO Schubladename ALL QUIET«.

Bei der Umbenennung einer Systemschublade oder der Verlegung eines Systemtools in eine andere Schublade, muß der Suchpfad auf dem INFO-Blatt des betroffenen Icons entsprechend geändert werden, sonst wird der Aufruf über Icon nicht gelingen.

draw(ing) modes

Die Zeichenmodi für -> graphics primitives, -> illustration data types. Für diese Modi stehen unter -> Intuition drei verschiedene »Zeichenstifte« (pens) zur Verfügung (jedem ist eine Farbe zugeordnet): der FgPen (Foreground - Primärstift) für das Zeichnen von Linien, Ausfüllen von Flächen oder Ausgabe von Texten, der BgPen (Background - Sekundärstift) für Zwischenräume innerhalb unterbrochener Linien oder Muster und der AOIPen zum Einrahmen von Flächen oder Texten. Die »drawing modes« nutzen diese Stifte wie folgt:

JAM1: es wird mit Farbe des FgPen gezeichnet (wörtlich: »quetsche eine Farbe in das Bild«).

JAM2: wie JAM1, aber mit zwei Farben; für Muster oder

unterbrochene Linien - gesetzte Bits erscheinen in der Farbe des FgPen, nicht gesetzte in der Farbe des BgPen;

COMPLEMENT: wie bei JAM2, aber mit umgekehrter Behandlung der Bits (gesetzte Bits FgPen/ungesetzte BgPen); diesen Modus benutzt man, um bereits bestehende Darstellungen zu überschreiben, ohne die Gesamtdarstellung zu zerstören;

INVERSEVID: Meist für Textdarstellung verwendet. Ergibt in Verbindung mit JAM1 transparente Zeichen mit Umrahmung in der Farbe des FgPen, mit JAM2 Zeichen in der Farbe des BgPen und Umrahmung in der des FgPen.

Die verschiedenen Darstellungsmöglichkeiten können mit dem Hilfsprogramm -> IconEd ausprobiert werden.

driver

(Gerätetreiber) Eine Routine, die ein Peripheriegerät »steuert«. In dieser Hinsicht können alle .devices des Amiga als Treiber betrachtet werden. Häufig wird die Bezeichnung aber auf Druckertreiber beschränkt. Die im Verzeichnis »devs/printers« enthaltenen Dateien sind aber keine »Treiber« in diesem Sinne, sondern -> table lookup-Dateien für die Umwandlung der Steuersequenzen des Amiga in druckerspezifische Sequenzen.

dual playfield

-> playfield.

dynamic allocation

(dynamische, nach Bedarf variierende Speicherzuweisung) Beim Amiga gilt der RAM-Speicher als -> resource; einzelne -> Tasks können je nach aktuellem Bedarf Speicherbereiche (etwa für Bitplanes, Windows, Variablen oder ähnliches) anfordern. Solche dynamisch zugewiesene Bereiche lassen sich noch während des Taskablaufes der »Allgemeinheit« wieder verfügbar machen. Ein Beispiel für eine andere Methode - der »statischen« Zuweisung - ist das Reservieren von Variablenspeicher innerhalb des zu ladenden Programmoduls. Dieser Bereich kann dann erst nach Beenden der Task, mit dem Streichen des Programmcodes aus den Systemlisten, freigegeben werden.

ECS

(»Enhanced Chip-Set«) Neue -> Customchips für den Amiga, die schneller und leistungs-

fähiger als ihre »Vorfahren« sind. Eine Besonderheit ist, daß das ECS nun insgesamt auf 1 MByte -> Chip-RAM statt wie bisher 512 KByte zugreifen kann. Sensationell ist die Fähigkeit des neuen Grafikchips Fat -> Agnus, die hohe Grafik-Auflösung im -> interlace mode ohne -> high-res flicker zu erzeugen. Die neuen Chips können in allen Amigas einfach nachgerüstet werden.

EditPointer

Ein Hilfsprogramm von -> Preferences, mit dem man die optische Gestaltung des Mauszeigers verändern kann. Die Zeigergestalt kann eigentlich für ein -> Screen oder ein -> Window unterschiedlich definiert werden. EditPointer wirkt jedoch nur auf den Zeiger von Intuition (-> mouse pointer).

EA

(Electronic Arts) Ein amerikanisches Softwarehaus (DeLuxe Serie), das als Initiator maßgeblich an der Entwicklung und Einführung des -> IFF Dateiformats beteiligt ist/war.

equal-tempered scale

(wohltemperierte Noten) Eine musikalische Progression, in der jede Note in der Frequenz um die 12. Wurzel aus 2 höher als die vorangegangene Note liegt. Ausgangspunkt ist in der Regel der Kammerton A (440 Hz) (siehe Notentabelle).

event

(Eingabeereignis) Jede Eingabe wird als »event« betrachtet. Das System kennt mehrere Klassen von »events«: rohe Tastencodes (ohne Umwandlung in ASCII-Codes), Eingaben von Tastatur und Maus, programmbezogene Eingaben von der »Konsole«, Position des Zeigers, Timer, An- oder Abwahl eines Gadget, Requestertätigkeit, Menütätigkeit, fünf verschiedene Windowtätigkeiten, Diskettenwechsel und Änderung von Preferences. Diese Klassen lassen sich weiter in Unterklassen aufteilen, bei denen man noch weiter nach dem empfangenen Code und schließlich, bei Tastatureingaben, auch nach der aktuellen Kombination der gedrückten Shift-Tasten unterscheidet. Events werden vom -> input.device mit der augenblicklichen Systemzeit (Anzahl der Sekunden und Mikrosekunden seit Einschalten des Geräts) versehen, zu Meldungsketten zusammengestellt und der Reihe nach an alle Prozesse wei-

tergeleitet. Diese holen sich dann die relevanten Ereignisse aus der Kette heraus und geben die anderen weiter. Zu diesen Ereignissen kommen auch systemspezifische, wie etwa Fehlermeldungen, -> exceptions, -> traps und -> interrupts.

event trapping

Das »Abfangen« von relevanten Ereignissen, um diese dann zu verarbeiten. Systemspezifische events werden an entsprechende Verarbeitungsroutinen (handlers) übergeben, die sofort ans Werk gehen. Maus, Menü, Gadget, Laufwerk, Alerts werden z.B. von Intuition übernommen; hier muß man bei eigenen Programmen darauf achten, daß die -> Priorität von Intuition (50) nicht niedriger als die Programmpriorität ist, sonst können solche Ereignisse nicht verarbeitet werden. Steuersequenzen von der Tastatur werden in der Regel zuerst an das gerade aktive Window geleitet. Unter Amiga-Basic müssen die dort eintreffenden Ereignisse mit dem entsprechenden »event handler« - eine vom Programmator geschriebene Routine, die durch eine ON..GOSUB-Anweisung aufgerufen wird - identifiziert und ausgewertet werden. Dabei ist zu beachten, daß maximal 20 Meldungen in einer Warteschlange auf ihre Verarbeitung warten können. Alle weiteren werden nicht registriert (siehe polling).

exception

(Ausnahmestand) Ein zum Task gehörendes -> Interruptsignal, das die Tasksteuerung zu einer besonderen Verarbeitungsroutine umleitet. Dabei werden die aktuellen Registerwerte des Tasks auf den -> Stack gesichert. Solche Ausnahmezustände können die Folge eines -> events oder eines Fehlers sein. Kann die Situation problemlos verarbeitet werden, werden die gesicherten Registerwerte in die Register zurückgeschrieben und der Task fortgesetzt (siehe trap).

EXEC(utive kernel)

(auch »multitasking executive«) Als Kernel die tiefste Ebene des Betriebssystems mit den hardware nächsten Routinen für Interruptverarbeitung, Speicherzuweisung, Listen, Taskverwaltung, das Messagesystem, Libraries, Devices und Resources sowie, unter Version 1.2, Semaphores, Kickstart und Auto.config-Unterstützung. Die Hauptfunktionen auf dieser

Ebene betreffen die -> Systemlisten und das Signalsystem, mit dem die verschiedenen im Speicher befindlichen Code-segmente sich untereinander verständigen. Nebenbei bilden die -> Kernel graphics die Grundlage des ganzen Grafiksystems. Als »multitasking executive« verwaltet EXEC ein Listen, Queues, Message-Ports und systembezogene Interrupts beinhaltendes Subsystem. In beiden Funktionen ist Exec das Fundament, auf dem zuerst -> Amiga-DOS und dann -> Intuition aufgebaut sind.

exec.library

(auch SYSBase) Die Sammlung der verschiedenen Routinen von -> EXEC. Bei der Benutzung dieser Library ist darauf zu achten, ob die gewünschte Routine »public« (frei zugänglich) oder »private« (systemintern) ist. Diese Angaben können den -> .fd files entnommen werden.

Expansion

Eine Schublade der Workbench oder ein Verzeichnis der Diskette, in der (dem) die Schnittstellenroutinen (-> driver) für selbst einbindende Erweiterungen (-> auto.config) abgelegt werden müssen, damit diese beim Initialisieren gefunden werden. Die Einbindung erfolgt mit dem CLI-Befehl BINDDRIVERS innerhalb der -> startup-sequence.

expansion

(Erweiterung) Bauteil oder Gerät, das dem vom Händler gelieferten Computer hinzugefügt wird. Für Speicherausbau, Laufwerke, Festplatten, gepufferte Systemuhr und weitere Erweiterungen stehen folgende Anschlüsse zur Verfügung:

expansion board

Eine einsteckbare Erweiterungskarte, auf der alle Erweiterungsfunktionen untergebracht sind. Der Anschluß erfolgt über ein Expansion-Chassis/Slot.

expansion bus

(A 500/A 1000) Die (seitlich) nach außen geführten Steuerleitungen des Systems (86polige Steckerleiste).

expansion card

Wie ein board; wird aber in der Regel direkt am Expansionbus angeschlossen; kann eine nicht verkleidete Platine sein.

expansion chassis

(Erweiterungshäuser): Ein außen am Expansionbus angebrachtes Gehäuse für die Aufnahme von Expansionboards.

expansion connector

Die unten im Gerät eingebaute Leiste für die 512-K-Erweiterungskarte (A 500) oder die vorn im Hauptgehäuse angebrachte Leiste für die 256-K-Karte (A 1000).

Beim A 2000 wird diese auch »processor slot« genannt und ist die ungepuffert am Systembus (Expansionbus) angebrachte Leiste entweder für die 512 K-Speichererweiterung oder für eine Erweiterungskarte mit anderen CPUs (Beispiel: 68020; siehe parallel processing). Die Belegung dieses Anschlusses weicht von der des A 1000 etwas ab.

CPUs (Beispiel: 68020; siehe parallel processing). Die Belegung dieses Anschlusses weicht von der des A 1000 etwas ab.

expansion.library

Systemlibrary mit den Routinen für die Einbindung von -> auto.config.-Erweiterungen.

expansion port

Die in der Regel hinten angebrachten Anschlüsse für Peripheriegeräte wie Drucker, Modem, Laufwerk und andere.

expansion slot

(A 2000) Die eingebauten Leisten für die Aufnahme von Amiga- oder PC-Karten (siehe bridge board). Der A 2000 hat vier Amiga- und vier PC-Slots (je zwei für XT (8 Bit) und AT (16 Bit)). Die beiden XT-Slots können durch Einbau der fehlenden Teilleisten in AT-Slots umgewandelt werden, da alle Leitungen bereits vorhanden sind. Ebenfalls als Expansionslots bezeichnet man bei allen Amigas die 128 4-KByte-Abschnitte im Adressbereich von \$E80000 bis EFFFFF (auch external decoder expansion space genannt) für den Anschluß von -> auto.config.-Erweiterungen. processor slot (A 2000): -> expansion connector (A 2000). video slot (A 2000): Eine Leiste für den Anschluß einer Fernsehmodulatorkarte (NTSC oder PAL) oder einer -> Genlockkarte.

extra halfbrite

(halfbright) Der »halbhelle« Farbmodus, wodurch mit 32 -> colour registern 64 Farbtöne erzeugt werden können; erst ab Version 1.2 des PAL-Amiga voll unterstützt. Ähnlich wie bei -> Hold & Modify wird hier ein sechstes -> Bitplane zu Steuerzwecken benutzt. In diesem Fall bedeutet ein gesetztes Pixel volle, ein nicht gesetztes Pixel halbe Leuchtkraft des entsprechenden Bildpunktes. Mit

diesem Verfahren steigt der Speicherbedarf je Bitplane auf 48 KByte.

.fd files

(file descriptor) Eine Sammlung von reinen Textdateien, die eine Auflistung aller in den Systemlibraries vorhandenen Routinen mit den erforderlichen Übergabeparametern und den verwendeten Registern enthält; eine unentbehrliche Programmierhilfe. Für die Verwendung von Libraries unter Amiga-Basic müssen die .fd files in -> .bmap files umgewandelt werden. Das Verzeichnis »fd1.2« auf der »ExtrasD-Diskette 1.2« enthält die .fd files für die Version 1.2; die .fd files zu Version 1.1 sind auf der Development Assembler-Diskette (Ausgabe 1986) von Commodore zu finden (siehe library).

Fast RAM

Die Speicherbereiche \$0200 00 bis \$9FFFFF und \$C00000 bis \$CFFFFF (1024-KByte-Erweiterungen); sie erhielten diese Bezeichnung, weil nur die CPU auf diese Bereiche zugreifen kann und daher von den Sonderchips nicht gebremst wird. Mit -> ATOM werden Programme so vorbereitet, daß Codemodule im »Fast RAM« und Datenmodule im -> Chip RAM liegen. Die Auslagerung von Codemodulen in das »Fast RAM« soll Geschwindigkeitsgewinne von bis zu 30 Prozent ergeben.

Fat-Agnus

Nachfolger des Agnus-Chips aus dem Amiga 1000, der in den Amiga-Modellen A 500 und A 2000 Anwendung findet. Im Gegensatz zu Agnus sind im Fat-Agnus Schaltungen integriert, die zuvor noch auf der Hauptplatine des A 1000 zu finden waren. Der neue Chip enthält eine komplette -> Refresh-Logik für die RAM-Bausteine, so daß Speichererweiterungen am Amiga sehr einfach vorgenommen werden können. Zudem stellt Fat-Agnus alle Systemtakte, das heißt die »Stoppuhr«, nach der der Amiga läuft.

filing system

(Dateiverwaltungsstruktur unter Amiga-DOS) Die Grundlage des filing system beim Amiga ist eine Kombination dreier Komponenten: volume (die Diskette oder Festplatte), directory (eine Schublade) und file (die eigentliche Datei). Die dem System zugrundeliegende Struktur bezeichnet man als Baumstruktur.

Jedes »volume« enthält ein sogenanntes »root directory« (Wurzel-Verzeichnis; hier besser Stamm-Verzeichnis). Wie sich Äste an einem Baumstamm befinden, so enthält das Stammverzeichnis weitere Unterverzeichnisse oder bereits Dateien (Blätter des Baumes). Solche Unterverzeichnisse können selbst wiederum Verzeichnisse (Zweige eines Astes) enthalten oder Dateien. Diese Verzweigungen (Verschachtelungen) lassen sich ohne weiteres bis zu sechs Ebenen tief bilden. Die Speicherung von Dateien ist so eingerichtet, daß die »header blocks« (die Datei beschreibende Daten) auf einer Seite der Diskette und die reinen Datenblocks zunächst auf der anderen Seite abgelegt werden. Man unterscheidet zwei Dateitypen: Textdateien oder binäre Dateien (-> load files). Amiga-DOS identifiziert und verwaltet (Text-)Dateien mit Hilfe von zwei Datenstrukturen, dem »file handle« und dem »file lock«. Ein »file handle« ist ein Datenbereich (oder ein Zeiger auf diesen), in dem das System Informationen über diese Datei (Adresse des Datenpuffers, Zeiger auf das nächste zu lesende Zeichen und andere) findet. Der »file handle« als Zeiger (Zeiger: Adresse auf einen Speicherbereich) wird dann beim Eröffnen, Lesen und Beschreiben sowie beim Schließen der Datei angegeben. Dieses Prinzip wird auch bei -> Intuition, als »file descriptor« für gepufferte Daten oder »file pointer« für nicht gepufferte Daten in ähnlicher Weise verwendet.

Da Dateien unter Umständen von mehreren Routinen/Programmen gleichzeitig bearbeitet werden können, dient ein weiterer Datenbereich, der »file lock«, als »Zugriffsschlüssel«. Von diesen gibt es zwei Typen: shared read und exclusive write. Damit sind gleichzeitig Lesezugriffe von mehreren Programmen, ein Schreibzugriff aber nur von einem Programm möglich.

Neben den »data files« kennt das Amiga-DOS auch »device files«. Hierbei handelt es sich um die -> physical devices des Systems (par., ser: und andere), die für die oben erwähnten Zwecke genau wie Dateien betrachtet werden, das heißt die Übertragung von Daten an diese Geräte wird genauso gehandhabt, als handle es sich um eine Datei auf einer Diskette (device independent i/o). Der

Programmierer kann entscheiden, ob die ein- und auszugehenden Daten gepuffert oder ungepuffert übertragen werden sollen.

Neben den beschriebenen Funktionen spielt auch die »file maintenance«, die Pflege des filing systems, eine nicht geringe Rolle. Hierzu gehören Operationen wie Umbenennungen (RENAME, RELABEL), Dateien verbinden (JOIN), Daten anderweitig ablegen oder zuordnen (COPY, DISKCOPY), Löschen (DELETE), SORTieren, Aktennotizen anfertigen (FILENOTE), Zugangsstatus bestimmen (PROTECT) oder neue Schubladen für die Dokumentenaufnahme vorbereiten (FORMAT, INSTALL, MAKEDIR).

filename conventions

(Vereinbarung der Dateitypen) Amiga-DOS erlaubt Dateinamen mit einer Länge von bis zu 30 Zeichen. Zusätzlich kann jede Datei mit einem bis zu 80-stelligen Kommentar versehen werden. Eine einfachere Methode, den Inhalt einer Datei näher zu bestimmen, ist die Kennzeichnung des Dateityps durch einen sogenannten »dot suffix« (.Zusatz). Dieser besteht - in Anlehnung an eine für MS-DOS bestehende Konvention - in der Regel aus drei Buchstaben, die durch einen Punkt vom eigentlichen Dateinamen getrennt an diesen angehängt werden. Folgende »Namenserweiterungen« in Tabelle 1 haben sich bisher etabliert.

fill operation

(Ausfüllen einer Fläche) > graphics primitives.

first fit

(erster Anpassungsversuch) Beim Laden eines Programms (Codesegments) muß erst ein entsprechend großer Speicherbereich gefunden werden. Anhand der »available list« (auch: free list) wird der erste Speicherbereich aufgefunden gemacht, dessen Umfang größer als der Bedarf ist, und der Bereich zugewiesen (siehe fragmentation, heap).

Fish Disks

(offiziell: AmigaLibDisk) Die von dem Amerikaner Fred Fish zusammengestellte Sammlung von -> Public-Domain-Software für den Amiga (zur Zeit ca. 250 Disketten). In der Regel befindet sich auf der Diskette jedes Programm als -> load file sowie als Quellcode. So kann man eigene Änderungen und Verbesserungen hinzufügen.

Durch seinen guten Ruf erhält Fred Fish auch sonst nicht leicht zugängliche Materialien für diese Sammlung, wie etwa die IFF-Diskette für Softwareentwickler von Electronic Arts, Demonstrationsversionen kommerzieller Programme (Aegis Animator, Digiview, TxEd und andere) und Beispiele der Systemprogrammierung vom Entwicklerteam des Amiga.

Frame Grabber

(siehe digitizer) Die Bezeichnung für ein schon seit der amerikanischen Einführung des Amiga angekündigtes, aber bisher nicht erschienenenes Zusatzgerät für das Digitalisieren von Videobildern in »Echtzeit«. Das heißt etwa 25 Bilder pro Sekunde müssen erfaßt und gespeichert werden können, um eine Szene nicht zu stoppen.

filename	conventions
.acbm	Bitplanes: Amiga-Basic-Format
.alt	Alternativ(text)
.asc	Text als ASCII-Zeichen
.asm	Assembler-Quellcode
.bas	Basic-Programmtext
.bkp	Sicherheitskopie (backup)
.bmap	Amiga-Basic Library-Parameter
.bob	BOB-Daten
.c	C-Quellcode
.cmd(s)	Befehlsliste eines Programms
.com	Kommandodatei (batch file)
.cps	Farbregister (color palette)
.doc	Erläuterungen (documentation)
.fd	Parameter zu einer Library
.font	Zeichensatzdefinitionsdatei
.h	header-file für C Programme
.i	include-file für Assembler
.info	Icondatei
.lib	Programmiersprache-Library
.library	System-Library
.l(i)nk	Befehlsdatei für den Linker
.man	Handbuch (manual)
.o	object code
.pic	Bilddatei (picture)
.scr	screen (bei FORTH)
.script	Animationsbefehle (ANIMATOR)
.spr	Spritedaten
.s(rc)	Quellcode allgemein (source)
.tmp	Übergangsdatei (temporary)
.txt	Text (meist Systemformat)

Tabelle 1. Gängige Extensionen bei Filenamen

font

(Zeichensatz) -> diskfont.library, kerning, pitch, spacing

FONTS

Der Amiga sucht in diesem Systemverzeichnis die angeforderten Zeichensätze. Zu jedem Zeichensatz gehören eine Datei (fontname.font; Beispiel: diamond.font) und ein Verzeichnis, das seinerseits einzelne Dateien enthält, deren Name der Höhe des Zeichensatzes in Pixeln entspricht. Die .font-Datei enthält die allgemeinen Merkmale und Daten des Zeichensatzes, die Dateien im Verzeichnis enthalten die Datenbytes der einzelnen Zeichen. Diese werden zeilenweise vom ersten bis zum letzten Zeichen in der Datei abgelegt und dürfen bis zu maximal 256 Zeichen umfassen. Über die .font-Datei ist es auch möglich, nur bestimmte Tastencodes mit neuen Zeichen zu versehen (siehe illustration data types).

fragmentation

(Fragmentierung, Aufsplittung des freien Speichers) Bei der Zuweisung von Speicherbereichen nach dem -> first fit-Verfahren entstehen möglicherweise viele kleinere Bereiche, die zwar frei, aber dennoch für ein zu ladendes Codesegment zu klein sind. Es kann daher zu einer »out of memory«-Fehlermeldung kommen, obwohl insgesamt durchaus ausreichend Speicher vorhanden wäre. Eine Überprüfung des Speicherzustandes, vor allem beim Programmieren, ist mit dem CLI-Befehl FRAGS (auf der Development Asssembler-Diskette) möglich.

frequency modulation

(siehe attached channels) Die (fortlaufende) Änderung der Frequenz einer Wellenform, um besondere Effekte, wie etwa Glissandi, zu erzeugen oder auch um Informationen zu übertragen.

function

In C entspricht eine Funktion etwa einem Unterprogramm (einer Funktion in Fortran oder Prozedur in PL/1, Pascal...). Innerhalb des Programmsegmentes der Funktion können Variablen definiert werden, die in der Regel nur während der Ausführung der Funktion gültig sind. So lassen sich Funktionen unabhängig vom Hauptprogramm entwickeln und ausführen. Im Gegensatz zu einer $\rightarrow$ procedure gibt eine Funktion einen Rückgabewert zurück (Ausnahme: die void-Funktion in C). Dieser Wert kann eine Meldung über Erfolg oder Mißerfolg des Aufrufs oder der Ergebnisswert der Funktionsoperation sein. Die Überprüfung solcher »Return codes« ist empfehlenswert, um sicherzustellen, daß eine nachfolgende Funktion nicht durch einen falschen Wert auf Irrwege gebracht wird.

Funktionen können vom Anwender durch Verwendung der entsprechenden Sprache (C, Pascal) selbst programmiert werden oder aber als Assemblerunterprogramm zum Lieferumfang der Sprache gehören (Beispiel: die trigonometrischen Funktionen SIN(), COS() in Basic). In beiden Fällen sind Funktionen an den für die Parameterübergabe unentbehrlichen runden Klammern zu erkennen.

gadget

(ein »Werkzeug« mit einer nicht genau definierbaren Funktion(sweise), ein »Apparat« oder »Dings«; laut Anwenderhandbuch: Symbol). Ein Gadget ist die grafische Darstellung einer Steuerungsfunktion. Diese kann eine der vier als »custom gadget« zur Verfügung gestellten Typen sein:

Boolean gadget: Ein einfacher Schalter, der die Wahl einer ja/nein-Operation ermöglicht. Hierzu gehören beispielsweise die Wahlschalter für CLI oder 80 Zeichen bei $\rightarrow$ Preferences.

string gadget: Ein Eingabefeld für Texteingaben, wie etwa bei der Menüfunktion »Rename« oder bei der Eingabe eines Dateinamens bei Textcraft. Unter Kickstart 1.1 muß man ein solches Gadget mit der Maus zuerst aktivieren, bevor man Text eingeben kann, ab Version 1.2 erfolgt die Selektion in der Regel automatisch beim Aufruf des entsprechenden Gadgets.

integer gadget: Ein Eingabefeld für die Eingabe von Zahlen-

ketten; sonst wie string gadget. **proportional gadget:** Ein Regler, wie etwa für die Farbwerteinstellung oder die Mausfunktionen bei Preferences. Die Regelung erfolgt entweder über einen Knopf (englisch: knob), gelegentlich auch als »autoknob« mit veränderbarer Gestalt, oder über einen Schieberegler (slider).

Solche Reglergadgets können vom Benutzer frei gestaltet und im zugehörigen Window nach Belieben positioniert werden. Hierzu stellt $\rightarrow$ Intuition die $\rightarrow$ illustration data types zur Verfügung.

Die sogenannten »system gadgets« dagegen sind in Form und der Position festgelegt. Der Programmierer kann lediglich den enthaltenen Text variieren.

Oben links in einem $\rightarrow$ Window erscheint das »close gadget«, der Abstellknopf (Schließsymbol) für den dazugehörigen Darstellungsbereich. Die Wahl dieses Knopfes signalisiert dem Programm, daß der Benutzer dieses Window nicht mehr benutzen will; das dazugehörige Programm kann das Fenster also ausschalten. Oben in der Titelleiste eines Windows oder $\rightarrow$ Screens befindet sich das »drag gadget«, die Ziehleiste oder der Verschiebeapparat, mit dem man die Position des Window beliebig, die eines Screen nur in vertikaler Richtung, ändern kann, indem man das Gadget mit der Maus anklickt und bei weiterhin gedrückter linker Maustaste den Bildschirm(bereich) in die gewünschte Position zieht.

Oben rechts findet man das »depth gadget«, der Sicht-, Stapel- oder Prioritätsknopf, mit dem man bestimmt, welches von zwei sich überschneidenden Windows oder Screens sich vorn – und damit voll sichtbar – befindet (die rechte Hälfte), oder ein verdecktes Window voll sichtbar macht beziehungsweise ein sichtbares Window nach hinten versetzt (die linke Hälfte). Die schwarzen Bereiche des gadgets entsprechen dem unsichtbaren Teil des Windows.

Unten rechts in einem Window befindet sich das »sizing gadget«, der Größenregler, mit dem man die Größe des Windows bestimmen kann.

Zu den »system gadgets« gehören besondere »window gadgets«, die die Darstellung innerhalb eines Windows steuern. Sie befinden sich in der unteren und rechten Leiste des Fensters und heißen »scroll bars«. Mit

diesen Reglern wird das Fenster über den Inhalt hin- und hergeschoben. Schließlich gibt es bei dem Hauptfenster einer Diskette, dem $\rightarrow$ root directory (Hauptverzeichnis) entsprechend, ein zusätzliches Gadget, das »disk gauge« – auch Füllanzeige genannt, welches annähernd anzeigt, wieviel freier Platz sich noch auf der Diskette befindet. Dieses Gadget ist rein passiv und kann vom Benutzer nicht geändert werden.

Aus der geschilderten Vielfalt kann man erkennen, welche Rolle diese Systemregler spielen (siehe Requester).

gameport

Anschluß für einen »controller« oder sonstige Peripheriegeräte, wie etwa externe Uhr oder Audio-/Digitalisiergerät. Drei verschiedene »controller« werden unterstützt: Maus oder Trackball-/Joystick, absoluter oder relativer Joystick. Beide Anschlüsse sind für eine Maus mit drei Steuertasten ausgelegt. Lichtgriffel werden in der Regel an Port 1 erwartet, damit deren Eingaben vom $\rightarrow$ input.device verwaltet werden können.

gameport.device

Die Systemroutinen für die Verwaltung der beiden Maus- oder Joystickanschlüsse. Das $\rightarrow$ physical.device besteht aus den beiden Anschlüssen, das gameport.device bezieht sich in der Regel nur auf den zweiten Anschluß, weil der Anschluß Nummer 1 als Mausport vom $\rightarrow$ console.device oder input.device verwaltet wird.

garbage

(Müll, Abfall) nicht brauchbare Daten oder Dateien; Zufallswerte bei nicht initialisierten Variablen.

GCR

$\rightarrow$ pulse code modulation.

Garry

(Gate ARRAy) Ein spezieller Baustein beim A500/A2000, der mehrere Schaltungen des A1000 beinhaltet, die für die Chipselect-Logik und Teile des Floppy-Controllers benutzt werden.

GELs

(graphic elements) Die »animierbaren« Teile des Grafiksubsystems; bestehend aus: Simple Sprites, VSprites, BOBs, AnimComps und AnimObjs. Die verschiedenen Softwareelemente dieses Systems lassen sich (aufeinander) aufbau-

end zu komplexen Animationsgebilden vereinigen; so besteht ein AnimObj aus mehreren AnimComps, die ihrerseits aus mehreren BOBs oder Sprites bestehen können.

Simple Sprites sind die Hardware-sprites. Davon gibt es insgesamt acht; jedes mit einem eigenen $\rightarrow$ DMA-Kanal. Die Breite – wie bei allen Spritevarianten – ist auf 16 Pixel niedriger Auflösung (320 Pixel/Zeile) beschränkt. Die Höhe ist nur durch den verfügbaren Speicher des Chip-RAM begrenzt. Solche Sprites können »direkt« von einem Programm angefordert werden und gehören dann nicht mehr zum eigentlichen GELS-Subsystem. Bei der Benutzung von Simple Sprites ist darauf zu achten, daß Sprite 0 für den Mauszeiger reserviert ist und daher stets von $\rightarrow$ Intuition in Anspruch genommen wird. Der Zeiger kann als einziges Sprite nicht durch überbreite $\rightarrow$ Raster ausgeschaltet werden (siehe time slot allocation). Außerdem teilen sich immer zwei Sprites einen Satz Farbre-gister ($\rightarrow$ colour registers); Sprite 1 erhält also immer dieselbe Farbkombination wie Sprite 0, es sei denn, diese sind verschiedenen $\rightarrow$ View-Ports zugeteilt worden. VSprites (virtual sprites: scheinbare Sprites) werden den Hardware-Sprites zugewiesen (sofern diese nicht anderweitig als simple sprites in Anspruch genommen wurden, weil ein Sprite-DMA-Kanal mehrfach verwendet werden darf, sofern mindestens eine Bildschirmzeile zwischen den beiden verschiedenen VSprites liegt. Dadurch ist es möglich, erheblich mehr als acht Sprites gleichzeitig am Bildschirm darzustellen. Ein Problem gibt es allerdings bei VSprites: diese werden immer relativ zum $\rightarrow$ View und nicht zum $\rightarrow$ View-Port positioniert, das heißt sie können – obwohl für ein Fenster bestimmt – sich trotzdem selbstständig machen und überall im aktuellen $\rightarrow$ Screen auftauchen! Die Verwendung ist dadurch effektiv auf Darstellungsbereiche eingeengt, die als ganzer Bildschirm definiert werden.

BOBs (blitter objects) sind eigentlich Teile eines $\rightarrow$ playfield, also »Bildschirmabschnitte«, die vom $\rightarrow$ Blitter von einer Position zu einer anderen kopiert werden und dadurch Animation »vortauschen«. Anders als Sprites haben BOBs die Farb- und Auflösungsattribute desjenigen Screens, in

dem sie definiert werden. Bei 32-Farben-Darstellung können unter Umständen Farbregisterkonflikte zwischen BOBs und Sprites auftreten, wenn sie sich gleichzeitig in demselben horizontalen Bildabschnitt befinden. In den Ausmaßen sind BOBs auf Höhe und Breite des verwendeten playfield oder Raster beschränkt. Wird bei der Benutzung von BOBs kein -> double buffering benutzt, kann es vorkommen, daß beim Kopieren eines BOBs die einzelnen Bitplanes nicht ganz synchron verschoben werden. AnimComps (animation components: animierte Bestandteile) bestehen aus einer Serie von unterschiedlichen BOBs oder Sprites, die die Einzelphasen eines Bewegungsablaufs darstellen. Dies kennt man auch aus der traditionellen Animation (siehe animation).

AnimObjs (animation objects) sind die wesentlichen Bestandteile einer Animation und bestehen normalerweise aus mehreren (auch mit unterschiedlicher Geschwindigkeit ablaufenden) Animationssequenzen (Beispiel: ein Radfahrer, dessen Beine sich nicht so schnell bewegen wie die Räder sich drehen).

Bei allen GELs ist zu beachten, daß die Positionierung zuerst nach der Bildschirmzeile (y-Koordinate) erfolgt. Dies beeinflusst gegebenenfalls die Darstellungspriorität. Sprites haben eine feste Prioritätszuweisung: das Sprite mit der niedrigsten Nummer wird immer vorrangig, also vor allen anderen im sichtbaren Bild dargestellt. Dies kann eventuell bei VSprites etwas problematisch sein (siehe oben). Als Grundregel gilt, daß das Objekt mit der höchsten Priorität als Vordergrundobjekt dargestellt wird (siehe animation, motion control, sequenced drawing, masks).

genlock

(auch video sync) Ein Zusatzgerät, das die Bildsignale einer externen Videoquelle (Fernsehbild, Videokamera, Videorecorder) mit den Bildsignalen des Amiga synchronisiert und kombiniert, indem die Hintergrundfarbe des Amigabildes (Farbregister 0) durch das externe Bild ersetzt wird. Auf diese Weise können Amiga-Grafiken in Fernsehbilder eingearbeitet und über einen weiteren Videoausgang aufgezeichnet werden. Diese Technik verwendet man für das Video-Titeln und für Trick-Szenen.

ghosting 1

Die »Geisterschrift« bei Menüs und Fenstern. Die Titelleiste eines nicht gewählten Fensters sowie nicht wählbare Menüoptionen erscheinen mit einem überlagerten Punkteraster. Durch diese Unterbrechung der Schrift wird die Nichtverfügbarkeit signalisiert.

ghosting 2

Unter bestimmten Bedingungen treten »Geisterbuchstaben« bei der Tastatureingabe auf, die auf die Beschaffenheit der Tastaturmatrix zurückzuführen sind. Es kann vorkommen, daß beim gleichzeitigen Drücken mehrerer Tasten der Code einer nicht gedrückten Taste mit ausgegeben wird. Diese Situation tritt normalerweise erst bei gleichzeitigem Druck von drei oder mehr Tasten auf, was in der Regel sehr selten ist.

graphics.library

Die Sammlung der Systemroutinen für die ROM-Kernel-graphics.

graphics primitives

(auch: ROM Kernel graphics) Diese Routinen des Kernel stehen in direkter Verbindung mit der Grafikhardware des Systems; sie sind von den -> illustration data types (Intuition) völlig unabhängig, unterstützen diese aber auf der niedrigsten Systemebene und können auch von Intuition aus direkt aufgerufen werden. Die meisten Routinen benötigen hierfür die Angabe eines -> RPorts oder RastPorts. Ein Problem bei diesen Routinen besteht darin, daß sie, anders als die illustration data types, nicht an den Grenzen des Window oder Screen gebunden sind und deswegen unter Umständen Daten außerhalb des eigentlichen Bitplanespeichers ablegen können. So etwas kann durchaus zu einem Absturz des Programms führen. Folgende Routinen stehen zur Verfügung: ReadPixel, WritePixel, Move, Draw, Polydraw, RectFill, AreaMove, AreaDraw, Flood, SetDrPat, SetAfPat, SetRast, SetAPen, SetBPen, SetOPen, ScrollRaster (siehe -> draw modes), DrawEllipse und AreaEllipse. Die erforderlichen Parameter können dem entsprechenden -> .fd file (graphics.lib__fd) entnommen werden.

Durch die Verwendung von Routinen der »intuition.library« statt des graphics primitives kann sich der Programmierer viel Verwaltungsarbeit ersparen.

guru meditation nr

(system alert: Absturzwarnung) Hier handelt es sich um eine besondere Variante eines -> requester. Die einzigen Parameter, die hier möglich sind, lösen entweder einen -> reset (linke Maustaste drücken) oder die Einschaltung einer -> debug-Routine (siehe WACK) aus. Die debug-Routine leitet Informationen über den Fehlzustand des Systems an ein am seriellen Port angeschlossenes externes -> Terminal weiter (rechte Maustaste drücken). Die in den symbolischen Farben Rot und Schwarz gestaltete Meldung erscheint in den obersten Bildschirmzeilen und - anders als bei anderen Requestern - schiebt der Amiga den Bildschirm um die entsprechende Anzahl Zeilen nach unten. Bei Erscheinen dieses »alerts« ist meist nichts mehr zu retten. Die »guru meditation nr« selbst sollte ursprünglich dem Programmator Informationen über die Art des aufgetretenen Fehlers geben. Zu diesem Zweck dient zunächst die linke Hälfte der Nummer. In diesem -> longword werden verschiedene Angaben verschlüsselt dargestellt. Hierfür ist das Langwort in vier Zahlen aufgeteilt: das erste Zeichen kann eine 0 oder eine 8 sein. Die 0 bedeutet »recovery alert«, das bedeutet, daß der Fehler zu beheben ist (Voraussetzung: externes Terminal). Die 8 steht für »dead-end alert« (Sackgasse); ein Schaden ist bereits angerichtet und das System teilt dies lediglich als »Abschiedsgeschenk« mit. Es bleibt nur noch der Griff zu den Resetstasten.

Die beiden ersten Zeichen werden auch als »subsystem id« verwendet. Dieses id ist die Kennzahl des Systemteils, in dem der Fehler aufgetreten ist: 01 bis 0A sind die -> Libraries: Exec, Graphics, Layers, Intuition, math, CList, DOS, RAM, Icon, and Expansion; 10 bis 15 sind die -> devices: audio, console, gameport, keyboard, trackdisk und timer; 20 bis 22 sind die -> resources: CIA, disk und sonstige; 30 bis 32 verweisen auf die Workbench: bootstrap, Workbench selbst und DiskCopy. Gelegentlich erscheinen diese Zahlen auch in dem zweiten Zeichenpaar, das eigentlich für die Identifizierung des allgemeinen Fehlertyps gedacht ist. Hier kommen die Zahlen 01 bis 07 vor: Speichermangel, Library nicht anlegbar, Library nicht zu öffnen, Device nicht frei, Resource nicht frei,

I/O-Fehler, Signalfehler. Die letzten vier Zeichen werden für subsystemspezifische Angaben benutzt. Davon gibt es über 60. Die vollständige Liste ist dem Exec Reference Manual, Seite A-2/A-3 oder A-16/A-17, zu entnehmen.

Das letzte Zeichenpaar kann auch das Auftreten einer -> Trap der CPU bedeuten. In einem solchen Fall erscheint nur die entsprechende Zahl mit vorangestellten Nullen; die wahrscheinlichsten Nummern hier sind: 2=Busfehler, 3=Adreßfehler, 4=illegaler Befehl, 5=Division durch Null und 8=Privilegverletzung; die Zahlen 10 und 11 können nur bei einer 68010-CPU erscheinen und betreffen die diesem Prozessor spezifischen Befehle, die beim 68000er nicht vorkommen.

Das zweite Langwort, rechts von dem Punkt, enthält theoretisch die Anfangsadresse des fehlerhaften Tasks bzw. den letzten im -> task frame abgelegten Datenwert. Einige Programmierer erlauben sich den Spaß, das Wort HELP als ASCII-Zeichenkette (48454C50) hier einzusetzen.

handler

(Verarbeitungsroutine) Eine solche Routine hat eine eng begrenzte Aufgabe und wird meistens systemintern für die Verarbeitung von -> events, exceptions, traps, interrupts und messages benutzt. Einzelne Routinen werden auch für die Herstellung bestimmter Geräteverbindungen (Beispiele: Port-handler, RAM-handler) benutzt (siehe server).

hard disk

(Festplatte) Ein Speichergerät, das, anders als ein Disketten-Laufwerk, eine fest eingebaute Platte zur Aufnahme der Daten enthält. Dies können auch mehrere übereinander angebrachte Platten sein, wovon jede mit einem eigenen Schreib/Lesekopf ausgerüstet ist (siehe SCSI, ST506).

hash table

(Identifikationszahltablelle) »hashing« bezeichnet ein Verfahren, das unter anderem bei relationalen Dateien benutzt wird, um einem bestimmten Datensatz eine (möglichst) einmalige Kennzahl zuzuweisen. Aus dieser Kennzahl kann die Position des Datensatzes auf dem Speichermedium ermittelt werden. Amiga-DOS benutzt ein ähnliches Verfahren für die Zuweisung von oder die Suche nach den zu einer Datei zuge-

hörigen Datenblöcken. Die »Adressenliste« der Blöcke wird in einem $\rightarrow$ directory oder file header oder file list block eingetragen (siehe filing system). In erster Linie braucht man dieses Verfahren bei der Arbeit mit dem Entwicklerprogramm DISKED, das mittlerweile durch DISKDOCTOR abgelöst wurde.

header file

$\rightarrow$ include file.

heap

(wörtlich: Haufen) Der frei verfügbare Speicher des Systems oder eines Programms (siehe fragmentation, stack).

highlight

(hervorheben) Die in der Regel farbliche, gelegentlich auch formmäßige, Änderung eines $\rightarrow$ Icon oder einer Menüoption als Zeichen dafür, daß das Icon bereits gewählt wurde oder daß die Funktion durch Loslassen der rechten Maustaste gewählt werden kann (siehe IconEd).

high-res flicker

(auch »the jitters«) Das Flackern des Bildschirms im hochauflösenden Interlace-Modus (640 x 400 bzw. 512 Pixel).

Dieses Phänomen tritt beim europäischen Amiga stärker als bei den amerikanischen Geräten auf. Die Ursache ist jedoch in beiden Fällen dieselbe. Im Interlace-Modus besteht das Bild aus zwei vertikal um eine Bildschirmzeile gegeneinander versetzte $\rightarrow$ display fields; der Bildaufbau dauert dadurch doppelt so lange wie bei einem normalen Bild: $\frac{1}{25}$ Sekunde statt $\frac{1}{50}$ in Europa, $\frac{1}{30}$ Sekunde statt $\frac{1}{60}$ in Amerika. Während das zweite display field vom Elektronenstrahl »gezeichnet« wird, fängt das erste bereits an, zu verblenden. Die Leuchtkraft der einzelnen Bildpunkte in der Phosphorbeschichtung des Bildschirms läßt so deutlich nach, daß beim erneuten Aufbau der Teilbilder die Schwankungen der Leuchtkraft vom Auge als Flackern registriert werden. Die Wahl von kontrastarmen Farbkombinationen mindert diesen Mangel. Durch Sonnenbrillen oder einen als »Jitter-rid« bezeichneten Aufsatz kann der Kontrast ebenfalls verringert werden. Die kostspieligste Alternative ist ein sog. »high persistence monitor« mit nachleuchtender Phosphorbeschichtung. Oder aber man wartet auf das neue $\rightarrow$ ECS, das den Amiga befähigt, die Auflösung von 640 x 512 Punkten ohne Flimmern darzustellen.

hold & modify (HAM)

Ein $\rightarrow$ display mode, bei dem alle 4096 Farben des Amigas dargestellt werden können. Da die Bitplanes 5 und 6 zur Steuerung dieses Modus verwendet werden, stehen trotz der Verwendung aller Farbbregister nur 16 Grundfarben zur Verfügung. Die Bezeichnung dieses Modus bedeutet, daß beim Abbilden eines Pixels zwei der Farbwerte eines benachbarten Pixels »festgehalten«, also unverändert übernommen werden, während der dritte Farbwert eventuell modifiziert wird. Welcher Farbwert geändert wird, hängt von der Bitkombination in den beiden Steuer-Bitplanes ab (00: keine Änderung, 01: Blauwert, 10: Rotwert, 11: Grünwert). Geändert werden die Farbbregister 16 bis 31 dadurch, daß der zu ändernde Farbwert des Pixels unmittelbar links von der aktuellen Bildschirmposition aus der Bitkombination der Bitplanes 1 bis 4 ersetzt wird. Dieser Modus ist zeilenorientiert; eine Änderung wird nicht in die nächste Bildschirmzeile übertragen. Dafür wird dem äußersten linken Pixel der Zeile die Hintergrundfarbe zugewiesen.

horizontal blanking

Der Bereich einer Bildschirmzeile, in dem keine Darstellung am Bildschirm erfolgen kann (siehe overscan, system timing frequencies, vertical blanking).

hot spot

(auch: activation point) Der »heiße Punkt«, die aktive Stelle des Intuition-Zeigers, welche sich im Aktivierungsbereich eines $\rightarrow$ gadget, $\rightarrow$ Menü oder $\rightarrow$ Icon befinden muß, damit eine Mauswahl erfolgen kann. Bei der Definition eines neuen Zeigers mit $\rightarrow$ EditPointer (Preferences), muß diese Stelle markiert werden.

hue

Farbton. Die Farbvielfalt des Amiga mit 4096 Farben zu beschreiben, ist insofern unkorrekt, als es sich dabei um 4096 Farbtöne handelt, die aus den Grundfarben Rot, Grün und Blau gebildet werden.

Eine besondere Variation des Themas »hue« ist das sogenannte »artifacting«. Hierbei entstehen als Folge der Interaktion benachbarter Farbtöne neue, in den Farbbregistern nicht enthaltene Farbtöne. Aufsteiger vom VC 20 kennen vielleicht diesen Effekt vom Superexpander her: bei der zufälligen Farb-

auswahl für die COLOR-Parameter entstehen verschiedene Grautöne, die vom System aus eigentlich nicht vorhanden sein können. Diese Grautöne sind eine Interferenzerscheinung am Bildschirm, wie der begleitende Störton im Lautsprecher beweist.

hunk

(Teilstück, Segment) Ein Hunk ist ein Teil eines Objektcodemoduls. Je nach Inhalt unterscheidet man Hunks für Programmbefehle (Code), (initialisierte) Daten und nicht initialisierte Daten. Für besondere Zwecke eingesetzt werden Relokationshunks für die Adressenanpassung beim $\rightarrow$ scatter load, symbol_hunks mit Listen von externen Symbolen, debug_hunks mit Symboltabellen und sonstigen für die Fehlersuche erforderlichen Daten. Am Anfang eines $\rightarrow$ load file befindet sich ein hunkheader mit Daten für die Laderoutine (DOS-Loader); gegebenenfalls kommt noch ein hunk_overlay mit der $\rightarrow$ overlay-Tabelle hinzu. Näheres über diese Bestandteile einer binären Datei ist dem Amiga-DOS-Manual (Technical Reference) zu entnehmen.

icon

(Sinnbild, Piktogramm) Das grafische Symbol für ein Verzeichnis oder eine Datei auf der Workbench. Man unterscheidet fünf verschiedene Icontypen: Diskette, Schublade, $\rightarrow$ Tool, $\rightarrow$ Project und Trashcan (der Mülleimer). Das Diskettenicon und der Mülleimer symbolisieren besondere Verzeichnisse: das root directory und die Ablage für überflüssige Dateien. Im Gegensatz zur Menüoption »delete« erlaubt die Verwaltung des Mülleimers, »weggeworfene« Dateien wieder aus dem Eimer zu holen, um sie weiter zu verwenden. Das funktioniert jedoch nur so lange, wie dieser Eimer nicht mit »empty trash« entleert worden ist. Das Diskettenicon und der Mülleimer können von der Workbench aus nicht gelöscht werden.

Die anderen Icons symbolisieren Verzeichnisse (user directory), Programme oder Programmdateien. Dem Diskettenicon, Trashcan oder Schubladenicon ist immer ein Window zugewiesen, in dem der Inhalt dieser Verzeichnisse – falls mit Icon versehen – sichtbar wird.

IconEd

Das Icon-Editierprogramm der Workbench. Mit diesem Tool kann man eigene Icons

(Bildsymbole) definieren, indem ein bereits bestehendes Icon in einen der Editierrahmen geladen und dann bearbeitet wird. Der zu ladende Icontyp muß dem Programm mitgeteilt werden. Dies geschieht entweder mit dem Namen oder, bei einem Diskettenicon, mit der Bezeichnung DISK. Wie beim Laden eines Icons von einem externen Laufwerk muß gegebenenfalls der vollständige $\rightarrow$ path-Name angegeben werden. Das neu erzeugte Icon übernimmt die Systemdaten des alten Icons und ist somit auch vom selben Typ (Project, Tool..).

Neben dem IconEd gibt es weitere Editoren. Erwähnenswert sind: Zap-Icon und SetAlternate (auch als Icon-Merge zu finden). Zap-Icon besitzt eine Funktion, mit der sich »brushes« aus Deluxe Paint als Icons speichern lassen. SetAlternate kann Icons mit unterschiedlichem Aussehen für die Darstellung vor und nach einem Mausklick (select status) erzeugen.

Hierzu werden zwei getrennte Icons miteinander verkettet. Jeweils eines für den normalen und den $\rightarrow$ highlight-Zustand. Ein weiteres Hilfsprogramm für diese Zwecke heißt IconExec. Mit diesem läßt sich ein Programm ohne Icon (z.B. ein CLI-Befehl oder ein $\rightarrow$ batch file) über ein Icon aufrufen. Dieses Programm befindet sich zusammen mit SetAlternate auf der $\rightarrow$ Fish Disk # 12.

iconic interface

Eine mit Icons betriebene Benutzeroberfläche (wie Workbench).

icon.library

Die Sammlung der Systemroutinen, mit denen die Icons erzeugt, dargestellt und verwaltet werden.

IDCMP

(Intuition Direct Communication Message Port) Ein Subsystem, das einem $\rightarrow$ Window zugeordnet wird, um die Kommunikation zwischen diesem Window angehörenden Programmteilen und anderen Routinen oder $\rightarrow$ devices zu verwalten. Ein IDCMP besteht aus zwei $\rightarrow$ message ports, jeweils eins beim Programm und eins bei Intuition. Das Programm bekommt auf diesem Wege $\rightarrow$ input events wie Texteingaben, Mausbewegungen, Fensterveränderungen und sonstige Aktionen mitgeteilt. Da die Daten vom input.device und nicht vom console.device kommen, sind

es »rohe« Werte in den system-eigenen Codesequenzen, die vom Programm noch aufbereitet werden müssen. Informationsträger bei dieser Kommunikation ist eine Datenstruktur namens -> IntuiMessage.

IFF

(Interchange File Format) Ein in erster Linie von -> EOA entwickeltes Dateiformat für den Amiga, welches die Übertragbarkeit (portability) von Dateien zwischen unterschiedlichen Anwenderprogrammen gewährleistet. Grundlage des Formats sind sogenannte -> chunks. Solche Chunks bestehen aus einzelnen Datentypen (Text, Bilder...) und bilden aneinandergereiht die IFF-Datei. Eine ausführliche (englischsprachige) Dokumentation befindet sich auf der -> Fish Disk 16; Ergänzungsmaterial auf Fish Disk 44. Diese Informationen entsprechen der offiziellen Commodore-Dokumentation.

Gelegentlich taucht auch die falsche Bezeichnung »Interleaved File Format« auf. Dieser Name ist darauf zurückzuführen, daß der chunk-Typ »interleaved bitmap (ILBM)« als erster der breiten Öffentlichkeit bekanntgemacht wurde und irrtümlicherweise als Modell für alle IFF-Dateien angesehen wurde.

illustration data types

Die Grafikroutinen von Intuition. Das sind Unterprogramme zur Gestaltung von Menüs, Gadgets und Requestern sowie Routinen zum Zeichnen von borders, images und text:

- Border (Umrahmung, Umrandung) ist eine allgemeine Zeichenstruktur für Polygone, mit der man einfache Linien, regelmäßige und unregelmäßige Figuren darstellen kann.

- Image (Bild) ist die Struktur für die farbliche Darstellung eines Gegenstandes am Bildschirm; anders als bei border erfolgt die Darstellung nicht linien-, sondern pixelorientiert.

- Text (IntuiText) ist die Struktur für die Darstellung von Text. Anders als bei den meisten Computersystemen wird Text beim Amiga als ein Element der Grafik behandelt.

Die obengenannten Strukturen sind Datenbereiche, welche die Parameter für das darzustellende Objekt enthalten. Sie werden alle von den -> graphics primitives unterstützt. Die illustration data types können allerdings zu verketteten Listen zusammengestellt werden. Dadurch lassen sich mehrere Gra-

fikelemente derselben Art mit einem einzigen Intuitionaufruf erzeugen.

image shadow

-> masks.

immediate mode

Der »Direktmodus« von Amiga-Basic und dem CLI. Jeder eingegebene Befehl (unter Amiga-Basic im Ausgabefenster) wird nach Betätigung der RETURN-Taste sofort ausgeführt (siehe interactive mode). Nach der Ausführung befindet sich der Amiga wieder im immediate mode und wartet auf einen weiteren Befehl.

Bei einigen CLI-Befehlen (insbesondere RUN) ist darauf zu achten, daß der Befehl auf alle Fälle sofort ausgeführt wird und nicht mit einem »?« für die Ausgabe eines -> command template von der Ausführung zurückgehalten wird.

include file

(auch header file) Include- oder Headerfiles enthalten Daten (Konstanten, Strukturdefinitionen, Deklarationen externer Variablen, Makros...), die dem jeweiligen Sprachtyp gemäß definiert werden. Beispiel: Mit der Include-Datei <intuition.h> wird die Datenstruktur »window« definiert. Diese Struktur enthält unter anderem Daten über Länge, Breite und Art des Fensters. Bindet der Programmierer diese Datei mit #include <intuition.h> in sein Programm ein, so wird mit dem Laden und Ausführen dieser Datei die Struktur definiert. Der Programmierer erspart sich selbst damit die Definition und kann mit der Anweisung »Mein-Fenster STRUCT window;« eine eigene Variable von der Art einer Window-Struktur definieren. Das Beispiel zeigt, daß sich das Anlegen einer Include-Datei nur dann lohnt, wenn die jeweiligen Definitionen in mehreren Programmen verwendet werden können.

Bei der Angabe des Dateinamens in spitzen Klammern durchsucht der Compiler nicht das aktuelle Verzeichnis nach der Datei. Dies kann aber durch Einschluß des Namens in Anführungszeichen erzwungen werden.

info

CLI-Befehl oder Workbench-Menüfunktion.

Als CLI-Befehl liefert INFO Informationen über die augenblickliche Diskettenausstattung des Systems. Angezeigt werden die verfügbaren Units (dfx:,

jhx:, ram:) mit ihrem maximalen, belegten und freien Speicherplatz, die Anzahl der vom -> Diskvalidator festgestellten Lesefehler (soft errors), der Status (Schreibschutz aktiviert oder nicht) und Name der verfügbaren -> Volumes (Disketten). Zu den »verfügbaren« (mounted) Volumes zählen die SYS-Diskette (-> system disk) sowie alle anderen Disketten, von denen Verzeichnisse (mit dem ASSIGN-Befehl) oder Daten (als Inhalt des Diskpuffers) noch im System eingebunden sind.

Anders als der CLI-Befehl gibt die Menüfunktion Info von den mit einem Icon versehenen Dateien Informationen aus. Bei einem Disk-Icon enthält das »Info-Blatt« bis auf Angabe der Lesefehler dieselben Daten wie die Ausgabe des CLI-Befehls. Bei Tools und Projects werden angezeigt der Dateityp (Project, Tool), die Größe der Datei in Bytes und Blocks, die Größe des notwendigen Stapelspeichers (Stack), der Status (Löschschutz aktiviert oder nicht), der Dateikommentar und Angaben über -> default Tool und Tool Type. Das Feld »Tool Types« wird oft verwendet, um dort allgemeine Informationen oder Übergabeparameter für die default Tools zu vermerken.

.info file

(dot info) Die Icondatei einer Datendatei. Sie enthält die zur Darstellung des Icons und zum Starten des Programms von der Workbench erforderlichen Informationen für das Betriebssystem: Icontyp (einschließlich -> Kickstart-Diskette), Position des Icons am Bildschirm, das Fenster des Icons mit der Position untergeordneter Icons, den Stackumfang und anderes. Um ein Icon »verschwinden« zu lassen, braucht man nur die Dateikennung »info« zu ändern (Beispiel: .noinfo) oder die Infodatei zu löschen.

info.library

Eine Sammlung von Systemroutinen, mit deren Hilfe die Icondateien verwaltet werden (auch als -> icon.library bekannt).

input.device

(Eingabeschnittstelle von Intuition) Eine übergeordnete Systemschnittstelle, die Eingaben aus mehreren anderen .devices (gameport, trackdisk, keyboard) sowie von Peripheriegeräten zu einem einzigen »input stream« (Eingabedatenstrom) vereinigt und entsprechende »input

events« signalisiert. Intuition überprüft den input stream zunächst auf relevante Daten und leitet ihn dann an andere daten-erwartende Routinen weiter. Der Datenstrom wird nach Entnahme der relevanten Daten an die nächste Routine in der Verarbeitungskette über das -> IDCMP-Subsystem oder -> console.device weitergeleitet. Die erforderlichen Systemroutinen sind in den Libraries -> CLIST und -> CSTRING zu finden.

interactive mode

Der »Gesprächsmodus« des CLI. Ähnlich wie beim -> immediate mode werden Eingaben nach Betätigung der RETURN-Taste sofort ausgeführt. Allerdings kehrt der Amiga nicht in den Befehlsmodus zurück, sondern bleibt im Eingabemodus des aktiven Befehls, das heißt, der Befehl oder das Programm kann ohne neuen Ladevorgang weitere Eingaben verarbeiten. Einige CLI-Befehle verfügen über beide Ausführungsmodi.

interlace mode

Verdoppelung der vertikalen Auflösung durch Zerlegen eines Bildes in zwei Teilbilder. Im normalen Betrieb zeichnet der Elektronenstrahl des Monitors ein Bild mit 200 Zeilen (256 Zeilen bei der PAL-Version). Diese Zeilenzahl ist nahe an der Grenze dessen, was der Elektronenstrahl relativ preisgünstiger Monitoren während des vertikalen Bildschirmlaufs darstellen kann. Im Interlacemodus besitzt ein Bild 400 Zeilen. Um diese auf dem Bildschirm sichtbar zu machen, zeichnet der Elektronenstrahl zunächst alle ungeraden Zeilen (1,3,5...), springt dann wieder nach oben und beginnt zwischen der ersten und dritten Zeile mit den geraden Zeilen (2,4...). Durch diese Technik werden pro Sekunde statt 50 kompletten Bildern (PAL-Version) im Normalmodus nur noch 25 dargestellt. Das Auge registriert diese Minderung der Geschwindigkeit als Flackern.

Ein mit Interlace erzeugtes Bild wirkt bei normaler Darstellung »zusammengestaucht«. Dieser Modus kann über -> Preferences eingeschaltet werden. Der Modus wird aber erst nach einem -> reset aktiv. Damit ist Funktion »use« wirkungslos und »save« muß gewählt werden.

interpreter

Ein Hilfsprogramm, welches die Zeilen eines in einer höheren Programmiersprache ge-

schriebenen Programms nicht wie ein -> Compiler in Maschinencode übersetzt, sondern nach Erkennen eines Befehls in eine Programmroutine verzweigt, die diesen Befehl bearbeitet. Aufgrund der in jeder Zeile notwendigen »Interpretation« des Befehls, der Umwandlung der Variablenamen in Speicheradressen bzw. ASCII-Konstanten in Fließkomma- oder Integerwerte ist ein mit einem Interpreter laufendes Programm sehr viel langsamer als compilierte Programme. Der Vorteil eines Interpreters ist der fehlende Aufwand für das Übersetzen und Linken des Quellprogramms. Das verführt jedoch nicht nur Programmieranfänger dazu, sich beim Programmieren weniger Gedanken zu machen, da man sich ja »schnell« mal ansehen kann, wie das Programm abläuft.

Intuition

Das »user interface«, die Benutzeroberfläche des Amiga. Intuition ist die oberste Ebene des Betriebssystems. Für den Anwender durch die Verwendung von -> icons, -> Menüs, -> gadgets, -> requesters, -> screens und -> windows am einfachsten zu bedienen, weil keine Spezialkenntnisse über Programmiersprachen oder Computer erforderlich sind. Auch bei der Programmierung nehmen die Routinen der »intuition.library« dem Programmierer viel Verwaltungsarbeit ab. Deshalb sollten sich Einsteiger zunächst mit dieser Seite des Amiga befassen.

intuition.library

Die Sammlung der Routinen, welche die Benutzeroberfläche Intuition verwalten. Wie aus dem entsprechenden -> .fd file ersichtlich, ist dies ein Bereich des Betriebssystems, der bei jeder neuen Version ständig verfeinert und erweitert worden ist.

IntuiMessage

-> IDCMP.

IntuiText

-> illustration data types.

IntuiTicks

Die Zeitmeßeinheit von Intuition. Streng genommen werden 50 IntuiTicks pro Sekunde (laut Reference Manual) erzeugt. Unter Version 1.1 jedoch erfolgten diese doppelt so schnell. Ab der Version 1.2 ist dieser Mangel behoben.

i/o redirection

Die »Umleitung« von Ein-/Ausgabeoperationen der CLI-

Befehle. In der Regel benutzen die CLI-Befehle für die Eingabe die Tastatur und für die Ausgabe den Bildschirm. Durch die Angabe eines Dateinamens im Anschluß an die beiden Richtungsanzeiger »>« (Ziel, neues Ausgabegerät) und »<« (Quelle, neues Eingabegerät) ist das jeweilige »Gerät« nun die angegebene Datei. Das bedeutet, daß Eingaben nun aus einer Datei gelesen bzw. Ausgaben in eine Datei geschrieben werden. Die Verwendung beider Richtungsanzeiger in einem Befehl bedeutet nicht, daß die Operation als Hintergrundtask ausgeführt wird. Dies ist nur mit dem Befehl RUN möglich (siehe pipelining). Die Gültigkeit der Umleitung beschränkt sich auf die jeweilige Befehlszeile. Die Richtungsoperatoren müssen stets vor allen erforderlichen Parametern der Befehlszeile - also direkt hinter dem Befehlswort - stehen.

ISO

(International Standards Organization) Das internationale Gegenstück des -> ANSI; 1946 gegründet. Die Vorschrift ISO-7 (bzw. ISO 646-1973) entspricht nahezu der 7-Bit-Zeichendefinition des -> ASCII, hält aber bestimmte Codes für nationale Sonderzeichen frei. Diese Vereinbarung wird bei den verschiedenen nationalen Zeichensätzen der (Matrix-)Drucker angewendet. Die ASCII-konforme Codierung des Zeichensatzes wurde beim Amiga voll übernommen. Die erweiterten Steuersequenzen richten sich teilweise nach ISO und teilweise nach -> DEC.

Janus

Der römische Gott der Türen und Torbögen; in der Regel als zweigesichtige (glattrasiert und bärtig) Gestalt dargestellt. Beim Amiga wird mit Janus eine Schaltung bezeichnet, welche die »Tür« zwischen dem Amiga 2000 und einer hardwaremäßigen PC-Erweiterung (-> Sidecar, bridge board) verwaltet.

Die Janus-Schnittstelle besteht aus drei Sonderchips auf der PC-Karte und dem Amiga-Controller. Beide Systeme kommunizieren über einen als »dual port RAM« bezeichneten Bereich mit einem Umfang von 128 KByte. PC-Werte werden hier in drei verschiedenen Formaten (Byte, Wort, Langwort) in entsprechend vorgesehene Speicherbereiche abgelegt. Da die Adressen dieser Bereiche weit auseinander liegen, benötigt diese Schnittstelle insge-

samt 2 MByte im oberen Bereich des -> fast RAM (siehe memory map). Ein auf der PC-Seite angeschlossenes Festplattenlaufwerk trägt als Kennung die Bezeichnung JH0: oder JH1: (statt DH0:/DH1:). Auf diesen Platten kann auch der Amiga eine »partition« (einen Teil) benutzen.

janus.library

Die Sammlung der Routinen für die Verwaltung der Janus-Schnittstelle.

jiffy

Die von DeLuxe Video verwendete, 1/60 Sekunden andauernde Zeitmeßeinheit (siehe tick, IntuiTick).

jitter

-> hires flicker.

job

Eine vom Computer zu erledigende Arbeit. Diese Bezeichnung gilt hauptsächlich für Rechenanlagen der großen und mittleren Datentechnik (mainframe) und beschreibt die Ausführung eines Programms bzw. einer Reihe von miteinander verknüpften Routinen, die - in der richtigen Reihenfolge abgearbeitet - eine sinnvolle datenverarbeitende Sequenz ergeben. Auf den Amiga bezogen wäre ein Job etwa die Erstellung eines Dokuments mit einem Textverarbeitungsprogramm, ein Compilervorgang oder das Kopieren einer Datei oder Diskette.

kernel

Der Kern bzw. die unterste Ebene des Betriebssystems. Die Routinen, welche die Hardware-Resources des Computers direkt ansprechen. Beim Amiga gehören hierzu alle -> WCS-residenten Routinen, die bis Version 1.1 von der Kickstart-Diskette eingelesen werden mußten bzw. ab Version 1.2 im ROM (nur A500 und A2000) untergebracht sind. Der Kernel-Speicher belegt zusammen mit dem -> boot ROM die obersten 256K der -> memory map.

Kerning

Der Abstand zwischen dem ersten Punkt einer Zeichenmatrix und dem ersten darzustellenden Punkt des eigentlichen Zeichens. Hierdurch ergibt sich der Abstand eines Zeichens vor dem vorangegangenen Zeichen. Das ist beim Amiga deshalb von Bedeutung, da die Zeichen am Bildschirm normalerweise proportional dargestellt werden. Durch Kerning ist der Abstand der Zeichen voneinan-

der - egal ob es sich um ein I oder M handelt - immer gleich. Das kerning kann als Variable bei der Zeichensatzdefinition angegeben werden (siehe font).

keyboard

Die Tastatur des Amiga. Diese ist selbst ein (Mini-)Computersystem. Der eingebaute Prozessor - ein 6500/1 - läuft mit 1,5 MHz. Dazu kommen das »System ROM« mit 2 KByte, vier Ein-/Ausgabeports für die Datenübertragung zum Amiga und 64 Byte RAM als Tastaturpuffer für etwa zehn Zeichencodes. Die Übertragung erfolgt mit 17 KBit/s. Da die Tastatur beim Einschalten einen Selbsttest durchführt, kann sie auch an einem bereits eingeschalteten Amiga angeschlossen werden. Die Ergebnisse des Selbsttests werden durch ein codiertes Blinken der Leuchtdiode an der Caps-Lock-Taste angezeigt. Liegt kein Fehler vor, so leuchtet diese Anzeige nur einmal kurz auf. Andernfalls haben die Blinkzeichen folgende Bedeutung: einmal - ROM Prüfsummenfehler; zweimal - RAM-Test-Fehler; dreimal - Tastaturtimer-Fehler; viermal - Kurzschluß in der Tastaturmatrix. Die Mengenangaben gelten pro Sekunde.

Die Tasten selbst sind mit Ausnahme der verschiedenen Steuertasten (SHIFT, ALT, AMIGA und CTRL) in einer 6 (Reihen) x 15 (Spalten)-Matrix angeordnet. Die Steuertasten werden über einen eigenen Port übertragen. Beim A1000 sind acht der 90 möglichen Tasten nicht benutzt, beim A500 und A2000 nur drei, weil die Tastatur dieser Computer erstens einen mit allen mathematischen Operatoren ausgerüsteten Zeichenblock und zweitens zwei zusätzliche Tasten (zwischen SHIFT links und <z> sowie rechts von RETURN) erhalten haben (zumindest in der europäischen Ausführung). Nicht in der Matrix enthalten ist die CAPS LOCK-Taste, die besonders verarbeitet wird.

Die von der Tastatur erzeugten Codes sind 7-Bit-Werte. Das achte Bit wird benutzt, um das Loslassen der Taste zu signalisieren. Der alternative Zeichensatz wird mit der ALT- oder SHIFT-ALT-Kombination erreicht. Die Shift-Tasten erzeugen eine eigene Codesequenz. Dieser Sequenz wird bei der Verarbeitung der -> input events in Verbindung mit den -> keymaps der entsprechende Code der gedrückten Tastenkombination zugeordnet.

Das Anschlußkabel der A1000-Tastatur ist ein amerikanisches Telefonanschlußkabel mit einem Stecker vom Typ RJ11 und soll durch ein längeres Kabel desselben Typs ersetzbar sein. Berichten aus Amerika zufolge ist der Versuch, eine solche Verlängerung selbst zu machen, nicht zu empfehlen, da ein Kurzschluß beim Anbringen des Steckers verheerende Folgen für den Ein-/Ausgabebaustein (CIA 8520) des Amiga hat. Wegen des geänderten Anschlußsteckers und der erweiterten Tastaturbelegung ist der Anschluß der Tastatur des A2000 am A1000 nicht zu empfehlen.

keyboard.device

Die Sammlung der Systemroutinen für die Verwaltung der Eingaben von der Tastatur (siehe console.device).

keymap

(Tastaturbelegungsdatei). Diese Datei enthält die Definitionstabelle für die Umwandlung der »rohen« Steuersequenzen, die von der Tastatur kommen, in darstellbare Zeichen oder Steuercodes für die Bildschirmdarstellung.

Mit diesem Verfahren kann jeder Taste (mit Ausnahme der Shift-Tasten) eine bis zu sieben Zeichen lange (Steuer-)Sequenz zugeordnet werden. Zeichenketten mit einer Länge bis zu 32 Zeichen lassen sich auf einzelne Tasten legen (»hot keys«).

Die »Standardbelegung« (usa0) ist im ROM enthalten. Alle weiteren Belegungen müssen sich als entsprechende Datei im Verzeichnis »keymaps« befinden. Die Belegung »usa2« ist für das Dvorak-System (siehe setmap). Jedem Fenster kann eine andere keymap und damit ein anderer Zeichensatz zugeordnet werden.

keyboard shortcuts

(Tastaturabkürzungen)

1. Die Benutzung der rechten Amiga-Taste in Verbindung mit einer alphanumerischen Taste, um Menüfunktionen zu wählen, ohne das Menü selbst abzurufen. In der Regel sind diese Abkürzungen programmspezifisch. Das Intuition Reference Manual enthält auf Seite 229 eine Vorschlagsliste mit Standardkombinationen für Workbench-Programme. Diese sind weitgehend den shortcuts des Notepad ähnlich.

2. Bei der Benutzung der linken Amiga-Taste zusammen mit einer alphanumerischen Taste wird eine bestimmte Aktion

ausgelöst. Als Systemfunktionen sind die beiden Kombinationen <Amiga N> und <Amiga M> realisiert. Diese setzen die Workbench vor oder hinter den sichtbaren Screen. Leider hat diese Funktion einen Fehler.

keyword

(Schlüsselwort, reserviertes Wort, Pflichtbefehl) Jedes Befehlswort einer Programmiersprache wird als Schlüsselwort bezeichnet. Solche Wörter können nicht als Variablenname verwendet werden. Innerhalb eines CLI-Befehls dienen Schlüsselwörter der eindeutigen Identifizierung eines Parameters (Beispiel: die Worte drive und name beim FORMAT-Befehl). Während die Parameter in der Regel an die im -> argument template beschriebene Reihenfolge gebunden sind, können mit Schlüsselwort eingeleitete Parameter an beliebiger Stelle stehen. Im template werden solche »Pflichtbefehle« mit dem Zusatz »/K« gekennzeichnet (siehe special characters).

Kickstart Disk

(der »Tretstarter« des Amiga) Beim A1000 wird das -> ROM-Kernel von dieser Diskette in den Speicherbereich von \$FC0000 bis \$FFFFFF gelesen. Zum Zeitpunkt der ersten Auslieferung des Amiga 1000 war die Betriebssystemsoftware noch nicht fertig. Um die Ausstattung älterer Computer mit neuen Versionen des Kernels zu erleichtern, hat man es auf Diskette herausgegeben. Mit der Fertigstellung des Kernels in der Version 1.2 erhielten der A500 und A2000 ROM-Bausteine anstelle dieser Diskette (siehe bootstrap, WCS).

Kickstart Journal

Eine seit der europäischen Einführung des Amiga zweimonatlich erscheinende technische Zeitschrift, die im Auftrag von Commodore Electronics von der englischen Software-Firma Ariadne (London) für die Verteilung an registrierte Software-Entwickler in der ganzen Welt ausgegeben wird. Einzelhefte kosten 5 Pfund, die ersten sechs werden als Sonderpaket für 25 Pfund angeboten (siehe developer's kit).

kludge

(auch »hack« - zusammenstüchern) Eine schnelle, aber nicht besonders elegante Methode eine Unzulänglichkeit (vor allem bei Hardware) zu beheben. Ein Beispiel dafür ist die

Ausrüstung eines externen 5 1/2-Zoll-Laufwerks mit einem Schalter, der die Diskettenwechselleitung simuliert

L:

Ein »Library«-Verzeichnis des Systems. Es enthält -> overlays für längere Befehlsroutinen sowie die nicht speicherresidenten Teile des Betriebssystems. Im Gegensatz zu den libraries im Verzeichnis -> LIBS: werden diese jedoch nicht mehr aus dem Speicher entfernt, wenn der Amiga sie nicht mehr braucht. Das Verzeichnis L: ist für die korrekte Funktion von Amiga-DOS unentbehrlich (siehe logical device).

label

Der Name eines in Programmen enthaltenen Code- oder Datenabschnitts. Mit den Sprunganweisungen der jeweiligen Programmiersprache läßt sich die Programmausführung zu diesen Abschnitten lenken. Das Label eines Datenbereichs wird beim Zugriff auf diese Daten verwendet. Label wurden eingeführt, da alphanumerische Bezeichnungen einfacher zu merken sind als Speicheradressen. Manche Sprachübersetzer oder -> Interpreter verlangen einen »:« am Ende des Labels.

layers.library

Eine Sammlung von Routinen, mit deren Hilfe -> Intuition die Verwaltung der -> Windows steuert. Ein »layer« ist eine Darstellungsebene des sichtbaren Bildschirms. Sie sollte aber nicht mit einer -> Bitplane verwechselt werden. Die Routinen der layers.library verwalten unter anderem die Sichtpriorität sich überlagernder Windows. Verdeckte Bereiche eines Window werden nach bestimmten Kriterien (-> refresh) nach ihrer Freilegung wiederhergestellt. Dazu führen diese Routinen sogenannte »damage lists«, in denen Daten zu den verdeckten Bereichen abgelegt werden. Grundlage des Subsystems sind die sogenannten »clipping rectangles«, rechteckige Windowausschnitte, die verdeckten und sichtbaren Bereichen eines Window entsprechen.

library

(Bibliothek) Im Betriebssystem des Amiga sind Programmroutinen eines Aufgabenbereiches in einem library genannten Modul zusammengefaßt. Eine solche Bibliothek ist eine Sammlung von Anspringadressen, ein Datenbe-

reich und schließlich die Routinen, auf welche die Anspringadressen verweisen. Diese Bibliotheken können von allen Programmen im Multitasking-System genutzt werden. Man unterscheidet zwei Typen: residente und transiente libraries. Eine residente Bibliothek ist im Programmcodex enthalten, der von der Kickstart geladen wird. Solche libraries »residieren« immer im Speicher. Transiente Bibliotheken befinden sich auf der Diskette im Verzeichnis -> LIBS: bzw. L: und werden mit dem Aufruf (Öffnen) der library in den Speicher geladen.

Die Systembibliotheken können auch von Basic benutzt werden. Der Interpreter benötigt dazu die aus den -> .fd-files erzeugten -> .bmap-files. Bis zu fünf geöffnete Bibliotheken kann Amiga-Basic verwalten.

Höhere Programmiersprachen (C besonders intensiv) nutzen oft sogenannte Compilerlibraries. Hierdurch kann der Sprachumfang des Compilers relativ klein gehalten werden. Ein Aufruf einer nicht im Sprachumfang enthaltenen Funktion wird im Objektprogramm (siehe object code) entsprechend markiert und der nach dem Compilerlauf gestartete Linker (siehe link editor) versucht diese Funktionen in der Bibliothek zu finden.

Der Aufbau dieser Bibliotheken wird jedoch vom Hersteller des Compilers individuell bestimmt und unterscheidet sich somit von den Systemlibraries.

LIBS:

Das Systemverzeichnis für »open library«-Aufrufe. Es enthält die nicht speicherresidenten system libraries sowie vom Benutzer eventuell zusammengestellte Routinesammlungen (siehe logical device).

link editor

(auch linker) Eine Hilfsroutine für Compiler und Assembler, mit der die erzeugten -> object code-Dateien zu einem lauffähigen -> load file (executable code) zusammengekoppelt werden. Der Linker bindet alle ausstehenden, externen Definitionen ein und paßt die relocierbaren Adressen dem endgültigen Format des load file an (siehe ALink, hunk). Externe Definitionen sind der Aufruf von Routinen oder Daten eines anderen Programmes oder einer Bibliothek.

linked list

(verkettete Liste) Eine Liste enthält Einträge in einer be-

stimmten Reihenfolge (Beispiel: Adressen alphabetisch sortiert). Beim Einfügen eines neuen Elements (hier Adresse) entsprechend der gewählten Sortierordnung müßten alle folgenden Elemente um einen Platz verschoben werden. Eine verkettete Liste enthält nummerierte Einträge in beliebiger Folge. Um dennoch eine sortierte Reihenfolge zu gewährleisten, enthält jeder Eintrag einen Verweis (Zeiger) auf die Nummer des in der Sortierfolge nächsten Eintrags. Der Zeiger des nach der Sortierfolge letzten Eintrages (er kann sich irgendwo innerhalb der Liste befinden) ist meist Null. Neue Einträge können nun einfach an das Ende der Liste angefügt werden. Statt einer Verschiebung werden lediglich die Zeiger korrigiert.

Das Prinzip der verketteten Liste spielt auf allen Ebenen des Betriebssystems des Amiga eine zentrale Rolle. Eine solche verkettete Liste besteht aus einem Listheader, dessen Adresse als Bezugspunkt für die ganze Liste gilt, und einer unterschiedlichen, wechselnden Anzahl von »nodes« (Einträgen). Die Adresse des derzeit ersten und letzten node wird im Header abgelegt. Die einzelnen nodes enthalten Zeiger auf die Adressen der unmittelbaren Nachbareinträge (nach vorn und hinten), den Datentyp, die Priorität und den Inhalt des Eintrags.

Mit den entsprechenden Verwaltungsroutinen von `exec` lassen sich diese Listen nach dem LIFO-(last in, first out) oder FIFO-(first in, first out)Prinzip gestalten. Auch nach Priorität (z.B. bei Tasks) verkettete Listen sind möglich, werden dann aber unter Umständen `queues` genannt. Mit den verschiedenen Listentypen führen sowohl Exec als auch Intuition Buch über das Geschehen im Computer.

load file

(lauffähige Programmdatei) `->` link editor.

lock

`->` file lock, filing system.

lockup

(Aussperrung) Eine besondere Form des Absturzes. Der Computer weigert sich, weitere Eingaben anzunehmen. Wahrscheinliche Ursachen beim Amiga können sein: a) ein `->` deadlock oder deadly embrace. b) fehlerhafte Programmierung (das Programm steckt in einer

Endlosschleife). c) disk swap lockup: nach dem Kopieren einer Diskette und dem Versuch, mit der Kopie weiterzuarbeiten, verlangt der Computer abwechselnd Original und Kopie. Die Ursache des Fehlverhaltens ist meistens nicht zu ergründen, vermutlich aber ein Konflikt in der Suche nach `->` logical devices oder fehlerhafte Übereinstimmung zwischen den Daten im Diskpuffer und den Daten auf der Diskette (»dirty track«). d) Speicherplatz erschöpft `->` wird meist durch einen Gruß vom `->` Guru begleitet. Im ungünstigen Fall wird der Videospeicher beeinträchtigt und der Totalabsturz erfolgt ohne weitere Warnung. e) Adresse des `->` message port für das aktuelle Window durch falsch abgelegte Daten »verbogen« (verändert). Eingaben werden ins Leere umgeleitet und können nicht verarbeitet werden. Die einzige Lösung einer solchen lockup-Situation ist in der Regel die unbeliebte Drei-Finger-Lösung `->` der `->` reset.

logical device

(auch logical directory oder system directory) Ein Element des `->` Amiga-DOS, um das Auffinden der Dateien für bestimmte systembezogene Zwecke oder für bestimmte Subsysteme zu erleichtern. Das jeweilige Verzeichnis wird vom System als eine besondere Diskette (volume) oder `->` physical device betrachtet. Dies ist an dem abschließenden Doppelpunkt bei der Bezeichnung dieser Systemverzeichnisse zu erkennen.

Amiga-DOS kennt sieben solche logical devices: SYS:, C:, L:, S:, LIBS:, DEVS: und FONTS:. Der Benutzer kann mit dem ASSIGN-Befehl entweder die Zuweisung dieser devices zu bestimmten Verzeichnissen ändern oder andere Verzeichnisse zu logical devices ernennen, um so z.B. einen Suchpfad (`->` path) zu vereinfachen.

long word alignment

Ein long word ist ein aus 32 Bit bestehender Wert (4 Byte). Die verschiedenen Prozessoren (Copper, Blitter usw.) des Amiga benötigen für bestimmte Operationen Daten, die an einer sog. Wortgrenze beginnen, das heißt an einer durch vier teilbaren Speicheradresse. Die genaue Positionierung von Daten an einer solchen Speicher-grenze bezeichnet man als »long word aligned« (siehe word alignment).

low pass filter

(Tiefpaßfilter) Ein besonderer Schaltkreis des Audio-Subsystems, der niedrige Frequenzen verstärkt und höhere Frequenzen herausfiltert. Der Tiefpaßfilter eines Audiokanals dient in erster Linie der Unterbindung von »aliasing distortion«. Das sind Störfrequenzen, die durch das Zusammenspiel der Samplingfrequenz und der zu erzeugenden Frequenz entstehen. Diese Störfrequenzen treten in zwei Bereichen auf, die sich durch die folgende Berechnung ermitteln lassen: Samplingfrequenz + Tonfrequenz und Samplingfrequenz - Tonfrequenz. Der Frequenzbereich des Filters reicht von 0 bis 7 kHz. Daraus folgt, daß die niedrigste Samplingfrequenz um mindestens 7 kHz höher als die höchste Tonfrequenz sein muß, sollen diese beiden Störfrequenzen nicht entstehen. Anders ausgedrückt, bei einer zu erzeugenden Frequenz bis zu 7 kHz kann diese nur bei einer Samplingfrequenz von 14 bis 28 kHz ohne aliasing distortion erzeugt werden (siehe quantization noise, sound synthesis).

magic (numbers)

(Zauberzahlen). Diese besser als Schlüsselzahlen bezeichneten Werte werden von bestimmten Subsystemen überprüft, um festzustellen, ob die angesprochene Datenstruktur zum richtigen Typ gehört. So wird z.B. bei einer .info-Datei überprüft, ob diese zu Beginn die Hexzahl \$E310 enthält `->` die Kennung für Icondaten.

mask 1

(Bitmaske) Eine bestimmte Bitfolge, mit der andere Daten logisch verknüpft werden. Vor allem vom Grafiksubsystem werden solche Masken benutzt. Beispiele:

- »RastPort.Mask« bestimmt, welche `->` Bitplanes des Screen mit Daten beschrieben werden dürfen.
- »HitMask« und »MeMask« werden bei der Kollisionsverarbeitung von BOBs und Sprites benötigt.
- »BitPattern« bestimmen bei der Verknüpfung der Maske mit den Daten der Bitmap durch den Blitter den aktuellen Bildschirminhalt.
- »ImageShadow« enthält alle gesetzten Pixel eines BOBs oder Sprites und kann mit HitMask identisch sein. Sie dient als Ausgangspunkt für eine 16-Bit-Maske, mit der Kollisionen mit Windowgrenzen festgestellt werden können.

- »PlanePick« und »PlaneOnOff« sind eigentlich Systemvariablen, funktionieren aber wie Masken:

Mit »PlanePick« wird bestimmt, welche Bitplanes aktiv sein sollen, und damit die Anzahl der verfügbaren Farben. Verschiedene Sprites mit unterschiedlichem PlanePick-Wert aber gleicher Gestalt können unterschiedliche Farben besitzen. »PlaneOnOff« bestimmt, was mit den von »PlanePick« nicht aktivierten Bitplanes geschehen soll. Dies hat wiederum Auswirkungen auf die Farbwahl, denn die mit »PlaneOnOff« bestimmten Bitplanes werden so behandelt, als wären alle Pixel der Bitplane gesetzt.

Mask 2

(Eingabemaske) Das »Formular« für die Dateneingabe. In der Regel für die Dateiverwaltung genutzter Bildschirmaufbau, bei dem die einzugebenden Felder durch Striche oder Rechtecke markiert sind.

Mathfp.library

Die Routinen für die Fließkomma-Arithmetik und die Grundrechenarten. In Assemblerprogrammen müssen diese Routinen durch die Angabe von »amiga.lib« beim Linken eingebunden werden. Die Library enthält auch die Umwandlungsroutinen für ASCII nach FFP, FFP nach ASCII und FFP nach BCD, die mit mathlink__lib.lib in C (nicht in Assembler) eingebunden werden können.

MathleeeDoubBas.library

Programmroutinen für Berechnungen mit Fließkommazahlen doppelter Genauigkeit.

MathTrans.library

Die Routinen für die »transzendente« Fließkomma-Mathematik. Transzendente Funktionen sind die trigonometrischen Funktionen (Sinus, Cosinus,...). In Assemblerprogrammen müssen die Routinen mit »mathlink.lib« eingebunden werden. Um die MathTrans.library zu benutzen, muß vom Programm zuerst die »mathfp.library« geöffnet werden.

memory

(der Speicher) Der RAM-Speicher im Computer wird allgemein »primary memory« (Primärspeicher) genannt, während die externen Speichermedien (Floppy, Festplatte, Kasette, CD-Disk, ...) als »secondary Memory« (Sekundärspeicher) bezeichnet werden. Die beiden Speichertypen unterscheiden sich in der Regel dadurch, daß

der Primärspeicher flüchtig (volatile) ist. Das bedeutet, daß die dort enthaltenen Daten beim Ausschalten des Gerätes oder einem Reset verlorengehen. Die Daten des Sekundärspeichers sind nicht flüchtig (semi)permanent, non volatile) und mittels eines Datenträgers meist von einem Gerät zum anderen zu übertragen.

memory access

(Speicherzugriff) Man unterscheidet den »memory access cycle« als die Folge der erforderlichen Operationen, um einen Datenwert aus dem Speicher zu lesen oder in den Speicher zu schreiben und die »memory access time« als Dauer dieser Operationsfolge (siehe system timing frequencies).

memory management

(Speicherverwaltung) Da der Amiga ein multitaskingfähiger Computer ist, müssen die Codesegmente überall im Speicher lauffähig sein. Das bedeutet, daß der Programmcode frei verschiebbar (relokable) sein muß. Die Stelle (Adresse), an der ein Code- oder Daten-Segment im Speicher plaziert wird, bestimmt -> Exec anhand einer -> free list und der Tasktabelle. Eine typische Situation sieht (auf einem 256-KByte-Amiga) folgendermaßen aus (Anfangsadressen der Codesegmente):

\$000EA8	file system
\$0027D2	input.device
\$004F92	trackdisk.device
\$00A8D0	CON:
\$02069E	printer.device
\$015EA8	Workbench
\$022F98	CON:

Anders als bei anderen Systemen wird beim Amiga keine »memory management unit« (Speicherverwaltungsbaustein), sondern eine Reihe von Systemlisten verwendet. Aus diesem Grund kann ein an falscher Adresse abgelegter Wert leicht das ganze System durcheinanderbringen. Ein blindes »POKE« im Speicher ist eine ziemlich sichere Methode, den Guru hervorzulocken (siehe fragmentation, heap, scatter load).

memory map

Die Belegung des von der CPU adressierbaren Speicherbereichs. Die CPU im Amiga, der MC 68000, kann mit seinem 24-Bit-Adreßbus insgesamt 16 MByte (16.777.216 Byte) adressieren. Bei der Aufteilung dieses Bereiches haben die Amiga-Entwickler sich anscheinend an das von den 8-Bit-Computern bekannte Modell mit acht Spei-

cherblöcken, diesmal mit je 2 MByte statt 8 KByte, gehalten. Zwischen dem Erscheinen des A1000 und der Modelle A500/A2000 haben die Entwickler auch diese Belegung anders gestaltet. Dadurch können 9,5 statt 8,5 MByte Speicher genutzt werden. Die derzeit aktuelle Speicherbelegung sieht folgendermaßen aus (alle Adressen in Hexwerten):

BLOCK 0:

000000 bis 03FFFF:

Die eingebauten 256 KByte des A1000.

040000 bis 07FFFF:

Die 256-KByte-Erweiterung des A1000.

Diese 512 KByte befinden sich beim A500 und A2000 ebenfalls an dieser Stelle.

080000 bis 1FFFFF:

reserviert

BLOCK 1-3:

200000 bis 7FFFFF:

Erweiterungen mit 3 x 2 MByte

BLOCK 4

800000 bis 9FFFFFF:

Erweiterung von 1 x 2 MByte oder Sidecar bzw. Janus oder sonstige Hardwareerweiterungen, die mehr als 4 KByte Speicher für die Einbindung in das System benötigen (z.B. die Festplatte).

Bei Sidecar bzw. Janus wird der Bereich folgendermaßen aufgeteilt:

800000 Speicher für Bytewerte
820000 Speicher für Wortwerte
840000 Grafikspeicher (Langwortwerte)

860000 I/O Register

880000 PC ROM

8C8000 PC Festplatte

8E0000 Schnittstellen-

initialisierung

BLOCK 5:

A00000 bis BFFFFFF:

Nur in den obersten 4 KByte von den Registern der beiden 8520 I/O-Chips belegt

BLOCK 6:

C00000 bis CFFFFFF:

Ein nachträglich für den Speicherausbau freigegebener Bereich von 1 MByte. Die 512-Karte des A500 und A2000 belegt den Bereich von C00000 bis C7FFFF.

D00000 bis D7FFFF:

Für -> auto.config-Erweiterung nutzbar. Bei der Erweiterung ist darauf zu achten, daß dieser Bereich eventuell von Hardwarezusätzen belegt werden kann.

ANMERKUNG: die auto.config-Routinen füllen anscheinend den nicht ausgebauten Speicherbereich von der höchsten Adresse abwärts in Blöcken zu je 64 KByte, das heißt, eine

fortlaufende Erweiterung einer RAM-Karte muß in 64-KByte-Schritten erfolgen. Bei vorhandenem Fast-RAM wird AmigaDOS in diesen Bereich geladen und das meist knappe Chip-RAM kann für andere Zwecke genutzt werden.

D80000 bis DFFFFFF:

Nur in den obersten 4 KByte von den Registern der Sonderchips Agnus, Denise und Paula belegt.

BLOCK 7:

E00000 bis E7FFFF:

Reserviert E80000 bis EFFFFFF:

Reserviert für 128 x 4 KByte »external expansion decoder slots«.

F00000 bis F7FFFF:

Für das Betriebssystem reserviert.

F80000 bis FFFFFFF:

ROM Kernel:

A1000: 64 KByte Boot-ROM + 192 KByte WCS (Kickstart)

A500/A2000: 256 KByte System-ROM Version 1.2

Menü

Die »Speisekarte« der Funktionen eines Programms. Diese können mit der Maus ausgewählt werden. Ein Menü wird immer einem bestimmten -> Window (und damit einem Programm) oder einem -> Screen zugewiesen, obwohl die verschiedenen Funktionsleisten immer in der Titelleiste des entsprechenden Screens erscheinen. Die erscheinende Menüleiste (menu strip) gehört immer zum Programm des aktuellen Windows bzw. Screens.

Beim Amiga werden sogenannte »drop down« oder »roll down«-Menüs benutzt. Diese erscheinen »fallen herunter«, wenn der aktive Punkt des Zeigers bei gedrückter rechter Maustaste in der Menüleiste an einer Stelle steht, an der sich das Menü befindet. Eine andere Technik verwenden die sogenannten »pull down«-Menüs, die für eine Funktionsanwahl tatsächlich mit der Maus - wie eine Art Rollo - nach unten gezogen werden müssen.

Ein Menü enthält eine Liste von Steuerfunktionen (menu items), die ihrerseits auch untergeordnete Funktionen (subitems) enthalten können. Das Betriebssystem erlaubt maximal 32 Menüs mit jeweils maximal 32 Funktionen. Für jede Menüfunktion kann der Programmierer wiederum maximal 63 Nebenfunktionen definieren. Bis zu 32 Menüoptionen können auch als -> mutual exclude Optionen deklariert oder mit -> keyboard shortcuts ver-

sehen werden. Nicht auswählbare Optionen werden in »Geisterschrift« (-> ghosting) angezeigt.

Menüfunktionen können »attributes« oder »actions« ändern oder auslösen. Eine Attributfunktion wird in der Regel durch das -> checkmark identifiziert und bleibt gewählt, bis das jeweilige Attribut wieder deaktiviert wird (Beispiel: Textattribut fett, unterstrichen, kursiv bei Textverarbeitung). Eine Aktionsfunktion hat nur einmalige Gültigkeit und muß jedesmal ausdrücklich neu gewählt werden. Solche Funktionen werden auch als »hit select item« bezeichnet (Beispiel: die Funktionen Cut, Copy, Paste von Amiga-Basic).

Es gibt auch ein sogenanntes Doppelklickmenü, welches aber eigentlich eine besondere Form von -> Requester ist. Anders als beim Requester kann ein Menü verlassen werden, ohne daß eine Eingabe (= Funktionswahl) erfolgen muß.

Die Beispiele zu Menüs im -> RKM sind nicht immer korrekt. Für die richtigen Strukturen sollte man besser die entsprechenden -> include files heranziehen.

message

(Mitteilung, systeminterne Kommunikation zwischen einzelnen -> tasks, processes) Die Grundlage dieses Kommunikationssystems sind die messages selbst und die jeweiligen Sender bzw. Empfänger: die message ports. Das ganze Subsystem wird von -> Exec verwaltet. Ein »message port« ist eine Exec-Datenstruktur, deren Speicheradresse Sender und Empfänger bekannt ist. Als »public port« ist dieser mit einem Namen versehen »Briefkasten« für alle Tasks erreichbar. Diese Datenstruktur wird als -> linked list geführt. Ankommende Mitteilungen werden hinten an die Liste angehängt, die Ankunft wird durch ein -> signal oder semaphore mitgeteilt. Dem empfangenen task/process obliegt es, diese Mitteilung(en) abzuholen und zu verarbeiten.

Die Nachricht im message port des Empfängers besteht allerdings nur aus einer Speicheradresse. Diese Adresse verweist auf einen Speicherbereich des Senders, in dem die eigentliche Nachricht (message) zu finden ist. Eine message ist somit die Erlaubnis des Senders an den Empfänger, auf diese Daten zugreifen zu dürfen.

Nach der Verarbeitung der Daten schickt der Empfänger die message (die Speicheradresse) an den Sender zurück. In diesem Fall wird das empfangende message port »reply port« genannt. Dadurch wird der vorübergehend beanspruchte Speicherbereich dem »Eigentümer« wieder zurückgegeben. Erfolgt diese Rückgabe nicht, geht der Speicherbereich für den Sender verloren. Wiederholte Nichtrückgabe der Nachricht kann zu einem Systemabsturz (out of memory oder stack overflow) führen.

Das message-Subsystem steuert auch den Zugang zu den -> resources. Die laufenden Prozesse synchronisieren auf diese Weise - mit Hilfe von Exec - ihre Abläufe und Ressourcenzugriffe.

MetaComCo

Englische Softwarefirma, die - der Legende nach - erst dann von Commodore beauftragt wurde, nachdem eine amerikanische Firma fast zwei Jahre lang vergeblich versucht hatte, ein multitaskingfähiges Betriebssystem zustande zu bringen. MetaComCo soll innerhalb von drei Monaten nicht nur -> Amiga-DOS, sondern auch das ABasic geliefert haben. Obwohl ABasic von Microsofts Amiga-Basic abgelöst wurde, hat sich MetaComCo, ähnlich wie -> EOA beim -> IFF, als führender Entwickler des DOS und -> CLI etabliert. Dies nicht zuletzt deshalb, weil diese Teile des Betriebssystems weitgehend in der in Amerika selten benutzten Sprache -> BCPL geschrieben worden sind.

MFM

-> pulse code modulation.

MIDI

(Musical Instrument Digital Interface) Ein normiertes Übertragungsprotokoll für elektronisch gesteuerte Instrumente (Beispiel: Synthesizer). Der Amiga kann ein solches Protokoll unterstützen, ist aber nicht serienmäßig mit der erforderlichen Schnittstelle ausgerüstet. Entsprechende Nachrüstungen sind aber relativ preiswert. Fast alle bisher erschienenen Musikprogramme für den Amiga enthalten eine MIDI-Funktion.

minterm variable

Bei den Logikoperationen des -> Blitter eine von acht möglichen logischen Kombinationen der Datenbits aus den drei verschiedenen Quellen

(den drei »Eingabekanälen« des Blitter). Daraus ergeben sich die insgesamt 256 Logikoperationen des Blitter. Die gewählten minterms werden in das Register BLTCON0 des Blitter geschrieben (siehe Hardware Reference Manual, Seite 171 bis 177).

MMU

-> memory management.

mnemonic In der Regel bei der Assemblersprache verwendete Bezeichnung für einen Maschinensprachebefehl (MOVE, ADD...). Statt der schlecht merkbaren binären Verschlüsselung von Befehlen werden symbolische Formen verwendet, die sich der natürlichen Sprache möglichst annähern und damit die Merkfähigkeit verbessern.

modem

(Modulator-DEModulator) Ein für die Datenfernübertragung genutztes Verbindungsgerät zwischen Computer und Telefonnetz, das für den Sendevorgang digitale Signale des Computers in analoge Telefonsignale, und diese umgekehrt beim Empfang wieder in digitale Signale umwandelt.

Ursprünglich sollte der Amiga - damals unter dem Namen »Lorraine« bekannt - mit einem eingebauten Modem ausgerüstet werden. Nach Übernahme der Firma Amiga durch Commodore wurde das Modem aber aus der Planung genommen.

modulo

Der Rest bei einer arithmetischen Division. Beim Amiga werden Modulo-Werte insbesondere für den -> Blitter beim horizontalen Verschieben des Bildes benötigt. Der Wert des Modulos ist die Anzahl der Datenbytes zwischen dem letzten Byte einer Bildschirmzeile und dem ersten darzustellenden Byte der folgenden Zeile. Beim normalen Bildaufbau spielt dieser Wert eine wesentliche Rolle, denn mit dessen Hilfe werden die linear gespeicherten Daten einer -> Bitplane in das rechteckige Format für die Bildschirmdarstellung umgewandelt. Dasselbe Verfahren wird auch für die Verschiebung der -> BOBs benötigt.

motion control

Die Geschwindigkeits- und Beschleunigungssteuerung von -> GELs. Als eingebaute Animationsfunktionen bewegen die zuständigen Systemroutinen des Amiga BOBs und

Sprites gemäß eingestellter Werte für Grundgeschwindigkeit und gegebenenfalls Beschleunigung. Dieses Verfahren wird auch bei den Amiga-Basic-Befehlen OBJECT.AX/.AY und OBJECT.VX/.VY sowie den entsprechenden Funktionen angewendet. Die Geschwindigkeitsangabe wird als binäre Festkommazahl mit dem Dezimalpunkt zwischen Bit 6 und Bit 7 interpretiert. Das bedeutet, daß sich die Geschwindigkeit auf $1/64$ einer Zeiteinheit (in diesem Fall 1 Sekunde) genau regulieren läßt. Alle Bewegungen werden als Anzahl Pixel pro Zeiteinheit ausgeführt. Die Beschleunigungsangabe wird jede Sekunde zu der Grundgeschwindigkeit addiert. Vorsicht ist bei der Beschleunigungsangabe geboten, denn bei hoher Beschleunigung erreichen die GELs sehr schnell die »Fluchtgeschwindigkeit« und verschwinden auf Nimmerwiedersehen, weil die Windowgrenze zwischen zwei Positionen liegt und das System keine Zeit hat, eine Kollision zu registrieren, geschweige denn zu verarbeiten (siehe animation, sequenced drawing).

mount

(in Stellung bringen, aufstellen) Auf den Amiga bezogen: Ein externes Speichergerät in das Betriebssystem einbinden (siehe auto.config). Erst ab Version 1.2 aktuell, weil sich die entsprechenden CLI-Befehle (MOUNT, BINDDRIVERS) vorher nicht auf der Workbench befanden.

Der Begriff bezieht sich nicht nur auf Laufwerke jeder Art, sondern auch auf Disketten (siehe INFO vom CLI aus), die auch in das System eingebunden werden; entweder enthält die Diskette ein Verzeichnis, das einem -> logical device zugewiesen wurde, oder der Diskpuffer enthält Daten, die zu einer bestimmten Diskette gehören (»dirty track«: die Daten sind eventuell verändert, aber noch nicht auf die Diskette zurückgeschrieben worden). Ferner kann der Diskpuffer unter Umständen das »root directory« einer Diskette als »current directory« enthalten. In all diesen Fällen werden die Disketten beim CLI-Befehl INFO als »mounted« (verfügbar) gemeldet.

mount list

Datenstruktur für die Einbindung eines externen Speichergeräts in das System. Die Ein-

träge in der Datei »mountlist« enthalten Angaben, die vom System bzw. dem zuständigen -> .device benötigt werden, um das neue Gerät in die entsprechende Systemliste einzutragen.

mouse

(deutsch: Maus) Das Hauptsteuergerät des Amiga unter -> Intuition. Eine Maus ist ein auf den Rücken gedrehter Trackball-Controller mit bis zu drei Bedienungstasten anstelle der Feuerknöpfe. Die Amiga-Maus hat zwei Tasten. Das Betriebssystem kann bei entsprechender Installation Mousausführungen mit bis zu drei Tasten unterstützen.

In erster Linie steuert die Maus den Mauszeiger (mouse pointer oder window pointer). Mit den beiden Tasten werden entweder Menüoptionen gewählt (rechte Taste = select key) oder irgendeine Manipulation am Bildschirm vorgenommen (Icon, Screen oder Window verschieben) bzw. irgendeine Tätigkeit (Programm starten, requester aktivieren) ausgelöst (linke Taste = action key). Den Maustasten sind die beiden Amiga-Tasten zugeordnet (rechts: Selektion; links: Aktion).

Das Betätigen einer Maustaste wird als »klicken« bezeichnet. Dabei unterscheidet man zwischen »anklicken« (einmalige Betätigung) und »doppelklicken«. Es kann bei manchem Programm aber auch vorkommen, daß man beide Tasten gleichzeitig betätigen muß. Dafür gibt es allerdings noch keine gängige Bezeichnung.

Bei der Wahl einer Menüfunktion genügt es in der Regel, das Symbol mit der Maus anzusteuern und - sobald sich das entsprechende Symbol optisch ändert (-> highlight) - die rechte Taste loszulassen. Ein Sonderfall ist die »Doppelklickfunktion«, mit der ein -> requester abgerufen werden kann.

Die Steuerkugel der Maus ist gegen Dreck und klebrige Substanzen empfindlich. Auf Hochglanzoberflächen und bei zu großem Druck auf das Gehäuse der Maus gleitet die Kugel anstatt zu rollen. Dagegen werden sogenannte »mouse pads« angeboten. Eine normale Schreibtischunterlage erfüllt denselben Zweck.

Angeschlossen wird die Maus am sogenannten »mouse port« (eigentlich gameport 1). Die Auswertung der Maustätigkeiten erfolgt über das -> input.device und -> Intuition.

MS-DOS

(Microsoft Disc Operating System) Von Microsoft in Seattle/USA entwickeltes Betriebssystem. Ist auf Personal-Computern sehr verbreitet und wird daher auch als PC-DOS bezeichnet.

multiprocessing

Ursprünglich bedeutete dieser Begriff die Verwendung von mehreren, parallel geschalteten CPUs, um die Datenverarbeitung zu beschleunigen. Neuerdings geht man mehr dazu über, dieses Verfahren (genauer) »parallel processing« zu nennen und den Begriff »multiprocessing« für die Beschreibung von Systemen zu benutzen, in denen eine CPU durch mehrere untergeordnete Hilfsprozessoren unterstützt wird.

In diesem Sinne ist der Amiga ein Multiprozessorsystem. Operationen, die sonst die CPU bearbeitet hat, werden von Hilfsprozessoren übernommen: Bildschirmprozessor Copper; Animations- und DMA-Prozessor Blitter; Audio- und I/O-Prozessor Paula, Video und I/O-Prozessor Denise und Tastaturprozessor 6500/1. Dazu kommen weitere Hilfsprozessoren bei verschiedenen Hardwareerweiterungen: Festplattencontroller, die Chips der Janusschnittstelle und andere. Alle diese Hilfsprozessoren dienen dazu, die CPU zu entlasten.

Beim A2000 bietet der »processor slot« die Möglichkeit, ein echtes Parallelprozessorsystem zu realisieren. Eine parallele Datenverarbeitung ist auch in Verbindung mit dem bridge board und dem PC-seitigen Prozessor vorstellbar, vor allem, wenn dieser durch einen (x)xx 87Mathematik-Coprozessor unterstützt wird.

multitasking

Die Fähigkeit eines Computers (scheinbar), mehrere Programme gleichzeitig ablaufen zu lassen. Im weiten Sinne des Begriffes wird auch die Fähigkeit, mehrere Benutzer gleichzeitig arbeiten zu lassen, eingeschlossen. Hier ist aber die Bezeichnung »multiuser« (Mehrbennutzersystem) zutreffender.

Das Multitaskingsubsystem des Amiga wird von -> Exec auf der task-Ebene und von -> Amiga-DOS auf der process-Ebene verwaltet. Die Grundlage hierfür ist das -> message-Subsystem, mit dem die verschiedenen Programme synchronisiert werden, und reichum Zugang zu den -> resources erhalten (siehe timesharing).

mutual exclusion

Gegenseitige(r) Abschaltung/Ausschluß. Vor allem bei Menüfunktionen gibt es Operationen, die nicht gleichzeitig ausgeführt werden können (Beispiel: LOAD und SAVE). Durch das Setzen einer entsprechenden Flagge (eines Wertes) in der jeweiligen Datenstruktur wird dafür gesorgt, daß die Systemroutinen solche Konflikte automatisch lösen, indem eine bereits gewählte, aber kollidierende Funktion bei der Wahl der neuen Funktion abgeschaltet wird (siehe toggle).

narrator.device

Die Systemroutine für die Umwandlung von Phonemcodes in Wellenformen für die Sprachsynthese. Als Beigabe liefert diese Routine auch Werte für Höhe und Breite des Mundes bei den einzelnen Vokallauten (siehe speech synthesis).

nesting

(Verschachtelung) Das Einbetten einer bestimmten Programm- oder Datenstruktur in eine andere Struktur von (in der Regel) demselben Typ. (Beispiel: FOR..NEXT-Schleifen in Basic oder die user -> directories (Unterverzeichnisse) auf einer Diskette).

NEWCON

Dieses Gerät ist der neue Wandler für die komfortable Amiga-Shell (-> CLI). NEWCON: die Editierung einer Eingabezeile in der Shell durch die Cursortasten.

NIL

Ein besonderes -> logical device von -> Amiga-DOS. Es kann nur als Ausgabegerät benutzt werden. Eingaben von NIL: werden stets als Dateiendemarkierung interpretiert. Das Gerät hat den englischen Beinamen »bit bucket« (Abfluß oder Abfalleimer) erhalten. Über NIL: ausgegebene Daten werden einfach »weggespült«. Sinn dieser Einrichtung ist es, den Programmierern eine Möglichkeit zu geben, Programm (abschnitte) zu testen, ohne daß der Speicher mit Daten vollgestopft oder sogar überschrieben wird. So kann in Testläufen der Programmfluß ohne Berücksichtigung der Ergebnisse überprüft werden. Als Nebenaufgabe dient NIL: zur Unterbindung der ungewollten Textausgabe von -> argument templates bei der Benutzung von bestimmten CLI-Befehlen in -> batch files (LoadWB in der -> startup-sequence).

node

(Eintrag, Codesegment) Ein Eintrag in einer -> linked list oder ein Codemodul in einem binären -> load file (siehe hunk).

nondestructive readout

(NRDO) Bei einigen Registern, die man nur auslesen aber nicht beschreiben kann, führt das Auslesen zum Zurücksetzen des Registerinhalts (destructive readout: -> collision stack von Amiga-Basic). Beim NDRO bleibt der aktuelle Inhalt des Registers unverändert erhalten.

NOTEPAD

Der »Notizblock« der -> Workbench.

Er ist ein kleiner Ersatz für ein Textverarbeitungsprogramm. Er bietet unter anderem eine Seitennumerierung, frei wählbare Zeichensätze, Editioptionen (cut, copy, paste, find, replace), Blockmarkierung, automatisches »word wrap« (ein Wort, das nicht mehr in die Zeile paßt, wird vollständig in die nächste Zeile übernommen). Um von allen Zeichensätzen Gebrauch machen zu können, muß die Funktion »global fonts« ausgeschaltet werden.

Notentabelle

Das Amiga-Basic-Handbuch enthält zwar eine Notentabelle, aber es fehlen sämtliche Zwischentöne. Die folgende Tabelle enthält die volle Bandbreite, geordnet nach Frequenz und mit Angabe der Oktave und der Nummer für MIDI-Anwendungen (siehe equal tempered scale).

Oktave/ Tonlage	Frequenz (Hz)	Midi-Nr.
A	27.50000	21
A#	29.13523	22
H	30.86770	23
C	32.70319	24
C#	34.64782	25
D	36.70809	26
D#	38.89087	27
E	41.20344	28
F	43.65352	29
F#	46.24930	30
G	48.99942	31
G#	51.91308	32
	55.00000	33

Oktave/ Tonlage	Frequenz (Hz)	Midi-Nr.
	58.27047	34
	61.73541	35
C2	65.40639	36
	69.29565	37
	73.41619	38

Oktave/ Tonlage	Frequenz (Hz)	Midi-Nr.
	77.78174	39
	82.40688	40
	87.30705	41
Baß	92.49860	42
	97.99885	43
	103.82617	44
	110.00000	45
	116.54094	46
Bariton	123.47082	47
C3	130.81278	48
Tenor	138.59131	49
	146.83238	50
	155.56349	51
	164.81377	52
	174.61411	53
Alt	184.99721	54
	195.99771	55
	207.65234	56
	220.00000	57
Mezzo-Sopran	233.08188	58
	246.94165	59
C4	261.62556	60
Sopran	277.18263	61
	293.66476	62
	311.12698	63
	329.62755	64
	349.22823	65
	369.99442	66
	391.99543	67
	415.30469	68
Kammer-ton	440.00000	69
	466.16376	70
	493.88330	71
C5	523.25113	72
	554.36526	73
	587.32953	74
	622.25396	75
	659.25511	76
	698.45646	77
	739.98884	78
	783.99087	79
	830.60939	80
	880.00000	81
	932.32752	82
	987.76660	83
C6	1046.50226	84
	1108.73052	85
	1174.65907	86
	1244.50793	87
	1318.51022	88
	1396.91292	89
	1479.97769	90
	1567.98174	91
	1661.21879	92
	1760.00000	93
	1864.65504	94
	1975.53320	95
C7	2093.00452	96
	2217.46104	97
	2349.31814	98
	2489.01586	99
	2637.02045	100
	2793.82585	101
	2959.95538	102
	3135.96348	103
	3322.43758	104
	3520.00000	105
	3729.31009	106
	3951.06641	107
C8	4186.00904	108

Oktave/ Tonlage	Midi-Nr. Frequenz (Hz)
	4434.92209 109
	4698.63628 110
	4978.03173 111
	5274.04091 112
	5587.65170 113
	5919.91076 114
	6271.92697 115
	6644.87516 116
ab hier	7040.00000 117
Distorsion	7458.62018 118
C9	7902.13282 119
	8372.01808 120
	8869.84419 121
	9397.27257 122
	9956.06347 123
	10548.08182 124
	11175.30340 125
	11839.82152 126
	12543.85395 127
	13289.75032 128
	14080.00000 129
	14917.24036 130

NuDos.library

(sprich: »niudos«) Die Bezeichnung der -> dos.library im Programmcode vieler Systemroutinen ab der Version 1.2.

Nyquist sampling

(Regeln für Sampling) Beim Amiga tritt der Begriff »sampling« meistens in Verbindung mit dem Digitalisieren von Tönen (Audioidigitalisierung) auf. Ein Signal wird in bestimmten Zeitabständen (hold phase) gemessen und die ermittelten Meßwerte geben in binärer Form den Verlauf des analogen Signals wieder. Gemessen wird für den Amiga die Amplitude (Lautstärke) des Signals als Funktion der Samplingfrequenz. Nach den Nyquist-Regeln kann man die optimale Häufigkeit des Samplings bestimmen, um sicherzugehen, daß a) keine wesentlichen Werte des Eingangssignals verlorengehen und b) die statistische Unabhängigkeit der einzelnen Messungen gewährleistet wird. Grob formuliert besagen die Nyquist-Regeln, daß die Samplingfrequenz doppelt so hoch wie die höchste zu messende Frequenz des Eingangssignals sein sollte, um die beiden genannten Kriterien zu erfüllen (siehe digitizer, quantization).

object code

Ein Assembler erzeugt Code in der für den Prozessor notwendigen Maschinensprache. Solcher Code ist nicht unbedingt lauffähig, weil er noch nicht aufgelöste, externe Querverweise enthalten kann. Ein solcher Querverweis ist etwa

der Aufruf einer Routine, die sich nicht in der Quelldatei befindet. Solche werden mit Hilfe eines -> link editor eingebunden, um ein -> load file zu erzeugen.

object file

Eine Datei, die -> object code enthält.

ObjEd

(Objekteditor) Ein Hilfsprogramm für Amiga-Basic, mit dem man die Gestalt eines Sprites bzw. BOBs definieren kann. Bei der Anwendung des Programms ist darauf zu achten, daß die hohe Bildschirmauflösung (640 x 200) benutzt wird. Da Sprites aber stets in niedriger Auflösung dargestellt werden, erscheinen sie beim Editieren nur in halber Breite.

object module disassembler

(OMD) Ein Hilfsprogramm bei C-Compilern, welches eine binäre Objektcode-Datei in Assembler-Quellcode zurückverwandelt. Unter Umständen werden auch die Quellcodetexte eingefügt (siehe Handbuch zu Lattice C).

offset

Der Abstand eines bestimmten Datenwertes (Bytes) von einer vorgegebenen Basisadresse.

Die meisten Systemlibraries haben z.B. ein Offset (auch »bias«) von -30 (dezimal) zwischen Anfangsadresse der library und dem ersten Eintrag in der Sprungtabelle für die in der library enthaltenen Routinen, weil sich zwischen der Basisadresse und dem Beginn der Sprungtabelle noch ein von der library genutzter Datenbereich befindet.

operating system

Das Betriebssystem eines Computers. Beim Amiga besteht das »OS« aus mehreren Ebenen. Je höher die Ebene, desto einfacher die Bedienung für den Benutzer. Die einfache Bedienung macht die Programmierung komplizierter.

Die tiefste Systemebene bildet -> exec. Darüber liegt -> Amiga-DOS, auf das der Anwender über ein -> CLI Zugriff hat. Auf beide Ebenen baut -> Intuition auf. Die obere Ebene bilden die Anwenderprogramme unter der -> Workbench. Die Schnittstellen, an denen der Benutzer mit dem Computer in Kontakt treten kann, nennt man »user interface« (CLI) bzw. -> iconic interface (Benutzeroberfläche: Workbench).

origin

(Ursprungspunkt eines Koordinatensystems) Anders als bei der Mathematik liegt bei einem Bildschirm dieser Punkt in der oberen linken Ecke des Bildes. Dort beginnt der Elektronenstrahl mit dem Abbilden der Grafik. Beim Amiga ist darauf zu achten, daß dieser Punkt nicht immer die (Elektronenstrahl-)Koordinaten 0,0 hat, weil bei der Angabe dieser Koordinate der -> overscan berücksichtigt werden muß. Die Standardwerte für den Beginn der Bildschirmdarstellung sind die Koordinaten 64,44 (x,y). Bei der Positionierung von Sprites (insbesondere -> simple sprites) muß das beachtet werden. Ein Sprite erscheint sonst unter Umständen nicht auf dem Bildschirm.

overlay

(Überlagerungsmodul) Ein Programm oder Codemodul, das während des Programmverlaufs nachgeladen wird und einen bereits im Speicher befindlichen Teil eines Programms ganz oder teilweise überschreibt und damit unbrauchbar macht. Dieses Verfahren wird benutzt, um umfangreiche Programme in einzelne, voneinander unabhängig ablaufende Unterprogramme aufzuteilen und den Speicherbedarf des Programms zu reduzieren. Dies kann besonders für den Amiga von Nutzen sein, da damit mehr Speicherplatz für andere Programme zur Verfügung steht.

overlay file

Eine Datei, die ein -> nachzuladendes Overlay-Modul enthält.

overscan

Ein Bereich des Bildschirms, der in der Regel nicht für die Darstellung genutzt wird. Um sicherzugehen, daß das zu erzeugende Bild vollständig und ohne Unschärfen am Bildschirm erscheint, ist bei jedem Fernseher/Monitor der Elektronenstrahl so justiert, daß dieser seitlich sowie oben und unten einen größeren Bereich als für die Darstellung notwendig »abfährt«. Beim NTSC-Amiga liegt der darstellbare Bereich normalerweise zwischen der 64. und 363. Spalte (LoRes) und zwischen der 44. und 243. Zeile. Der PAL-Amiga hat einen um 56 Zeilen erweiterten Darstellungsbereich.

Hardwaremäßig läßt sich die Lage des Bildes am Bildschirm innerhalb gewisser Grenzen

ändern. Mit den Verschiebegadgets der -> Preferences kann der Anwender eine Einstellung vornehmen. Bei Verzicht auf die Sprites kann die horizontale Auflösung vergrößert werden (siehe horizontal blanking, vertical blanking, system timing frequencies).

packet

Eine besondere Form von -> message, die von Amiga-DOS benutzt wird, um die Kommunikation zwischen Amiga-DOS-Tasks zu regeln. Insgesamt gibt es 21 verschiedene Typen, die teilweise eng mit CLI-Befehlen verbunden sind. Eine ausführliche Beschreibung befindet sich im entsprechenden Kapitel des Amiga-DOS-Manual.

paper size

(bei Preferences) -> spacing.

PAR:

Die parallele Schnittstelle. Voreingestellt ist diese Schnittstelle für die Übertragung von Daten an einen mit Centronics-Anschluß ausgerüsteten Drucker.

parallel.device

Die Systemroutinen für die Verwaltung der parallelen Schnittstelle. Unter Version 1.1 kann es bei gleichzeitigem Betrieb der seriellen Schnittstelle zu Störungen an der Parallel-Schnittstelle kommen. Seit Version 1.2 ist dieser Mangel behoben.

pass through connector

Ein Stecker, der einen belegten Anschluß am Computer nicht blockiert, sondern »durchschleift«.

path

Suchpfad des -> filing systems. Die volle Namensbezeichnung einer Datei: Laufwerksnummer, erforderliche Verzeichnisnamen und Dateinamen.

Ab der Workbench Version 1.2 läßt sich mit dem Befehl PATH der Suchpfad des CLI erweitern. Der voreingestellte Pfad für CLI-Befehle umfaßt das Hauptverzeichnis (bzw. aktuelle Verzeichnis) und das Verzeichnis c. Der Befehl EXECUTE sucht Batch-Dateien zuerst im Hauptverzeichnis oder aktuellem Verzeichnis und dann im Verzeichnis s. Bei jedem anderen Dateizugriff wird ohne Angabe des vollständigen Pfades die Datei im Hauptverzeichnis erwartet.

PATH SHOW gibt den aktuellen Suchpfad auf dem Bild-

schirm aus. PATH allein stellt den Standardsuchpfad wieder her. Mit PATH ADD- und dem gewünschten Verzeichnisnamen wird der Suchpfad entsprechend erweitert. Die Einstellung kann durch weitere PATH ADD-Anweisungen ständig ergänzt werden.

pattern matching

Der Vergleich von Zeichenketten anhand eines vorgegebenen Musters. Im CLI werden zwei solche Optionen angeboten. Einmal mit dem $\rightarrow$ keyword pattern für einzelne Zeichen ($\rightarrow$?) oder eine beliebige Anzahl desselben Zeichens ($\rightarrow$ #) und für eine bestimmte Zeichenfolge ($\rightarrow$ s) (siehe special characters).

Paula

(auch Portia) Der »peripheral/audio«- oder Port-Baustein unter den $\rightarrow$ Custom-Chips. Dieser Baustein wandelt im Speicher abgelegte digitale Werte in elektrische Schwingungen um, die bei Wiedergabe über einen Lautsprecher entsprechende Klänge ergeben. Mit Paula läßt sich die gespeicherte Wellenform (Klangfarbe) beliebiger Instrumente abspielen. Um die Musikqualität zu erhöhen, ist ein störende Geräusche entfernender Tiefpaßfilter eingebaut. Der Chip hat vier Soundkanäle, von denen jeweils zwei zu einem Stereoaussgang zusammengefaßt werden.

Eine weniger bekannte Aufgabe von Paula ist die Verwaltung der Hardwareschnittstellen (Parallel-Port, Seriell-Port) des Amiga.

period

Die Periode oder Dauer eines Signals (Tons). Diese Dauer steht im umgekehrten Verhältnis zur Frequenz, die erzeugt werden muß. Der Periodenwert eines Tons wird vom $\rightarrow$ Copper in die Register der Audiokanäle geschrieben und dort als Zählerwert benutzt. Ausgehend von diesem Wert wird nach Null heruntergezählt. Die dabei benötigte Zeit beeinflusst die Tonhöhe. Je größer der Periodenwert, um so niedriger der erzeugte Ton. Eine verzerrungsfreie Tonerzeugung erfolgt mit Periodenwerten zwischen 124 (der niedrigstmögliche Wert) und 256. Werte über 256 sind zwar möglich, aber wegen der Beeinflussung durch den Tiefpaßfilter kommt es zu »aliasing distortion« (siehe system timing frequencies).

physical device

Ein-/Ausgabeoperationen verrichtende Hardwarekomponenten des Amiga. Amiga-DOS kennt folgende physical devices: DF:, DH: (JH: bei der $\rightarrow$ Janus-Schnittstelle), SER:, PAR:, PRT:, CON:, RAW:, NIL:, RAM: und $\rightarrow$ *«. Bei der Workbench Version 1.3 ist das device NEWCON: hinzugekommen. Während die meisten devices nur Einzelgeräte sind, können bei den Laufwerken mehrere »device units« (Einheiten) angesprochen werden. Der Anwender spezifiziert die jeweilige $\rightarrow$ Unit durch die Angabe einer Ziffer im Anschluß an den Namen des devices (DF0: (internes Laufwerk), DF1:, ...) (siehe logical device).

pinout

(Pinbelegung) Die Verteilung der Anschlüsse für die Systemleitungen an einer Hardwareanschlußstelle des Computers. Während der A 500 und A 2000 Standardbelegungen bei der parallelen und seriellen Schnittstelle aufweisen, weicht der A1000 von der Norm geringfügig ab (siehe Anwender-Handbuch, Zorro).

pipelining

Ein Verfahren zur Behebung zeitlicher Engpässe bei der Ausführung von Anweisungen (Beispiel: Die Ausführung einer Datenverarbeitungsoperation dauert so lange, daß zwischenzeitlich neue Eingabewerte anstehen). Da Datenverarbeitungsoperationen meistens in verschiedene Einzelschritte zerlegt werden können, wird nur ein Teil des ganzen Vorgangs von dem eigentlich zuständigen (Sub-)System erledigt und der Rest an ein anderes (Sub-)System für den Abschluß der Operation weitergegeben. Damit können neue Datenwerte ohne größeren Zeitverlust verarbeitet werden. Das Verfahren wird unter anderem bei Agnus benutzt, führt aber dazu, daß das erste Datenwort bei $\rightarrow$ DMA-Operationen verlorengeht. Ein ähnliches Verfahren auf höherer Ebene stellt das MetaComCo-TOOLKIT mit dem Befehl PIPES zur Verfügung.

pitch 1

Die Zeichendichte eines Zeichensatzes bei der Ausgabe auf einem Drucker. Die Dichte der Zeichen bestimmt die Anzahl der Zeichen, die sich auf einem bestimmten Zeilenabschnitt – in der Regel 1 Zoll (2,54 cm) – unterbringen lassen. Mit $\rightarrow$ Preferences sind drei Werte wählbar: 10 cpi (cha-

racters per inch: Zeichen pro Zoll) oder Pica, 12 cpi (Elite) und 15 cpi (Fine) mit jeweils 80, 96 und 120 cpl (characters per line: Zeichen pro Zeile). Diese Angaben gelten für ein Standardformular mit einer Breite von 8 Zoll.

pitch 2

Die Höhe (Frequenz) eines Tones. Gemessen wird diese in Hertz (Anzahl Schwingungen pro Sekunde, siehe period).

pixel

(picture element: Bildpunkt) Die Ausgangsgröße eines Bildpunktes wird beim Amiga im niedrigauflösenden Modus bestimmt. Demnach ist ein Bildpunkt mit den Abmessungen $\frac{1}{320}$ der Breite und $\frac{1}{200}$ (NTSC) bzw. $\frac{1}{256}$ (PAL) der Höhe des dargestellten Bildes definiert. Bei hochauflösendem Modus beträgt die Breite $\frac{1}{640}$ der Bildbreite. Das »aspect ratio« eines Pixel (Verhältnis Höhe zu Breite) ist also 8:5 bzw. 4:5 für NTSC und 5:4 bzw. 2,5:4 für PAL.

Ein Pixel auf dem Bildschirm entspricht einem Bit einer $\rightarrow$ Bitplane. Ist das Bit gesetzt, wird das Pixel in einer Vordergrundfarbe abgebildet. Ein gelöschtes Bit bewirkt die Darstellung in der Hintergrundfarbe.

PlanePick/PlaneOnOff

$\rightarrow$ mask 1.

playfield mode

Die Zusammenfassung mehrerer Bitplanes für eine Darstellung am Bildschirm. Anders als beim $\rightarrow$ display field ist ein Playfield nicht an die Abmessungen des darstellbaren Bildschirmbereichs gebunden. Es kann sowohl höher als auch breiter sein. Man unterscheidet Single-Playfields und Dual-Playfields. Während Single-Playfield der übliche Darstellungsmodus ist, lassen sich mit Dual-Playfield spezielle Effekte erzeugen. So kann ein Playfield z.B. die Bitmuster für das Kontrollpanel eines Flugzeuges enthalten, während aus den Planes des zweiten Playfields die Bildinformation für die umgebende Landschaft – also die Sicht aus dem Cockpitfenster – geholt wird. Durch horizontales und/oder vertikales Scrollen nur der zweiten Plane ergibt sich der Eindruck einer Bewegung des Flugzeuges.

Im Dual-Playfield-Mode ordnet das System dem ersten Playfield die Bitplanes mit den ungeraden Nummern (1, 3 und 5) und dem zweiten die restlichen (2, 4, 6) zu. Da sich die Playfields auch die Farbbregister

teilen müssen, stehen für jedes nur noch 16 Farben zur Verfügung. Im hochauflösenden Modus sind für jedes Playfield nur zwei Bitplanes verwendbar. Hier halbiert sich also die Anzahl der verfügbaren Farben ein weiteres Mal. Das Besondere am Dual-Playfield-Mode ist die gleichzeitige Darstellung beider Playfields in einem Fenster (Window), wobei diese unabhängig voneinander verschoben werden können. Die Priorität der Playfields untereinander sowie gegenüber den Sprites läßt sich je nach Bedarf bestimmen. Verweist die Bitkombination eines Pixels des Playfields im Vordergrund auf das Hintergrundfarbbregister (alle Bits 0), so erscheint der Punkt in der Farbe, die durch das zweite Playfield bestimmt wird. Damit läßt sich der oben erwähnte »Blick durchs Fenster« realisieren.

polling

Die Abfrage von Systemeinheiten nach einem bestimmten Ereignis. So kann der Prozessor z.B. durch Lesen bestimmter Register (Speicherstellen) der Ein-/Ausgabeebausteine erkennen, ob ein Datenwert zur Eingabe bereitsteht. Oft ist eine solche Abfrage in der Form einer Endlosschleife realisiert, die erst verlassen werden kann, wenn das gewünschte Ereignis eingetreten ist (auch »busy wait« oder »busy loop« genannt).

Dieses Verfahren ist beim Amiga unüblich, weil durch das laufende Abfragen wertvolle CPU-Zeit verlorengeht. Während der Prozessor nämlich für ein Programm auf bestimmte Ereignisse wartet, kann er genauso gut andere Programme bearbeiten.

Der Amiga verwendet deshalb das sogenannte $\rightarrow$ event trapping. Der Anwender definiert das Ereignis, auf das reagiert werden soll, und schickt das jeweilige Programm mit der Systemfunktion »wait()« in die Wartestellung. Diese Deaktivierung läßt sich in Amiga-Basic mit dem Befehl SLEEP auslösen. Dabei verliert das Programm den Zugang zu den $\rightarrow$ resources. Es bleibt inaktiv, bis das gewünschte Ereignis eintritt. Das Programm wird »geweckt« und von $\rightarrow$ exec wieder in das $\rightarrow$ ready queue (die Schlange der arbeitenden Programme) eingeordnet. Durch solche Verfahren können andere Routinen während dieses »Nickerchens« fast unbehindert weiterarbeiten.

Port 1

Hardware-Port (auch Schnittstelle): Die Bezeichnung einer Anschlußstelle für Peripheriegeräte. Man unterscheidet parallel port, serial port, drive port (Laufwerk), mouse port und printer port.

port 2

Software-Port: -> message port.

potgo.library

Systemroutinen für die Verwaltung von Eingaben über die Anschlüsse des game ports.

preemptive allocation

Die Zuweisung einer -> resource an den -> task mit der zur Zeit höchsten Priorität. Das bedeutet die Übernahme einer resource, bis ein anderer task mit höherer Priorität aktiviert wird, oder die dem task gestattete Benutzungsfrist (-> quantum) abgelaufen ist (siehe timesharing). Meldet sich ein task mit der höchsten Priorität (127), übernimmt er die resource voll und ganz. Damit ist Multitaskingfähigkeit nicht mehr möglich.

Preferences

(wörtlich »Vorlieben«; laut Anwenderhandbuch »der Voreinsteller«) Eine Sammlung von Systemroutinen, mit denen sich verschiedene Systemparameter einstellen lassen. Für die Änderung der Workbenchfarben stehen mehrere Schieberegler zur Verfügung, mit denen die Rot-, Grün- und Blauanteile der durch Anklicken gewählten Farbe bestimmt werden können. Mit weiteren Schaltern läßt sich Datum und Uhrzeit, Bildschirmauflösung (40/80 Zeichen, Interlace) oder das Zeitverhalten bei Mausklicks bestimmen. Mit einem Pixeleditor kann für spezielle Anwendungen ein neuer Mauszeiger entwickelt werden. Benutzer der seriellen Schnittstelle können mit Change Serial die Übertragungsgeschwindigkeit (Baud-Rate) und weitere Parameter für die serielle Datenübertragung einstellen. Change Printer ermöglicht die Einbindung einer Druckerparameterdatei in das System. Jede Parameterdatei enthält die Steuersequenzen (für Fettdruck, Breitschrift, Umlaute oder ähnliches) für einen bestimmten Drucker. Durch eine solche Anpassung können Programme unabhängig von der Arbeitsweise des Druckers entwickelt und eingesetzt werden. Mit der Papierlänge, dem rechten und linken Rand lassen

sich weitere Einstellungen für die Druckausgabe vornehmen.

Für den Grafikdruck sind Angaben über shade, aspect, image und threshold wichtig. Für shade (Schattierung) haben Sie die Wahl aus: black and white (schwarz auf weiß), gray scale (Graustufen) und color (Farbe). Geben Sie »schwarz und weiß« an, so werden ab einer bestimmten Helligkeitsstufe der Punkte »weiße« statt schwarze Punkte ausgegeben. Diese Grenze läßt sich mit threshold einstellen. Eine andere Art der Farbumsetzung ist die Darstellung der Farben durch unterschiedliche Punktdichten (gray scale). Die Einstellung color ist nur für Farbdrucker. Mit aspect (Orientierung des Ausdrucks) kann der sonst horizontale Ausdruck um 90 Grad gedreht werden. Die Wahl des image bestimmt schließlich die Art des Ausdrucks: positiv oder negativ. Bei negativer Ausgabe erscheinen helle Flächen im Bild dunkel und dunkle Flächen entsprechend hell. Diese Einstellung läßt sich nur bei schwarz-weißer Schattierung aktivieren.

Verlassen Sie Preferences mit CANCEL, ignoriert das System eventuelle Änderungen der Parameter. Drücken Sie auf SAVE oder USE, werden alle Änderungen sofort gültig (außer Bildschirmauflösung) und bei SAVE zusätzlich noch in der Datei »system configuration« gespeichert.

printer.device

Die Schnittstellen-Routinen für die Druckausgabe. Diese Routinen übersetzen mit Hilfe einer im Verzeichnis »devs/printers« abgelegten Datei die Steuersequenzen des Amiga in druckerspezifische Steuersequenzen. Die jeweilige Parameterdatei kann mit dem Programm -> Preferences bestimmt werden. Die Liste der Amiga-Sequenzen befindet sich im -> RKM (Rom Kernel Manual) Libraries & Devices, Seite 420/421, Seite E41/E42 oder in den -> include files »printer.i« bzw. »printer.h«.

PRT:

Das DOS-Device PRT: (Drucker) des Amiga entspricht dem Gerät LPRT: bei MS-DOS und kann auch als solches von Amiga-Basic aus angesprochen werden. Der Unterschied zwischen PRT: und -> PAR: ist allerdings nicht nur bei Amiga-Basic wichtig: als (L)PRT: kann das Gerät druckerspezifische Steuersequenzen nicht verstehen, in Programmen muß man

die Amiga-Sequenzen verwenden. Will man aber die Druckfunktionen unbedingt mit den druckerspezifischen Sequenzen ansteuern, muß der Drucker über PAR: oder SER: angesprochen werden. Außerdem wandelt PRT: die Zeilenendmarkierung des Amiga (Code 10: Linefeed) in die Standardsequenz Wagenrücklauf (Code 13) und Zeilenvorschub (Code 10) um. Eventuell muß der Drucker entsprechend eingestellt werden, sonst erhält man doppelzeiligen Abstand bei Texten und Grafiken.

priority queue

Nach Priorität geordnete verkettete Liste des Systems (siehe ready queue, timesharing).

process

Ein im Ablauf befindlicher Codeblock. Ein process ist mehr als ein -> task. Er hat einen eigenen Speicher, nutzt »eigene« -> resources und funktioniert auf einer höheren Systemebene. Tasks werden von -> exec, processes von -> Amiga-DOS verwaltet.

Ein process besteht unter Umständen aus mehreren tasks und kann als »parent process« auch neue processes erzeugen und starten (daughter processes). Ein Beispiel dafür ist das CLI. Jedes mit NEWCLI oder RUN gestartete CLI ist ein »Tochterprozeß« des ersten CLIs. Da aber tasks die Grundlage eines process sind, richtet sich die Priorität des process nach derjenigen des tasks.

program library

Eine Sammlung von lauffähigen Routinen, die von einem bestimmten Programm als »run time library«, also während der Programmausführung, benötigt werden. Ein Beispiel hierfür sind die meist mit der Endung ».lib« versehenen Bibliotheken von Compilern oder Assemblern.

programming environment

-> operating system.

project

(Arbeitsprojekt) Das unter Verwendung eines Programms angestrebte »Endprodukt«, welches nach Abschluß des Programmablaufs als Datei vorliegt (ein Dokument, Bild, Musikstück oder ähnliches).

protect tab

(auch write-protect tab, read/write tab) Die Schreibschutzlasche einer Diskette. Während bei einer Diskette mit 3 1/2 Zoll Durchmesser der Schreib-

schutz bei geöffnetem Loch aktiviert ist, muß bei einer 5 1/4-Zoll-Diskette der Ausschnitt zu diesem Zweck abgeklebt werden.

public domain software

Echte Public-Domain-Software wird vom Programmautor der Allgemeinheit gebührenfrei, frei kopierbar und frei verwendbar (freeware) zur Verfügung gestellt. Es werden keine Copyright-Rechte geltend gemacht. Die meisten Autoren behalten sich allerdings vor, daß bei der Weitergabe oder Weiterentwicklung der Namensvermerk nicht gelöscht werden soll. Mittlerweile gibt es etwa 2500 Disketten mit Public-Domain-Software.

Eine andere Form der Public Domain ist die sogenannte »shareware«. Hier bittet der Autor um einen in der Regel eher symbolischen Betrag für die Nutzungsrechte an seinem Programm. Dafür stellt er gegebenenfalls dem »registrierten« Benutzer verbesserte Versionen und/oder Bedienungshilfen und Dokumentation zur Verfügung. Im Gegensatz zu »freeware« darf »shareware« eigentlich nur mit Genehmigung des Autors entgeltlich weitergegeben oder in geänderten Versionen/Programmen von anderen benutzt werden.

Für den Amiga gibt es neben den -> Fish Disks mehrere Sammlungen mit solchen Programmen. Zum Beispiel: FAUG, Amicus, AMuse, Phase4, HotMix, BCS, Software Digest (alle USA), Amiga Users' Group und ICPUG (UK). Weitere Software kann von amerikanischen bulletin boards per Modem abgerufen werden. Es empfiehlt sich allerdings, diese Angebote vorher zu überprüfen, weil viele mit dem Inhalt mancher Fish Disk übereinstimmen bzw. Routinen von Fish Disks in einer neuen, meist themenorientierten, Zusammenstellung anbieten. Da der Vertrieb solcher Software nicht als Gewinngeschäft gedacht ist, sollte der Preis nur die Kosten für Diskette und Zeitaufwand des Kopierens abdecken. Preise von mehr als 8 Mark grenzen schon an Gewinnsucht. Mehr als 10 Mark ist Wucher.

pulse code modulation

Ein Sammelbegriff für verschiedene Formen der Signalmodulation, die insbesondere bei der En- und Decodierung von Daten bei der Übertragung an externe Speichermedien verwendet werden. Die Grund-

lage des Verfahrens ist das Umkehren der magnetischen Polarität während eines »clock pulse« (siehe system timing frequencies), um den binären Wert des aktuellen Signalpulses zu codieren. Auf dem Amiga spielt das Verfahren beim -> trackdisk.device eine besondere Rolle, da es so umprogrammiert oder durch Zusatzroutinen erweitert werden kann, daß verschiedene Aufzeichnungsformate (MFM (Commodore), GCR (Apple), IBM) gelesen und geschrieben werden können.

quantization

Ursprüngliche Bezeichnung für das Umwandeln eines analogen Signals in digitale Form. Heute unterscheidet man »Digitalisieren« (Audioquantisierung) und »Pixelisieren« (Bildquantisierung). Übriggeblieben ist nur die Bezeichnung »quantization noise« für Störgeräusche im Signal, die sich besonders bei niedriger Lautstärke des Signals und niedriger Samplingfrequenz bemerkbar machen. Diese Störungen entstehen meist, weil die einzelnen »Stufen« des digitalisierten Signals eine zu grobe Annäherung an das Originalsignal darstellen. Die Stellen, an denen gemessen wird (»hold phases«), sind in der Regel zu weit auseinander, um den richtigen Verlauf der Signalschwingung voll zu erfassen (siehe Nyquist sampling, digitizer).

quantum

(Menge) Ein zugewiesener Anteil. In diesem Fall ein Anteil an der Benutzung der -> system resources. Jeder -> task erhält beim Amiga ein Quantum von $\frac{1}{60}$ Sekunden. Das ist die Zeit, die benötigt wird, um eine normale (NTSC) Bildschirmdarstellung viermal (im Interlace-Modus zweimal) zu erzeugen (siehe time-sharing, system timing frequencies).

queue

(Warteschlange) Eine besondere Form der -> linked list. Beim Amiga spielen drei queues eine Rolle: ready queue, waiting queue und message queue.

Ready queue: Eine nach Priorität sortierte Liste der Tasks, die auf ihr -> quantum warten. Die Tasks warten nicht auf Eingabe, sondern auf die Möglichkeit, vorliegende Daten zu verarbeiten.

Waiting queue: Diese Liste enthält diejenigen tasks, die auf eine Eingabe (ein Ereignis) warten.

Mit beiden Warteschlangen steuert das Multitasking-System den Zugriff der einzelnen tasks auf die -> resources des Amiga.

Message queue: Diese Warteschlange entsteht, wenn an einem -> message port die eingehenden messages nicht schnell genug verarbeitet werden (siehe system log jam).

RAM:

Das »virtuelle Laufwerk«, die RAM-Disk des Amiga-DOS (siehe logical device, physical device). Der diesem »Gerät« zugewiesene Speicherbereich wird vom System als ein Disketten-Laufwerk verwaltet. Da für den Zugriff auf die Daten der RAM-Disk keine mechanischen Operationen (Schreib-/Lesekopfbewegungen) erforderlich sind, können Ein-/Ausgabeoperationen schnell durchgeführt werden. Da das System nur den tatsächlich benötigten Speicherplatz zuweist, ist die RAM-Disk stets mit 100 Prozent belegt. Die auf RAM: abgelegten Daten gehen beim Ausschalten und - falls nicht durch besondere Maßnahmen »resetfest« angelegt - auch beim -> reset verloren. Eine eingerichtete RAM-Disk kann nur gelöscht werden, wenn, wie bei einem normalen Verzeichnis, alle darin enthaltenen Dateien vorher gelöscht wurden bzw. alle der RAM-Disk zugewiesenen Systemverzeichnisse den entsprechenden Verzeichnissen auf einer Diskette zugewiesen worden sind. Die ab Version 1.2 erzeugte -> .info-Datei für das Icon benötigt zwei Disketten-Sektoren (1 KByte).

RAM-Handler

Systemdatei mit den Routinen für die Einrichtung und Verwaltung der RAM-Disk.

raster

Eine grafische Darstellung in voller Größe. Auf dem Bildschirm wird als -> Window (Fenster) oder Screen (Bildschirm) unter Umständen nur ein Ausschnitt dieses Bildes dargestellt. Ein raster entspricht einer vollständigen -> Bitmap(gruppe) und darf eine Maximalgröße von 1024 x 1024 Pixel pro Bitplane haben.

RastPort/RPort

Eine besondere Form von -> message port (IDCMP), mit der die Darstellung innerhalb eines -> Screens oder -> Windows (RastPort wird bei Screens, RPort bei Windows benutzt) verwaltet wird. Eine entsprechende Datenstruktur wird von

-> Intuition automatisch für jeden Screen bzw. jedes Window initialisiert. Bestimmte Routinen des -> Kernels (graphics primitives) finden hier die benötigten Parameter (Beispiel: die Farben der drei »Zeichenstifte« (pens)). Bei den -> illustration data types beschränkt sich der beeinflussbare Darstellungsbereich auf den mit RastPort/RPort festgelegten Bereich. Das bedeutet, daß nicht versehentlich beim Zeichnen von Punkten, die sich außerhalb der mit RastPort/RPort festgelegten Koordinaten befinden, angrenzende Speicherstellen beeinflusst werden.

raster graphic display

Ein Darstellungsverfahren, mit dem - wie bei der Erzeugung eines Fernsehbildes - das Bild in parallel zueinander verlaufenden Zeilen punktweise aufgebaut wird. Dieses Verfahren spielt beim Amiga nicht nur beim Bildaufbau, sondern auch beim Speicherformat der -> IFF-Bilddateien eine wichtige Rolle. Anders als beim -> ACBM werden in einer IFF-Datei nicht die einzelnen -> Bitplanes, sondern die einzelnen Bildzeilen hintereinander gespeichert (interleaved bitmap). Dies ermöglicht ein speicherplatzsparendes, zeilenweises Einlesen einer IFF-Bilddatei (siehe vector graphic display).

RAW:

Das Gegenstück des DOS zu Intuitions input.device. Anders als bei -> CON: bzw. console.device liefert RAW: die Codes der gedrückten Tasten (»rohe« Eingabesequenzen) und nicht bereits verarbeitete Steuersequenzen. Eine verarbeitete Sequenz ist z.B. der Code 10 für <Ctrl J>. CON: und das console.device setzen Tastenkombinationen sofort in die entsprechende Aktion, hier Linefeed (Zeilenvorschub), um. Abgesehen von den dadurch fehlenden Editierfunktionen und der Ausnahme, daß das Window von RAW: nur durch ein -> reset geschlossen werden kann, sind die Arbeitsweisen von RAW: und CON: gleich.

ray tracing

Mathematisches Verfahren zur Erzeugung möglichst realistischer Computerbilder. Das Verfahren berücksichtigt die Lichtreflexion spiegelnder bis diffuser bzw. absorbierender Oberflächen. Um die Berechnung zu vereinfachen, berücksichtigt man nicht die von Lichtquellen ausgehenden Lichtstrahlen, sondern vom Auge

des Betrachters »ausgesandte Sehstrahlen«.

refresh

»Auffrischen« der Bildschirmdarstellung. Die Wiederherstellung eines Fensterbereiches, der zeitweise von einem anderen Fenster bedeckt war, und nun eventuell wieder mit dem vorherigen Inhalt gefüllt werden muß. Intuition stellt dem Programmierer drei Verfahren zur Auswahl:

1. simple refresh (einfache Reparatur): das Programm muß die Wiederherstellung des zerstörten Fensterinhalts selber vornehmen.
2. smart refresh (»schlaue« Reparatur): verdeckte Bereiche des Fensters werden in Sicherungspuffer kopiert und nach erneutem Sichtbarwerden zurückkopiert. Allerdings kann nur der betroffene Ausschnitt erneuert werden. Damit lassen sich die bei einer Vergrößerung des Fensters vorher unsichtbaren Teile nicht restaurieren.
3. SuperBitMap refresh: Für »SuperBitMap-Windows« müssen eigene Bitmaps definiert werden. Der Fensterinhalt wird in der Bitmap gespeichert und von dort an die entsprechende Stelle des -> Screens kopiert. Bei der Erneuerung selbst vergrößerter Fenster kann nun der Inhalt automatisch aus den Bitmaps geholt werden.

Als letztes Verfahren bietet Intuition eine »no care refresh«-Einstellung für die Fälle, in denen der Inhalt des Windows nach dem Aufdecken nicht von Belang ist. Die Unterschiede zwischen den ersten drei Verfahren können bei den Workbench Demos »lines«, »dots« und »boxes« beobachtet werden.

regions

(siehe layers) Die Routinen der layers.library betrachten Windows als eine Sammlung von Rechtecken (»clipping rectangles«). Beim Aufruf eines -> refresh werden die jeweiligen verdeckten Rechtecke zu einer »Schadensmeldungsliste« zusammengestellt und je nach der bestimmten refresh-Art wiederhergestellt. Nur die Rechtecke, die einen »Schaden« tatsächlich erlitten haben, werden aufgefrischt. Dieses Verfahren beschleunigt das Auffrischen.

reply port

Das -> message port des »Senders«, an den eine -> message zurückgeschickt werden muß (beim Programmieren unbedingt berücksichtigt).

requester

Meldung des Systems mit Aufforderung zur Dateneingabe (Pflichteingabe). Ähnlich wie ein -> Menü ermöglicht ein requester dem Benutzer eine Eingabe an das aktuelle Programm. Anders als ein Menü verlangt ein requester aber eine Eingabe und hält das Programm solange an, bis diese erfolgt ist. »Custom requesters« können wie -> custom gadgets überall im aktuellen Fenster erscheinen. »system requesters« dagegen erscheinen immer in der linken oberen Ecke des »system screen«, also des Screens, von dem aus das aktuelle Programm gestartet wurde. Besitzt ein Programm einen eigenen Screen oder ein Fenster in voller Bildschirmgröße (wie das Ausgabefenster von Amiga-Basic), kann es vorkommen, daß ein system requester dem Benutzer nicht sichtbar ist und dadurch der Eindruck eines -> lockup entsteht. In solchen Situationen ist es immer sinnvoll, zuerst zu versuchen, den Workbench-Screen mit dem -> keyboard shortcut nach vorne zu holen.

rescue dump

Sicherungskopie des Arbeitsspeichers (insbesondere des -> Stack). Eine solche wird z.B. erstellt bei einem -> exception, trap oder nach Ablauf des -> quantum, damit das Task nach der Fehlerverarbeitung bzw. zu Beginn des nächsten quantum an derselben Stelle weitermachen kann. Meistens versteht man unter einem »rescue dump« die Sicherung des Stackinhalts bei einem Programmabsturz (siehe debug()).

reset

(auch reboot, warm boot) Die Neuinitialisierung des Systems entweder »freiwillig« durch Betätigung der Warmstart-Tastenkombination <Ctrl Amiga Amiga> oder notgedrungen nach einer -> guru meditation (siehe bootstrap). Bei diesem »Warmstart« geht der Inhalt des Speichers verloren. Das gilt ohne besondere Vorkehrungen (resetfeste RAM-Disk) auch für eingerichtete RAM-Disks. Nach einem DISKCOPY ist ein reset zu empfehlen, wenn man mit der Kopie gleich weiterarbeiten will. Sonst kann ein disk swap -> lockup den Computer lahmlegen.

resource

(wörtlich Rohstoff, Quelle, Reichtum) Ein Bestandteil des Systems, der in irgendeiner

Weise benötigt wird, um Daten zu verarbeiten. Hierzu gehören in erster Linie die CPU und Co-Prozessoren, CIAs, Speicher (auch Erweiterungen), Laufwerk, Schnittstellen, Libraries und Devices. Ein »shared resource« ist ein Teil des Systems, auf den mehrere (nahezu) gleichzeitig laufende tasks zugreifen können.

return code

Der von einer Systemfunktion übergebene Wert einer Operation. Beim CLI-Befehl WHY bedeutet die Meldung »The last command did not set a return code« lediglich, daß keine Fehlernummer von der Routine zurückgemeldet wurde (siehe stdin, stout, stderr).

restart

Die Neuinitialisierung eines Laufwerks nach einem Diskettenwechsel. Dies geschieht entweder automatisch (siehe Disk-Validator) oder mit dem CLI-Befehl DISKCHANGE.

RevPath

(reverse path: die umgekehrte Schreibweise) Ein Kennzeichen in der Zeichensatzdefinitionsdatei (.font), das den Zeichensatz als »rückläufig«, also von rechts nach links geschrieben, identifiziert und das System zur entsprechenden Darstellung veranlaßt.

RKG

(ROM Kernel Graphics) -> graphics primitives.

RKM

(ROM Kernel Manuals) Strenggenommen handelt es sich hier um den Doppelband »Libraries & Devices/Exec« aus der »Amiga Technical Reference Series« von Addison-Wesley. Mittlerweile aber werden in Amerika alle Bände der Serie mit dieser Abkürzung bezeichnet.

ROMWack

-> Wack.

runtime library

(auch in-memory library) Eine -> library, die beim Ablauf eines Programms benötigt wird. Bei vielen Amiga-libraries handelt es sich um »in-memory«-libraries, weil diese im ROM -> Kernel enthalten sind. Bei anderen System werden die erforderlichen Libraryroutinen in den Programmcodes aufgenommen. Beim Amiga werden die residenten Routinen aufgerufen bzw. die nicht-residenten libraries zunächst geladen und dann aufgerufen.

S:

Das »sequence«-Verzeichnis des Systems. Hier werden Kommandodateien (-> batch files) vom CLI-Befehl EXECUTE gesucht, falls sich diese nicht im aktuellen Verzeichnis (-> root directory) befinden (siehe logical device).

sampling

(Entnahme einer Stichprobe) Das punktuelle Messen eines analogen Signals, um einen entsprechenden digitalen Wert zu erhalten (siehe quantization).

sampling period

Anzahl der »clock cycles«, die vom -> audio.device benötigt werden, um eine Stichprobe zu »spielen«.

sampling rate

Die Anzahl der Stichproben pro Sekunde (siehe system timing frequencies, sampling).

scatter load

Das »gestreute« Laden von Programm- oder Datensegmenten. Es kommt beim Amiga selten vor, daß ein umfangreiches Programm zusammenhängend im Speicher steht. In der Regel werden die einzelnen Segmente dort abgelegt, wo gerade ein Bereich in der passenden Größe frei ist. Durch eine Art -> linked list koppelt das System die einzelnen Segmente. Der Ladevorgang beim Amiga ist daher etwas kompliziert. Zuerst wird vom DOS-Loader ein -> hunk_header geladen, der Informationen über Anzahl und Umfang der noch zu ladenen hunks und die Namen der zu öffnenden libraries enthält. Die Anfangsadresse des Bereichs wird in einer Tabelle abgelegt und nach dem Laden der erforderlichen Segmente werden alle Adressen und -> offsets, die ein Segment betreffen oder die in dem Segment enthalten sind, den Basisadressen der einzelnen Speicherbereiche angepaßt. Bei overlays wird der Anpassungsvorgang jeweils wiederholt.

screen

(wörtlich: Bildschirm) Die Darstellungsfläche sowie der Arbeitsspeicher für die Abbildung von -> windows entweder als standardisierter Workbench-Screen oder als vom Programmierer definierter Custom-Screen. Unter Intuition ist die Anzahl der screens nur durch den verfügbaren Speicher (Chip RAM!) begrenzt. Amiga-Basic erlaubt nur vier verschiedene screens. Verschiedene screens dürfen sich teilweise

oder ganz überlagern und sind daher auch mit einem -> depth gadget und/oder -> drag gadget versehen. Anders als windows müssen screens immer die volle Bildschirmbreite erfassen.

Screens brauchen nicht die volle Höhe des Bildschirms einzunehmen, müssen aber dann immer mit der untersten Zeile des darstellbaren Bildschirmbereichs abschließen, das heißt sie dürfen weder nach oben noch seitlich aus dem Bild gezogen oder gescrollt werden. Bei Amiga-Basic kann es Probleme mit solchen screens geben.

Die Datenstruktur »Screen« von Intuition enthält bestimmte Parameter wie Zeichensatz, Farbauswahl und Auflösung, die von allen auf dem screen dargestellten Fenstern übernommen werden. Es ist daher nicht möglich, ein Fenster mit einer anderen Auflösung als der dem screen zugrundeliegenden darzustellen. Jeder screen ist im Prinzip ein virtuelles Sichtgerät und kann nicht mit anderen screens in Verbindung treten. Daher können die Fenster eines screens nicht in einen anderen screen verlagert werden.

scrolling

(abgeleitet von »screen rolling«) Die vertikale und/oder horizontale Verschiebung der Bildschirmdarstellung (siehe gadget scroll bar/drag bar).

SCSI

(Small Computer System Interface, auch »scuzzy«, sprich »scasy«, genannt) Schnittstelle für den Anschluß von Festplatten (Hard-Disks). Eine solche befindet sich auf der Festplattenkarte des A2000 sowie auf den Karten verschiedener Drittanbieter. Die entsprechende Norm (ANSI X3T9.2) erlaubt den Anschluß von bis zu sieben SCSI-kompatiblen Peripheriegeräten mit einer Übertragungsgeschwindigkeit von etwa 1,25 MByte/s.

sector

Abschnitt eines -> track (Spur) auf einer Diskette. Das -> trackdisk.device weist bei einer »DOS-Diskette« einem track elf Sektoren mit einer Länge von 128 Langwörtern (512 Byte) zu. Zusätzlich bekommt jeder Sektor eine 16 Byte lange »sector label area«, die aber bei normalen Schreib-/Lesevorgängen nicht benutzt wird. Da die Anzahl der Sektoren pro Spur festgelegt ist, spricht man von

»hard sectoring«. Der Gegensatz »soft sectoring« bedeutet, daß die Spuren eine unterschiedliche Anzahl von Sektoren haben. Bei Disketten im IBM-Format sind acht oder neun Sektoren pro Spur üblich.

segment

Ein aus einer unterschiedlichen Anzahl von -> hunks bestehender Abschnitt eines ablauffähigen Programms.

select box

Der »Aktivierungsbereich« eines -> gadget. An dieser Stelle muß sich die aktive Stelle des Mauszeigers bei der Selektion des gadget befinden (siehe hot spot).

semaphore

Steuersignal -> signal.

sequenced drawing

Ein von Zeichentrickfilmen her bekanntes Verfahren, mit dem eine Animation in Einzelbilder »zerlegt« wird. Durch eine Aneinanderreihung der Einzelbilder (mit Amiga-Basic etwa durch den Grafikbefehl PUT) ergibt sich dann die bewegte Bildfolge oder -> Animation (siehe motion control).

SEIR:

Das Amiga-DOS-Gerät »serielle Schnittstelle«. Normalerweise ist dieses Gerät für -> debug()-Zwecke als Ausgabe für den Anschluß eines externen Terminals ausgelegt. Es kann aber auch als Druckerport oder Modemport umdefiniert werden. Mit dem AUXCLI-Befehl des TOOLKITS von MetaComCo kann ein externes Terminal für einen Zweitbenutzer angeschlossen werden. Eingaben erscheinen aber beim Amiga im AUXCLI-Window und nicht auf dem Terminal.

serial.device

Die Systemroutinen für die Verwaltung der seriellen Schnittstelle. Theoretisch ist eine Übertragungsgeschwindigkeit von 112 bis 292000 Baud möglich. In der Praxis aber dürfte 19200 Baud die obere Grenze darstellen.

server

(to serve: dienen) Eine untergeordnete Routine, die als eine Art -> spooler zwischen einem Programm und einer beanspruchten -> resource (z.B. Drucker oder Laufwerk) Daten überträgt (siehe handler).

SetMap

Workbench-Tool in der SYSTEM-Schublade für die Änderung der Tastaturbelegung.

Anwendung: SetMap-Icon einmal anklicken und INFO im Menü wählen, die Zeile »Tooltypes« ändern: »SETMAP = Name der Datei mit der gewünschten Tastaturbelegung«, Einstellung mit der SAVE-Option speichern, das INFO-Window schließen und das SetMap-Icon erneut anklicken. Danach sollte die gewünschte Belegung eingestellt sein. Bei der Benutzung in der »startup-sequence« genügt die Anweisung »SETMAP Dateiname«.

shell

(Schale, Hülle) Eine zusätzliche Benutzeroberfläche, die das Arbeiten mit dem Computer vereinfacht. Eine Shell für den Amiga enthält z.B. in der Regel eine Funktion, mit der bei der Arbeit im CLI die letzten Befehlszeilen noch einmal abgerufen oder aber die aktuelle Befehlszeile editiert (nachträglich Zeichen einfügen/löschen) werden kann. Die neue Amiga-Shell der Workbench Version 1.3 ist ein Beispiel für eine solche komfortable Eingabevorrichtung.

Sidecar

Der »Beiwagen« des Amiga, der den Computer zu einem PC-Kompatiblen (IBM-XT) macht. Der Anschluß des Sidecar erfolgt über den seitlichen -> expansion bus und eine -> Janus-artige Schnittstelle. Wie die PC-Karten des A2000 belegt Sidecar den Bereich \$800000 bis 9FFFFF. Intern verfügt Sidecar über drei Erweiterungsslots, wovon einer für eine Festplatten-Karte reserviert ist. Ab Betriebssystemversion 1.2 kann eine Partition der Festplatte auch vom Amiga genutzt werden. Der Amiga hat keinen Zugriff auf den internen Speicher des Sidecar.

sidemount device

Ein Zusatzgerät, das wie Sidecar am -> expansion bus angeschlossen wird.

signal

(auch semaphore) Ein Subsystem von -> exec, mit dessen Hilfe die Kommunikation zwischen -> processes über -> messages synchronisiert werden. Insgesamt stehen 32 Signale zur Verfügung, wovon exec selbst 16 benutzt. Die Bedeutung eines Signals hängt von der »vereinbarten« Definition ab. So kann das Signal Nummer 20 für einen -> process das Schließen eines Fensters signalisieren, während ein anderer process das Signal Nummer 23 verwendet.

Snapshot

Eine Menüfunktion der Workbench, um die Positionierung von Icons in einem Window zu fixieren. Manche Programme heben die Fixierung bei erneuter Bearbeitung der entsprechenden Datei wieder auf. Die Position wird dann wieder von Intuition bzw. den Routinen der icon.library festgelegt.

software failure

Diese Fehlermeldung bedeutet, daß ein Fehler beim -> memory management eingetreten ist. Das verursachende Programm hat wahrscheinlich Daten in einem nicht zugewiesenen Speicherbereich abgelegt. Ein -> reset läßt sich in der Regel nicht mehr vermeiden.

SoftVoice

-> speech synthesis.

sound synthesis

(Klangsynthese) Töne lassen sich mit zwei unterschiedlichen Verfahren synthetisieren: die additive oder subtraktive Synthese. Beim zweiten Verfahren werden unerwünschte Bereiche des Klangspektrums herausgefiltert (SID-Chip des C64). Der Amiga dagegen benutzt, wie auch eine Hammond-Orgel, das additive Verfahren. Töne werden als Kombination (Addition) des Grundtons und einer unterschiedlichen Anzahl (bis maximal 31) der sogenannten Obertöne aufgebaut. Jeder Oberton ist um einen ganzzahligen Faktor größer als der vorangegangene Oberton. Das bedeutet, daß der erste Oberton die doppelte Frequenz des Grundtons hat, der zweite Oberton die dreifache Frequenz und so weiter. Die Kombination der Töne bestimmt die allgemeine Form der erzeugten Welle. Für praktische Zwecke genügen meistens die ersten sieben oder acht Obertöne: Sägezahn: Grundton mit allen Obertönen

Rechteck (Puls): Grundton mit den ungeraden Obertönen
Dreieck: Grundton mit den geraden Obertönen

Beliebige Kombinationen können jeden denkbaren Klang erzeugen. Die Formel für die Berechnung einer Wellenform unter Amiga-Basic befindet sich im Programm Musicdemo im Abschnitt »InitSound« direkt vor den Datazeilen. Die dort angegebene Version erzeugt eine Sägezahnschwingung. Bei der Verknüpfung verschiedener Wellenformen ist darauf zu achten, daß der Übergang nach Möglichkeit beim Wert 0

erfolgt, um störende Klickgeräusche zu vermeiden. Solche Störgeräusche kommen unter Umständen bei der Ausführung von SOUND RESUME vor, wenn vorher eine gerade laufende Wellenform mit SOUND WAIT unterbunden wurde.

Durch die Höhe (Amplitude) der Schwingung wird die Lautstärke des Tons bestimmt. Die Verteilung der Schwingung über einen gegebenen Zeitraum - in der Regel eine Sekunde - ergibt die Frequenz in Hertz.

Besondere Effekte, wie Glissandi oder Portamenti, können mit -> frequency modulation erzeugt werden. Die »Hüllkurve« (oder ADSR-Kurve, nach den Bestandteilen »Attack, Decay, Sustain, Release«) wird mit -> amplitude modulation gesteuert.

Der Amiga kann mit den Noten der MIDI-Nummern 17 (E0) bis 130 (B9-) einem Frequenzbereich von 20 bis 15000 Hertz erzeugen (siehe Notentabelle). Ein Demo in C sowie ein -> Tutorial zum audio.device befinden sich auf Public Domain. Das RKM Libraries & Devices enthält auf Seite 225 eine Liste der vorgeschlagenen Prioritäten (precedences) für die Zuweisung von Audiokanälen.

source code

(Quellcode): Ein in einer (in der Regel zu compilierenden) höheren Programmiersprache oder Assemblersprache geschriebenes Programm. Anders als bei »interpretierbaren« höheren Sprachen wie Basic kann source code nicht direkt vom Computer nach Einlesen des Programms verarbeitet werden, sondern muß zuerst in vom Computer lesbare Maschinensprache-Anweisungen übersetzt und gegebenenfalls mit system libraries »gelinkt« werden, bevor lauffähiger Code entsteht (siehe object code, link editor).

spacing

(Zeilendichte) Mit diesem Schalter der -> Preferences wird die Anzahl der druckbaren Zeilen pro Papierseite bestimmt. Die absolute Länge der Seite wird mit dem Schalter »paper size« bestimmt. Die Blattgröße der verschiedenen Einstellungen (in Zoll) ist im Anwender-Handbuch enthalten. In Zeilen umgerechnet ergeben diese Größen 66 bzw. 85 (U.S. legal) Zeilen pro Seite (absolut). In beiden Fällen wird diese Anzahl mit der Standardeinstellung von 6 lpi (lines per inch: Zeilen pro Zoll) erzielt.

Die »internationale« Seitenlänge für das in Druckern verwendete Papier beträgt 11¼ Zoll (66 Zeilen). In Deutschland wird aber die Europa-Norm mit einer Seitenlänge von 12 Zoll (72 Zeilen) verwendet. Bei anderen Blattlängen ergibt sich die Anzahl der Zeilen aus der Formel Blattlänge (in Zoll) x 6.

Eine weitere Einstellung, 8 lpi, bringt entsprechend mehr Zeilen pro Blatt (x 8, statt x 6), kann aber unter Umständen dazu führen, daß außer im »pitch«-Modus »fine« die Zeichen untereinanderliegender Zeilen ineinander überlaufen.

Bei DTP-Programmen (DTP = Desktop Publishing) wird die Anzahl der Zeilen vorrangig von der Höhe des gewählten Zeichensatzes bestimmt (siehe pitch). Die Norm hierfür heißt »decipoint« und wird mit 720 Punkte pro Seite definiert. Das entspricht 10 Punkte pro Zeile bei 12 Zoll Papier. So läßt sich die Zeilendichte der zusätzlichen Zeichensätze mit der Formel 720/Höhe des Zeichensatzes bestimmen.

special characters

(Sonderzeichen). Den nicht alphanumerischen Zeichen der Tastatur werden oft Sonderfunktionen zugewiesen. Hier eine kurze Übersicht der Sonderzeichen von CLI und Amiga-Basic:

- CLI:

(wild card: Joker): eine beliebige Anzahl des folgenden Zeichens für das -> pattern matching.

? (wild card): das folgende Zeichen bei -> pattern matching oder die Help-Funktion des CLI (muß als solche mit Leerzeichen vom Befehl getrennt werden)

: das Hauptverzeichnis einer Diskette; Endzeichen beim Namen eines logischen Geräts oder einer Diskette

; das Kommentarzeichen. Nachfolgender Text wird nicht berücksichtigt

. Leitet den Punktbehl eines -> batch file ein. Von einem Leerzeichen oder Return gefolgt, kennzeichnet es eine Kommentarzeile eines batch files.

/ Trennzeichen bei Suchpfadangaben bzw. Parameterangaben beim Öffnen eines CON: Windows (muß nach jedem Parameter stehen).

/A Argument erforderlich

/K Keyword (Schlüsselwort) beim Befehl erforderlich

/S Option, muß zuerst »eingeschaltet« werden; keyword erforderlich.

> < Umleitungszeichen bei -> i/o redirection

<Parameter> markiert die Stelle, an der ein eingegebener Parameter in ein batch file eingesetzt werden soll.

+ unterbindet sofortige Befehlsausführung bei der Eingabe von -> batch files im Direktmodus; muß als letztes Zeichen der Zeile vor Betätigen der RETURN-Taste außer bei der letzten Zeile eingegeben werden.

\$ Austauschzeichen für Variablennamen in batch files

" " Abgrenzungszeichen für Zeichenketten, die Leerstellen oder Sonderzeichen (: /) enthalten; mit einem Leerzeichen dazwischen dasselbe wie : (root directory).

* das Ausgabegerät stdout; der aktuelle Bildschirm.

- Amiga-Basic:

! numerische Variable mit einfacher Genauigkeit

numerische Variable mit doppelter Genauigkeit oder Kanalnummer bei PRINT-Anweisungen für Dateien oder Geräte

\$ Zeichenkettenvariable

% kurze Integervariable

& lange Integervariable

&h hexadezimale Zahl

&o Oktalzah

: Trennzeichen für Befehle

; unterbindet den Zeilenvorschub am Ende einer Druckanweisung

' Kommentarzeile (= REM)

" " Abgrenzungszeichen für Zeichenketten

speech synthesis

(Sprachsynthese) Das Sprachsynthese-Subsystem des Amiga heißt offiziell SoftVoice und besteht aus drei Teilsystemen: -> translator.library, narrator.device und audio.device. Die Grundlage des Subsystems bilden die Sprachelemente der englischen Sprache. Eine eingegebene Textkette wird zunächst von den Routinen der translator.library in Buchstaben(kombinationen) zerlegt und in die entsprechenden Phonemcodes umgewandelt. Man kann diese Umwandlung umgehen, in dem man eigene Phonemketten bis zu einer maximalen Länge von etwa 16000 Codes angibt. Dadurch läßt sich eine genauere Kontrolle der zu erzeugenden sprachlichen Äußerung erzielen, denn die translator.library hat Schwierigkeiten mit seltenen englischen und fremdsprachlichen Lautfolgen. Die Phonemkette wird an das narrator.device weitergeleitet und dort nach mathematischen Regeln (Algorithmen) in die entsprechenden Wellenformdaten umgewandelt.

Die Grundlage dieser Umrechnung ist die Ermittlung der zu jedem Phonem gehörigen ersten drei »Formanten« oder Hauptfrequenzen. Aus dieser Umrechnung, die in etwa der für die Berechnung des Durchmessers eines mechanischen Tiefpaßfilters entspricht, gehen auch die Werte hervor, die das narrator.device für die passende Höhe und Breite des Mundes bei jedem Vokallaut »auf Anfrage« liefert (wie bei Speech-Toy). Die Wellenformdaten werden schließlich an das audio.device weitergegeben und in hörbare Töne umgewandelt.

Die Betonung/Intonation einer Aussage wird mit bestimmten Sonderzeichen bzw. an Vokalphonemcodes angehängten Betonungszahlen gesteuert. Pausen lassen sich durch Leerstellen in der Codekette oder den Code QX erzeugen. Neben den dokumentierten Codes gibt es noch die »versteckten« GH, GX, KX, UX, /B, /M und /R, wovon allerdings nur UX, für den Laut »oo« in Wörtern wie »boot« oder »hoot« ohne weiteres brauchbar ist. Der Code KX ist für die Kombination »ck« und GH und GX sind als Schlußlaut nützlich. Überraschenderweise erzeugt /B am besten den Laut »ch« des deutschen Wortes »ich«.

spooler

(Abkürzung für simultaneous peripheral operations on line) Datenausgabe auf langsame Peripheriegeräte nach vorheriger Speicherung auf schnelle Datenträger. Mit dieser Technik glied man ursprünglich in größeren Rechenzentren Engpässe aus, die bei der Datenausgabe mehrerer gleichzeitig laufender Programme auf nur ein oder zwei Drucker entstanden. Manche Textverarbeitungsprogramme auf dem Amiga nutzen Spooling, um nach dem Druckbefehl auch bei längeren Texten sofort weiterarbeiten zu können.

sprite ->

GELS, animation.

stack

(Stapelspeicher) Ein besonderer Speicherbereich des Systems (supervisor stack) wie auch eines jeden process, in dem Registerwerte, Rücksprungadressen und aktuelle Daten vorübergehend abgelegt werden. Nach Ablauf des Quantums eines process (task) werden sämtliche in Verarbeitung befindlichen Werte (der aktuelle »Kontext« des process) auf den Stapel gesichert. Der-

selbe Vorgang wird bei einem Interrupt oder -> alert ausgelöst. Anders als eine -> list oder queue sind die Werte »auf dem Stapel« nach dem LIFO-Verfahren (last in, first out) abgelegt. Das Prinzip, die zuletzt in diesen Bereich abgelegten Daten zuerst wieder abzuholen, wird am Beispiel der Unterprogrammssprünge deutlich. Bei jedem dieser Sprünge muß die Rückkehradresse auf den Stapel gesichert werden. So entsteht nach mehrmaligen Unterprogrammaufrufen von Unterprogrammen aus ein »Stapel« Rückkehradressen. Um vom letzten Unterprogramm in das vorletzte (und so weiter) zurückzugelangen, muß die zuletzt gespeicherte Rückkehradresse vom Stack geholt werden.

stack frame

Ein Teil des Stapelspeichers, der Programmblöcken oder Funktionen für die Ablage von lokalen Variablen zur Verfügung gestellt wurde. Die zweite Hälfte einer -> guru meditation-Meldung zeigt in der Regel den Anfang des »stack frame« zu einem fehlerhaften Block. Der letzte Eintrag der Meldung ist normalerweise die Speicheradresse des Programmblocks.

startup code

-> boot(strap).

startup-sequence

(die Bootdatei) Eine besondere Systemdatei, die sich im Verzeichnis »s« befindet und eine Reihe von Befehlen enthält (batch file), mit denen das Betriebssystem des Amiga initialisiert wird. Amiga-DOS sucht diese Datei nach Einlegen der Bootdiskette (Systemdiskette) als erstes und führt die darin enthaltenen Befehle aus. Findet das DOS die Datei nicht, kann die Systemsteuerung nicht an die Workbench oder ein anderes Programm übergeben werden, sondern bleibt bei Amiga-DOS, das dann nur das CLI-Window initialisiert. Ein solches CLI darf nie mit ENDCLI ausgeschaltet werden, solange das System nicht mit einer anderen Form der Eingabesteuerung (ein zweites CLI oder LOADWB) »ansprechbar« ist. Sonst erfolgt ein -> lockup. Die »startup-sequence« ist auch bei auto.config-Erweiterungen wichtig, weil diese erst mit den entsprechenden Befehlen dieser Datei, BINDDRIVERS bzw. MOUNT, in das Betriebssystem eingebunden werden.

stdin, stdout, stderr

Die vom System voreingestell-

ten Standardgeräte für Ein-/Ausgabe und Fehlermeldungen (Tastatur, Bildschirm, Bildschirm) können vom CLI mit $\rightarrow$ i/o redirection für die Ausführung eines Befehls »umgeleitet« werden.

stream

(Datenstrom) Ein Datenübertragungskanal zwischen dem Computer und einem Peripheriegerät. Der Kanal wird durch ein $\rightarrow$ file pointer (filing system) identifiziert, wenn die Daten nicht gepuffert werden; sonst durch ein $\rightarrow$ file descriptor oder, unter Amiga-DOS, von einem $\rightarrow$ file handle.

style

(Schriftart) Die Zeichensätze des Amiga können in vier verschiedenen Schriftarten dargestellt werden: plain (der normale Zeichensatz), bold (Fettdruck am Bildschirm, Breitdruck bei den meisten Druckern), italic (Schrägedruck) und underlined (unterstrichen). Die nicht normalen Zeichensätze werden unter Verwendung von Systemalgorithmen abgeleitet. Die Ergebnisse dieser Berechnungen sind nicht immer optimal. Zeichensätze mit fest definierten Schriftarten werden von den Systemalgorithmen nicht modifiziert, wenn die entsprechenden Schalter (flags) in der Zeichendefinitionsdatei (.font) gesetzt sind.

ST506

Festplatten-Controller für maximal zwei Geräte mit mehreren Platten (bis zu 8 pro Festplatte). Der Controller befindet sich neben dem $\rightarrow$ SCSI auf der Festplatten-Karte zum Amiga 2000. Die Übertragungsgeschwindigkeit beträgt etwa 700 KByte/s.

sub-menu

$\rightarrow$ menu.

SuperBitMap

Ein $\rightarrow$ Fenster (window) mit eigener Bitmap (siehe refresh). Diese Technik vereinfacht das Auffrischen des Fensterinhalts.

synchronous

(synchron, gleichzeitig) Strenggenommen bedeutet diese Bezeichnung, daß zwei oder mehrere Aktionen gleichzeitig stattfinden. Während die Autos des $\rightarrow$ RKM Hardware dieser Definition folgen, werden im RKM Exec nacheinander erfolgende Aktionen damit gekennzeichnet.

SYS:

Dieses Verzeichnis ist gleichbedeutend mit der Startdiskette. In der Regel ist das die Workbench. Wann immer Dateien

von der Workbench benötigt werden, sollte anstatt dem Namen des Laufwerks, in dem sich die Workbench (eventuell) befindet, SYS: angegeben werden (siehe logical device.)

SYSTEM:

Dieses Verzeichnis enthält von Amiga-DOS oft benötigte $\rightarrow$ tools. Darunter befindet sich das $\rightarrow$ CLI sowie die Befehle Format, Diskcopy, Screen-Dump, SetMap und andere.

system-configuration

Diese Datei enthält die mit $\rightarrow$ Preferences eingestellten Systemparameter.

system disk

Eine Diskette im Amiga-DOS-Format, die unabhängig von der $\rightarrow$ Workbench bootfähig ist (siehe boot block). Auch als Bezeichnung für Diskette benutzt, mit der das System initialisiert wird (siehe SYS:).

system log jam

Ein »Verkehrsstau« an einem $\rightarrow$ message port, der in der Regel dadurch verursacht wird, daß eingegangene messages nicht schnell genug bearbeitet bzw. zurückgeschickt werden. Bei einigen message-Typen wird in solchen Fällen dem message port zusätzlicher Speicher zugewiesen. Eine Freigabe des Speichers erfolgt jedoch nicht bei der Abarbeitung der message, sondern erst mit Schließen des dazugehörigen Fensters. Das kann unter Umständen zu Speicherengpässen (out of memory) führen. Eine solche Situation kann eventuell mit Schließen der betroffenen Fenster gelöst werden.

system power supply

Die Stromversorgung für am $\rightarrow$ expansion bus angeschlossenen Peripheriegeräte (z.B. Speichererweiterungen). Diese ist beim A 1000 ausreichend für etwa 2 bis 4 MByte zusätzliches RAM. Eine Überlastung der Stromversorgung kann die CPU zerstören.

system time

Grundlage für die Berechnung des Datums durch Amiga-DOS. Das Betriebssystem berechnet alle Datumsangaben relativ zum 1.1.1978. Falls der Anwender das Datum nicht ausdrücklich neu setzt (z.B. mit SETDATE), wird es der zuletzt auf der Bootdiskette angelegten Datei entnommen und auf »eine Sekunde« später eingestellt. Das Systemkalendarium ist auf insgesamt 100 Jahre (bis 31.12.2077) ausgelegt.

Die aktuelle Systemzeit (current time), die Zeit in Sekunden und Mikrosekunden seit dem letzten Einschalten (power up) des Systems, wird vom $\rightarrow$ input.device benutzt, um das Auftreten eines $\rightarrow$ events zu bestimmen.

system timing frequencies

Taktfrequenzen des Systems. Ausgangspunkt für den Systemtakt ist die amerikanische Bildschirmträgerfrequenz von 3,579545 MHz. Diese entspricht bei 60 »Bildern« in der Sekunde einer Zeilenaufbaudauer von etwa 63 Mikrosekunden.

Während dieser Zeit stehen 227,5 $\rightarrow$ memory access cycles zur Verfügung, was eine Dauer von etwa 280 Nanosekunden pro cycle ergibt. Dieser Wert wird als »system clock cycle«, kurz »clock«, bezeichnet (siehe time slot allocation). Die 68000-CPU wird mit einer Frequenz von 7,15909 MHz – also doppelt so schnell wie die Bildträgerfrequenz – getaktet. Die Sonderchips arbeiten mit 14,31818 MHz wiederum doppelt so schnell wie die CPU.

system timing interval

(auch system clock cycle) Der Systemuhrtakt von 280 Nanosekunden (siehe system timing frequencies). Dieser wird mit einem Periodenwert von 1 gleichgesetzt. Um den Periodenwert für die Tonerzeugung zu bestimmen, wird die Dauer einer einzelnen Stichprobe (sample) durch den Systemuhrtakt geteilt. Die höchstmögliche Stichprobenfrequenz liegt bei 28867, die kürzeste Stichprobendauer ist demnach 1/28867 Sekunde, was einem Periodenwert von 124 entspricht (siehe period). Bei niedrigeren Periodenwerten reicht die Zeit nicht mehr aus, um neue Daten einzulesen.

T:

Das »temporary«-Verzeichnis (temporär: zeitweilig, zum Übergang). Hier legen unter anderem die CLI-Editorroutinen ED und EDIT Temporärdateien oder Sicherungskopien der zuletzt bearbeiteten Versionen von Texten ab. Ab Version 1.2 erscheint dieses Verzeichnis nicht notwendigerweise auf einer Diskette, weil es, anders als bei der Version 1.1, nur bei Bedarf vom jeweiligen Editor eingerichtet wird.

table look-up/TLU

(Nachschlagetabelle) Ein Verfahren, um Datenwerte schnell umzuwandeln. Die »Zieldaten« werden tabellarisch in einer Da-

tei abgelegt und können entweder über einen Feldindex oder durch Zeichenkettenvergleich gegen die Eingangsdaten ausgetauscht und damit umgewandelt werden.

task

Ein lauffähiges Codemodul, bestehend aus Codeblock, $\rightarrow$ Stack und $\rightarrow$ task control block, welches nach Bedarf geladen oder gelöscht wird. Tasks sind die Grundlage eines $\rightarrow$ process. Ein task wird erst nach der Zuweisung eines $\rightarrow$ IDCMP und/oder message ports zu einem process.

task control block

(TCB) Eine Exec-Datenstruktur, in der sich Informationen zu einem task befinden. Diese wird bei der Einordnung des task in einem $\rightarrow$ queue (Warteschlange) angelegt. Das $\rightarrow$ include file »exec/task.h« bzw. »exec/task.i« enthält die Definition dieser Datenstruktur.

task state

Der aktuelle Status einer task. Man unterscheidet sieben Stufen: added (neu gestartet, noch nicht eingestuft), waiting (im »Schlafzustand«, wartet auf einen $\rightarrow$ event), ready (wartet auf Zugang zu den $\rightarrow$ resources), running (hat gerade Zugang zu den resources), exception (Fehler aufgetreten), removed (abgelaufen, befindet sich beim $\rightarrow$ cleanup) und invalid (wird ignoriert).

terminal

(wörtlich Endstation) Datenein-/ausgabegerät. Meist besteht dieses aus Tastatur und Bildschirm (Datensichtgerät/VDU).

terminator

(end-of-name) Namenabgrenzungszeichen. Beim Amiga wird normalerweise das Leerzeichen als terminator interpretiert. Aus diesem Grund müssen Dateinamen, die Leerstellen enthalten, mit doppelten Hochkommata (") eingeschlossen werden. Das geschiftete Leerzeichen (non-break space) dagegen wird nicht als terminator angesehen, es sind bei dessen Verwendung keine Hochkommata erforderlich.

tick

Die Zeitmeßeinheit des Systems. Sie beträgt $\frac{1}{50}$ Sekunde (siehe IntuiTick).

time division multiplexing

(TDM) Das »Aufteilen« eines Übertragungskanals oder eines $\rightarrow$ resource auf mehrere Routinen, die jeweils eine be-

stimmte Benutzungsfrist (-> quantum) zugewiesen bekommen. Die Verfahren von -> exec basieren auf einem ähnlichen Prinzip (-> round robin). Die Begriffe time-sharing/time-slicing sind Synonyme für TDM.

timer.device

Subsystem, das für die Zeitmessung benutzt wird. Es stehen zwei Verfahren zur Verfügung. Das erste Verfahren richtet sich nach dem -> vertical blanking interval (UNIT_VBLANK) und wird für langfristige Messungen empfohlen, weil die Genauigkeit dann ziemlich konstant bei $\pm 1/60$ Sekunde liegt. Das zweite Verfahren (UNIT_MICROHZ) benutzt einen programmierbaren Timer in einem der beiden 8520 CIA-Chips und ist empfehlenswert für kurzfristige Messungen. In beiden Fällen kann nur eine Mindestzeit und nicht eine genaue Zeit gemessen werden, weil die Zeitmessung vom Multitasking-Subsystem beeinträchtigt wird.

timer.library

Systemroutinen für Zeitarithmetik.

time slot allocation

(DMA) Von den 227,5 Speicherzugriffszyklen (siehe system timing frequencies) stehen 226 den auf den Speicher zugreifenden Subsystemen zur Verfügung. Die Systeme greifen in der folgenden Reihenfolge zu:

memory refresh (Speicherzellenaufrischung) 4 Zyklen; Laufwerk-DMA 3 Zyklen; Audio-DMA 4 Zyklen (1 Zyklus pro Kanal; 2 Datenbyte pro Kanal/Zyklus; Sprite-DMA 16 Zyklen (2 Zyklen pro Spritekanal; ein Datenwort pro Zyklus); Bitplane-DMA 80 Zyklen (bei 4 Bitplanes; 20 Zyklen zu jeweils einem Datenwort); frei 10 Zyklen

Bei mehr als vier niedrigauflösenden Bitplanes müssen Zyklen, die eigentlich der CPU zustehen, benutzt werden. Im hochauflösenden Modus werden die letzten beiden Zyklen der Sprite-DMA verwendet, wodurch Sprite 7 nicht mehr benutzt werden kann.

Um Zeit für das Auslesen überbreiter -> raster (Over-scan) zu haben, schaltet das System der Reihe nach die Sprite-DMA 7 bis 1 (je nach Breite des rasters) ab. Nur Sprite-Kanal 0, der für den Mauszeiger benötigt wird, ist nicht von dem erweiterten Anspruch der Bitplane-DMA betroffen.

title bar

Die Titel- oder Namensleiste eines windows (siehe gadget/drag gadget).

TmpRas

(temporary raster) Ein vorläufiger Arbeitsspeicher, der einem -> RastPort zugewiesen wird. In diesem Übergangsspeicher werden grafische Darstellungen »vorab« erzeugt, bevor sie in das -> raster selbst kopiert werden. In der Regel muß ein TmpRas denselben Umfang wie das raster haben.

toggle

(Kipp-/Druckschalter) Solche Schaltfunktionen finden Anwendung in attributiven Menüfunktionen (z.B. Schriftart bei Textverarbeitung) und Booleschen gadgets. Die Bezeichnung »toggle select« bedeutet, daß eine gewählte Funktion so lange wirksam bleibt, bis diese durch Wahl einer anderen Funktion wieder abgestellt wird.

tool

(Werkzeug-) (Hilfs)Programm der -> Workbench.

track

(Diskettenspur) Eine der kreisförmigen »Rillen« einer Diskette, die in eine systemspezifische Anzahl von -> sectors (auch blocks) eingeteilt wird (siehe cylinder).

trackdisk.device

Systemroutinen/Hardware für die Steuerung eines Laufwerks (disk controller) und die Übertragung von Daten zwischen primary -> memory und secondary memory. Mit dieser Lösung kann der Amiga, bei entsprechender Programmierung des trackdisk.device, Disketten in verschiedenen Aufzeichnungsformaten verwalten (siehe pulse code modulation). Ab Version 1.2 werden Laufwerke im 3 1/2- oder 5 1/4-Zoll-Format verwaltet, unter 1.1 nur im 3 1/2-Zoll-Format. Für 5 1/4-Zoll-Laufwerke ohne automatische Diskettenwechselleitung gibt es den CLI-Befehl DISKCHANGE.

Die Datenübertragung erfolgt als »track i/o«, das heißt eine vollständige Spur wird von der Diskette in den Diskpuffer eingelesen bzw. auf die Diskette geschrieben. »sector i/o« ist nur begrenzt möglich. Wegen des Diskpuffers, der über einen -> DMA-Kanal bedient wird, benötigt jedes angeschlossene Laufwerk etwa 17 bis 20 KByte im -> Chip RAM, damit die Daten vom -> Blitter verschoben werden können. Um die Zu-

griffszeiten zu verkürzen, kann man ab Version 1.2 mit dem Befehl ADDBUFFERS den Umfang des Diskpuffers erhöhen. Die zusätzliche Speicherzuweisung läßt sich nicht rückgängig machen (es sei denn, man hat ein Laufwerk mit Ausschaltknopf).

Das Trackdisk.device kann bis zu vier Laufwerke (ab 1.2 in beliebiger Mischung aus 3 1/2 und 5 1/4 Zoll) verwalten. Davon sollen laut Hersteller nur zwei vom Amiga selbst mit Strom versorgt werden. Es wird auch davor gewarnt, 5 1/4-Zoll-Laufwerke ohne eigene Stromversorgung am -> drive port anzuschließen, weil viele dieser Geräte beim Einschalten eine zu hohe Stromaufnahme haben.

Transformer

(der Umwandler) Die Software-Lösung der IBM-Kompatibilität beim Amiga. Das Programm läuft in der vorliegenden Version nur unter 1.1. Mit dem Erscheinen von -> Sidecar ist die weitere Entwicklung für die Version 1.2 eingestellt worden.

Nur Programme, die weder auf eine IBM-Grafikkarte (CGA, EGA, Hercules) noch auf das System-ROM des IBM-Computers zugreifen, können unter Transformer laufen. In anderen Fällen muß man Sidecar benutzen. Da IBM-Programme unter Transformer auch etwas langsam laufen, gibt es ein Public-Domain-Programm namens Speedy3 für einen schnelleren Ablauf.

translator.library

Die Systemroutinen für die Umwandlung von Zeichenketten in Phonemcodeketten (siehe speech synthesis).

trap

(Falle) Eine Art -> exception, die als Folge einer Operation der CPU entsteht. Anders als bei einer exception muß eine task auf eine trap-Meldung sofort reagieren, sonst wird die Ausführung des tasks abgebrochen (»task held«). Viele traps ergeben sich aus CPU-Fehlerzuständen, wie z.B. Division durch Null, Adressenfehler usw. Aus diesem Grund werden traps im »supervisor mode« verarbeitet, das heißt der Systemstapelspeicher (supervisor stack) nimmt die gesicherten Registerinhalte auf. Diesen wird eine »longword exception number« hinzugefügt (siehe guru meditation).

Trashcan

(Abfalleimer/Papierkorb) Eine besondere »Schublade« der ->

Workbench. In dieser Schublade abgelegte Icons stellen Dateien dar, die zum »Wegwerfen« bestimmt sind. Die im Trashcan abgelegten Dateien bleiben solange erhalten, bis die Menüfunktionen »empty trash« (Müll rausbringen) gewählt wird. Danach sind sie gelöscht. Anders als andere Schubladen kann Trashcan weder in eine andere Schublade verlegt noch - selbst im völlig leeren Zustand oder mit einem anderen Namen versehen - vom Anwender gelöscht werden. In der Regel wird diese Schublade beim Formatieren einer Diskette automatisch auf der Diskette erzeugt. Seit Version 1.2 verfügt der CLI-Befehl FORMAT über eine »NOICONS«-Funktion, welche die Erzeugung eines Trashcans verhindert.

troubleshooting

(Fehlersuche) Die Suche nach Hinweisen auf mögliche Fehlerquellen und Lösungswege. Die (gedankliche) Vorstufe zum »debugging«, der gezielten Fehlerbehebung.

tutorial

(Kurs) Anders als »Kurse« in deutschen Computerzeitschriften, die sich in der Regel über mehrere Folgen erstrecken, ist das angelsächsische »tutorial« üblicherweise ein einziger, in sich abgeschlossener und Wissen vermittelnder Artikel (oder Kapitel eines Handbuchs) zu einem einzigen Aspekt der Computerei.

UART

(universal asynchronous receiver/transmitter) Ein Baustein, der die Verbindung mit Peripheriegeräten über den seriellen Port steuert (siehe Paula/Portia).

unary minus

Vorzeichenwechsel.

unit

(unit: Einheit) In der Computersprache, eine Einheit, ein Teil des Computers, der eine bestimmte Aufgabe zugeteilt ist (CPU: »Central Processing Unit«). Aber auch Peripheriegeräte, die nicht direkt zum Computer gehören, wie Disketten-Laufwerke, Drucker oder Festplattenlaufwerke, werden als peripheral units bezeichnet. Dies ist in erster Linie auch die Bedeutung eines units beim Amiga.

user interface

-> operating system.

UserPort

-> IDCMP eines window.

UTILITIES

Schublade der -> Workbench. Sie enthält -> tools, welche die »Schreibtisch«-Funktion der Workbench unterstützen (-> Notepad und Calculator).

utility program

(Hilfsprogramm). Ein (meist relativ kurzes) Programm, das die Bedienung oder Programmierung des Computers in einem bestimmten Aspekt verbessert (erweitert, Fehler behebt), wie z.B. -> spooler, Menüeditor, Texteditor, -> shell usw.

VDU

(visual display unit) Datensichtgerät, Monitor oder Bildschirm.

vector graphic display

Eine Bilddarstellung, die aus Linien aufgebaut wird. Gegensatz: -> raster graphic, die zeilenweise aufgebaut wird. Der Unterschied zwischen den Verfahren läßt sich am besten mit dem Vergleich Matrixdrucker (Punkte, zeilenweise = raster graphic) und Plotter (Linien von unterschiedlicher Länge = vector graphic) verdeutlichen.

Version

Die jeweilige Fassung eines Programms oder eines Betriebssystems. Unter 1.1 ist diese Information nur über Menüoption abrufbar, ab 1.2 gibt es auch einen entsprechenden CLI-Befehl. In frühen Meldungen aus Amerika zum Entwicklungsstand des Amigas war von Kompatibilitätsproblemen zwischen Version 24 und 27 die Rede. Die Ursache dafür war vermutlich der Wechsel vom ursprünglichen Betriebssystem zu Amiga-DOS. Version 28 und 29 wurden in den Vorserienmaschinen benutzt, die den amerikanischen Software-Entwicklern zur Verfügung gestellt wurden. Version 30 wurde mit den ersten amerikanischen Geräten ausgeliefert (Version 1.0); Version 31 (Version 1.1) erschien zur europäischen Einführung des Amiga und Version 33 (1.2) ist die ROM-Version älterer Modelle des A500 und A2000. Die neuesten Amigas sind mittlerweile mit der Betriebssystem- und Workbenchversion 1.3 erhältlich. Kickstart 1.4 und Workbench 1.4 sind bereits in Vorbereitung.

Beim Programmieren kann die Versionsnummer insbesondere einer -> library (Programmbibliothek) wichtig sein, denn beim Öffnen einer library wird eine Versionsnummer erwartet. Ist diese Angabe un-

gleich Null (eine beliebige Version der library wird akzeptiert), kann die library nur dann erfolgreich geöffnet werden, wenn die aktuelle Versionsnummer entweder mit der angegebenen identisch oder höher ist.

version.library

Die Systemroutinen für die Verarbeitung von Version-Aufrufen.

vertical blanking interval

(auch vertical retrace) Die Zeit, in der der Elektronenstrahl keine Daten am Bildschirm darstellen kann, weil er sich gerade oben bzw. unten am Bildschirm außerhalb des -> display field befindet. Die Mindestdauer dieser Darstellungsunterbrechung beträgt 20 Bildschirmzeilen (etwa 1/560 Sekunde), die normalerweise ab der obersten Zeile des Bildschirms berechnet wird. Während dieser Zeit läuft eine Unterbrechnungsroutine (interrupt), in der das System verschiedene Systemoperationen (vertical blanking tasks, wie Auffrischung der Darstellung ohne die Gefahr, daß Daten während der Phase des Bildaufbaus plötzlich verändert werden) ausführt. Die Höchstdauer errechnet sich aus der Anzahl »freier« Bildschirmzeilen nach Abzug des verwendeten display field (200/256 Zeilen) (siehe horizontal blanking, overscan).

video persistence

Die Leuchtdauer der Phosphorbeschichtung eines Bildpunkts am Bildschirm, der vom Elektronenstrahl in den Farben Rot, Grün, Blau zum Leuchten gebracht wird. Je länger die Leuchtdauer, um so stabiler ist das erzeugte Bild. Um -> hires flicker im -> Interlace-Modus zu vermeiden, wird die Benutzung eines »high persistence« Monitors mit lang nachleuchtender Beschichtung empfohlen.

View

Die Bilddarstellung des Amiga besteht aus einem oder mehreren View-Ports (-> ViewPort). Die Datenstruktur View enthält unter anderem die Adresse des ersten View-Ports und der beiden für die Darstellung benötigten -> Copperlisten.

ViewPort

Ein View-Port ist ein mit eigenen Bildattributen ausgestatteter, rechteckiger Darstellungsbereich auf dem Bildschirm. Er bildet den Hintergrund, auf dem windows erzeugt werden und damit die Grundlage der Bild-

schirmverwaltung unter Intuition (siehe layer). View-Ports lassen sich nicht nebeneinander positionieren, sondern müssen übereinander mit jeweils einer Bildschirmzeile Abstand angeordnet werden. Die Datenstruktur ViewPort enthält die Bildattribute: Farbkombination, Auflösung, Höhe und Breite, Koordinaten...

virtual

(scheinbar) Gegenstand, der die Attribute und Funktionen eines anderen Gegenstandes emuliert.

- virtual machine: ein »Scheinc-computer«. Da beim Multitasking ein -> process - zumindest zeitweilig - die volle Kontrolle über alle Systeme des Computers hat, kann ein Multitasking-System aus mehreren virtuellen Computern bestehen.

- virtual processor: die Funktion einer task innerhalb eines process

- virtual screen: ein »Scheimbildschirm«. Ein screen von Intuition

- virtual terminal: ein »Scheinterminal«. Obwohl sich Fenster nur einen Bildschirm und eine Tastatur teilen, verhält sich ein Fenster wie ein -> terminal mit eigenem Ein- und Ausgabegerät

- virtual sprite (VSprite): -> GELS

VLSI

(very large scale integration) Ein Baustein, der mehr als 100000 Transistoren enthält.

volume

(wörtlich Band; z.B. Band eines Lexikons) Eine Diskette des Amiga.

voice

(Stimme) Ein Audiokanal des Amiga.

Wack

Ein »debugger«, also ein Hilfsprogramm, mit dem man Fehler in einem abgestürzten Programm finden und eventuell beheben kann. Eine Fassung mit reduziertem Befehlssatz ist als ROMWack im -> Kernel enthalten und bildet die debug()-Funktion von -> exec.

Um das Programm benutzen zu können, ist der Anschluß eines auf 9600 Baud Übertragungsgeschwindigkeit eingestellten Terminals am seriellen Port des Amiga notwendig. Bei einigen der älteren Programme kann man nach Erscheinen einer -> guru meditation nr. das ROM-Wack durch Klicken der rechten Maustaste aktivieren.

Eine andere Methode ist das Betätigen der DEL-Taste am externen Terminal. Der Befehlssatz des Wack wird im -> RKM Exec ausführlich beschrieben.

Es handelt sich bei Wack um ein »memory editor program«, das heißt die Daten werden direkt im Speicher des Computers geändert. Alle Adressensprünge erfolgen relativ zu einer gegebenen Basisadresse (in der Regel die Anfangsadresse der abgestürzten Routine).

Laufende Programme lassen sich mit ROM-Wack »überwachen«. Dazu muß die Anweisung LOADWB in der -> startup-sequence in »LOADWB-debug« geändert werden. In der Menüleiste der Workbench erscheint dann rechts neben »Special« ein weiteres Menü, dessen einziger Unterpunkt ROM-Wack startet.

wait list

(Warteliste) Die Exec-Liste der tasks, die auf ein -> event warten, um wieder aktiv zu werden (siehe task state).

warm boot

-> reset, reboot.

window

(Fenster) Das Grundelement der Bildschirmdarstellung unter Intuition. Die Schnittstelle zwischen Programmcode und dem Benutzer, also die optische Darstellung eines -> virtual terminal. Ein Fenster erscheint immer relativ zum -> origin eines Screens und kann innerhalb desselben frei bewegt, positioniert, in der Größe verändert und hinter oder vor anderen windows »gestapelt« werden. (siehe gadget)

Der Amiga unterstützt vier verschiedene Fenstertypen:

- backdrop: erscheint grundsätzlich hinter allen anderen windows (back: hinter).

- borderless: besitzt keine »Einrahmung«; kann in Verbindung mit »backdrop« eine Art Screen simulieren

- GimmeZeroZero: ein Fenster mit getrenntem Speicher für die Einrahmung. Im Gegensatz zu anderen Fenstertypen hat dieses die Koordinaten 0,0 (Zero, Zero) tatsächlich in der linken oberen Ecke des Darstellungsbereiches und nicht in der Ecke unterhalb der Einrahmung.

- SuperBitMap: ein Fenster mit eigenem Speicher für den Darstellungsbereich (siehe refresh).

Die verschiedenen Windowtypen lassen sich auch kombinieren.

window pointer

-> mouse pointer.

WindowPort

-> IDCMP.

Windows Amiga-Basic

Fenster Nummer 1 ist das Ausgabefenster von Amiga-Basic und daher nicht immer unter voller Programmkontrolle, weil der Benutzer Größe und Position ändern kann. In diesem Fenster erscheinen auch die Fehlermeldungen von Amiga-Basic und nicht in vom Programm definierten Fenstern.

Wegen der Fenstereinrahmung können von Amiga-Basic aus definierte Fenster nicht den vollen Bildschirmumfang einnehmen. Die Maximalgrößen betragen bei

niedriger Auflösung:

0,0 - 311,186 ohne size gadget

0,0 - 297,186 mit size gadget

mittlerer Auflösung:

0,0 - 631,186 ohne size gadget

0,0 - 617,186 mit size gadget

Im Interlace-Modus erhöht sich die Grenze 186 auf 386. Bei den PAL-Amigas betragen die maximalen y-Koordinaten 242 und 498.

word alignment

Ein »word« ist ein 16 Bit breiter Datenwert (2 Byte). Für verschiedene Operationen werden

Daten benötigt, die an der Grenze eines solchen Wertes beginnen, also an einer genau durch zwei teilbaren Speicheradresse. Werte, die auf diese Weise im Speicher abgelegt werden, bezeichnet man als »word aligned« (siehe long word und alignment).

Workbench

(wörtlich »die Werkbank«; meistens »Arbeitsstisch« oder »Schreibtisch«) Konsequentermaßen werden die Programme, mit denen der Computer arbeitet, Werkzeuge (Tools) genannt. Die Workbench selbst hat zwei Aufgaben:

1. Als »Werkbank« der Hauptbildschirm (screen) von Intuition; eine »Arbeitsfläche« für die Programme.

2. Als »Werkzeug« selbst ein Programm, mit dem sich Dateien aus dem -> filing system von Amiga-DOS bearbeiten lassen. In dieser Eigenschaft gilt die Workbench neben Amiga-DOS als Ausgangspunkt für die Erzeugung von -> processes und ist somit auch ein wesentlicher und zugleich komfortabler Bestandteil des Multitasking-Systems.

Workbench disk

Ein Mehrzweckbegriff: Eine Diskette mit Programmen, die

-> Icons haben und mit der Maus gestartet werden können, eine bootfähige Diskette (siehe system disk) oder eine Programmdiskette, die einen Workbench-Screen benutzt.

workspace

(Arbeitsspeicher) Ein Speicherbereich, in dem eine Routine oder ein Codesegment (meist lokale) Datenwerte ablegen kann. Ein -> hunk_bss in einem Codesegment erfüllt diese Funktion ebenso wie ein -> stack frame.

writable control store

(WCS; auch write-once memory (WOM) oder Kickstart-RAM). Anstelle von ROM-Chips hat der A 1000 einen besonderen RAM-Speicher (siehe memory map), in welchen das Betriebssystem (-> Kernel) von der Kickstart-Diskette gelesen wird. Nach Übernahme des Kernels sperrt das Betriebssystem diesen Speicher für Schreibzugriffe, und damit arbeitet er wie ein ROM-Speicher.

x/y coordinates

Die Position eines Punktes am Bildschirm relativ zum »Ursprung« (origin), der in der linken oberen Ecke des Bildes liegt. Die y-Koordinate bestimmt die Zeile und die x-

Koordinate die Spalte des betreffenden Bildpunktes. Bei -> Sprites hat die y-Koordinate bei der Feststellung von Kollisionen Vorrang vor der x-Koordinate.

zero crossover

Der »Nullpunkt« in einer Wellenform. Nur an dieser Stelle lassen sich zwei verschiedene Wellenformen ohne weiteres miteinander verbinden, ohne daß dabei klickende Störgeräusche entstehen.

zorro

Die von Commodore-Amiga entwickelte Steckerbelegungsvorschrift für selbsteinbindende Erweiterungskarten (siehe auto.config). Anders als der -> expansion bus, der eine 86polige Belegung hat, schreibt die Zorro-Vorschrift einen 100poligen Stecker vor. Solche Karten werden in der Regel über ein -> expansion chassis am Computer angeschlossen. Leider gibt es für den Amiga zwei unterschiedliche Belegungen. Bei den rechteckigen Karten des Amiga 2000 sind sieben Pins anders belegt als bei den quadratischen Karten des Amiga 1000. Beide Vorschriften verlangen, daß mehrere Karten nach dem -> daisy chain-Prinzip zusammengeschaltet werden können.

Aus unserem Bücher-Regal

Drei Dinge braucht der Mann: einen schnellen Amiga, die für ihn richtigen Bücher und viel Freizeit.



Amiga 500 für Einsteiger

Das Buch von Christian Spanik liegt in der 5. überarbeiteten Auflage vor. Der Autor geht darin sehr geschickt auf die Fragen von Computerneulingen ein und beantwortet sie auch für den Laien verständlich. Dabei achtet er darauf, daß erst die gleichen Voraussetzungen für alle Leser geschaffen werden, was für den Anfänger sehr wichtig ist. So beginnt das Buch mit dem Auspacken, Aufstellen und Anschließen des Amiga. Nach diesem, für den Leser sehr erfreulichen, kurzen Einstieg geht es rasant mit der Bedienung des Amiga los. Dabei werden beide Betriebssystemversionen 1.2 und 1.3 berücksichtigt. Wobei grundsätzlich von der aktuellen Version 1.3

ausgegangen und nur bei Abweichungen auf die Version 1.2 hingewiesen wird. Auf einem etwa 200seitigen Kurs wird die grafische Oberfläche, genannt Workbench, leicht verständlich und vollständig behandelt. Darauf folgt eine Einführung in »AmigaBasic«, welche sich hauptsächlich mit kleinen Demos und Grafik-Spielereien beschäftigt. Ein Kritikpunkt an dem Taschenbuch ist der anschließende Teil über das AmigaDos oder auch CLI genannt, die Text-orientierte Benutzeroberfläche des Amiga. Erstens ist es völlig unverständlich warum dieser Teil erst nach der Einführung in »AmigaBasic« folgt und zweitens ist die Einführung für den Stellenwert, welche diese Oberfläche auf dem Amiga einnimmt, zu kurz gehalten. Abgerundet wird das ganze von

einer kleinen ersten Hilfe, einem sehr kurzen Fachwörterlexikon, einer Kurzübersicht des Icon-Editors und der Vorstellung von Btx am Amiga.

»Amiga 500 für Einsteiger« wendet sich an den Computerlaien und sollte am besten beim Computerkauf mit erworben werden. Dem Leser mit Vorkenntnissen wird dieses Buch nicht weitreichend genug sein, da vor allem der für den interessierten Amiga-User wichtige AmigaDOS-Teil etwas dürrig ist. Besonders auffallend ist der jugendlich flachsige Schreibstil des Autors, der nie Langeweile aufkommen läßt, die vielen Bilder und Karikaturen sowie der brandaktuelle Stand der Ausgabe, mit dem keines der hier besprochenen Büchern konkurrieren konnte. *pe*

Amiga 500 für Einsteiger; Christian Spanik; Data Becker Verlag; 393 Seiten; Preis 39 Mark.



dungen und – für meinen Geschmack völlig deplaziert – eine Auflistung des 68000-Prozessor-Befehlssatzes.

Das Konzept des Buches weist eindeutige Mängel auf, und ist für Computerlaien nicht sonderlich geeignet. Dagegen findet der Umsteiger hier einen schnellen Einstieg und der Hardwarebastler Daten und Werte, um gleich loszulegen. Damit bleibt nur noch zu sagen, daß das Buch seit dem Erscheinen 1987 nicht mehr überarbeitet wurde. *pe*

Amiga 500 – Das Einsteigerbuch; Thomas Binzinger; Sybex Verlag; 280 Seiten; Preis 29,80 Mark.

Amiga 500-Buch

Das Standardwerk von Markus Breuer richtet sich nicht ausschließlich an den Anfänger, sondern ist für jeden Amiga 500-Besitzer interessant. So beginnt der Autor mit einer etwas langgezogenen Vorstellung des Amiga. Erst dann geht es mit der Arbeit am Computer los.

Zuerst werden die Themenbereiche Workbench, die Programme der im Lieferumfang enthaltenen Disketten und die Besonderheiten von Intuition (grafische Benutzeroberfläche des Amiga) in einer für den Leser verwirrenden Reihenfolge besprochen. Dafür sind die einzelnen Programme und Funktionen verständlich und sehr ausführlich erklärt. Aber auch hier werden die Erklärungen breitgetreten und sind für die Belange des Einsteigers, der ja darauf brennt, schnell voranzukommen, zu detailliert. High-Light in diesem Buch ist die folgende Einführung in das AmigaDOS. Sie ist sehr ausführlich und fachlich fundiert geschrieben und läßt auch den Anfänger mit seinen Problemen nicht im Stich. Vollendet wird dieser gelungene Abschnitt durch ein CLI-Lexikon, in dem alle AmigaDOS-Befehle alphabetisch aufgeführt sind und ihre Funktion kurz beschrieben wird. Anschließend finden Sie noch Tips und Tricks, eine kurze Erklärung zur Spezial-Hardware des Amiga und einen Hinweis auf AmigaBasic.

Für den meist ungeduldigen Einsteiger sind die 50 Seiten Einleitung einfach zu lang. Er wird mit anfangs eher nebensächlichen Informationen überschüttet. Dafür eignet sich dieses Werk vorzüglich als Begleitlektüre (z.B. für dieses Heft) und Nachschlagewerk für Workbench und AmigaDOS. Leider ist die aktuelle Betriebssystemversion 1.3 in dieser Ausgabe noch nicht berücksichtigt und somit sind einige Ausführungen unvollständig oder fehlerhaft. *pe*

Amiga 500-Buch; Markus Breuer; Markt & Technik Verlag; 490 Seiten; Preis 49 Mark.

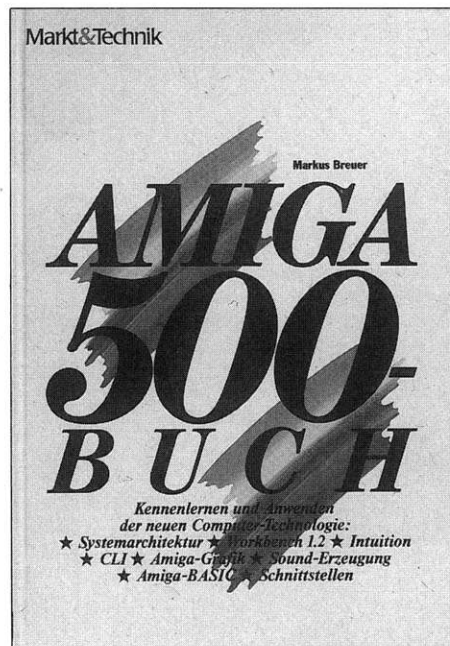


Amiga 500 - Das Einsteigerbuch

Das Buch von Thomas Binzinger erweckt durch seinen unglücklich gewählten Titel beim Käufer eine falsche Vorstellung. Da es sich nicht wie erwartet um ein Amiga Anfängerbuch handelt, sondern um einen Schnellkurs für Umsteiger.

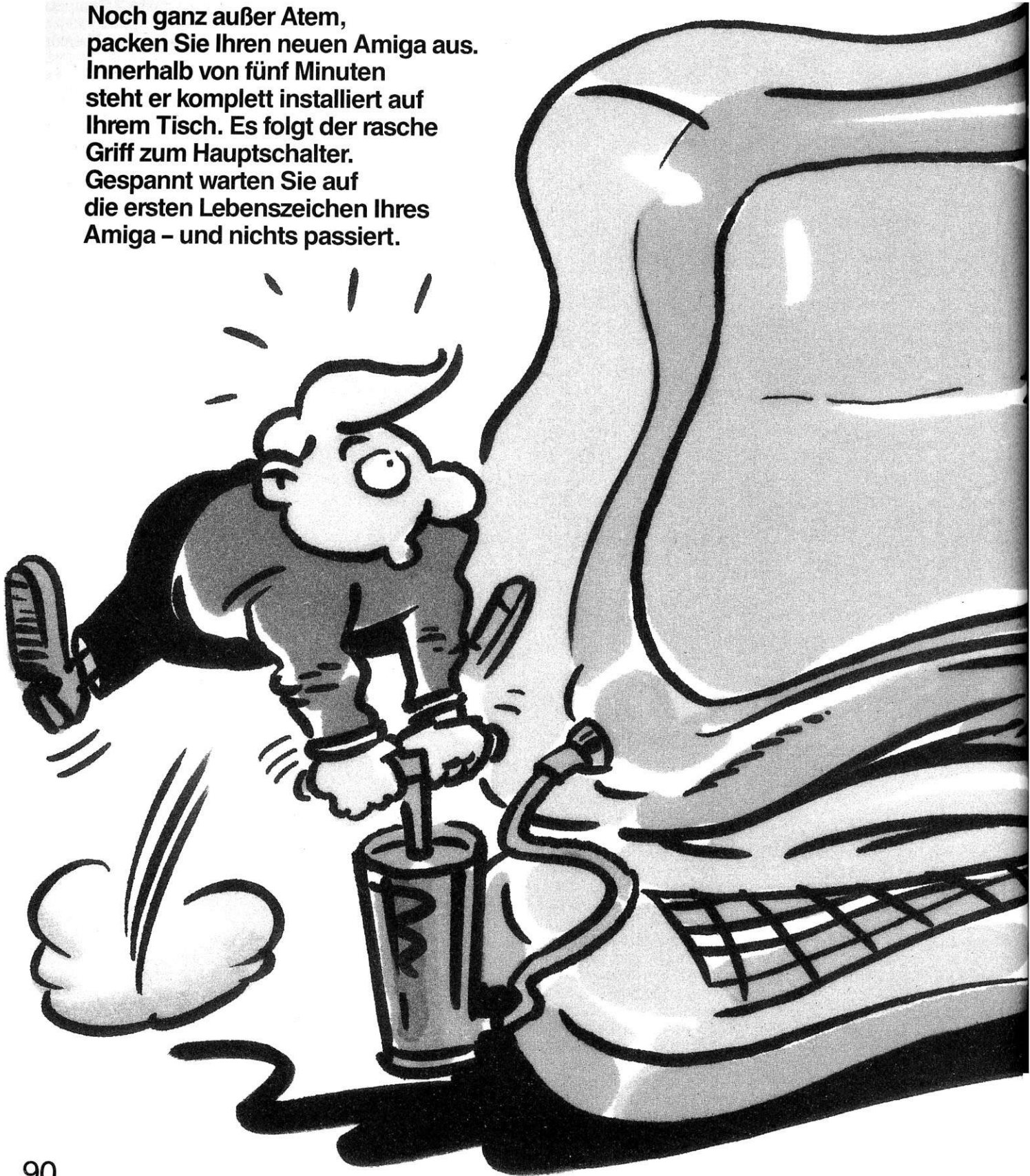
Der Autor beginnt zwar mit dem Aufstellen und Anschließen des Computers, geht dann aber zu schnell für den Computerlaien und oft nur stichpunktartig auf die Bedienung der Workbench und der Programme auf den mitgelieferten Disketten ein. Dies kommt dem Umsteiger entgegen, stellt aber für den Einsteiger ebenso eine

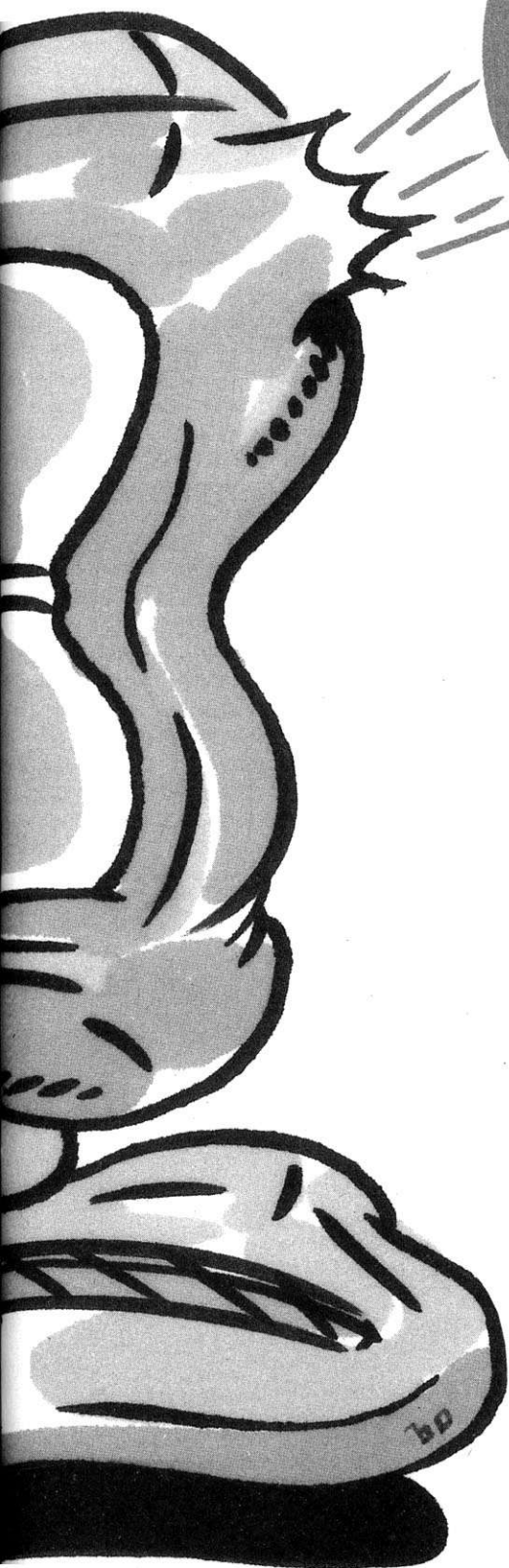
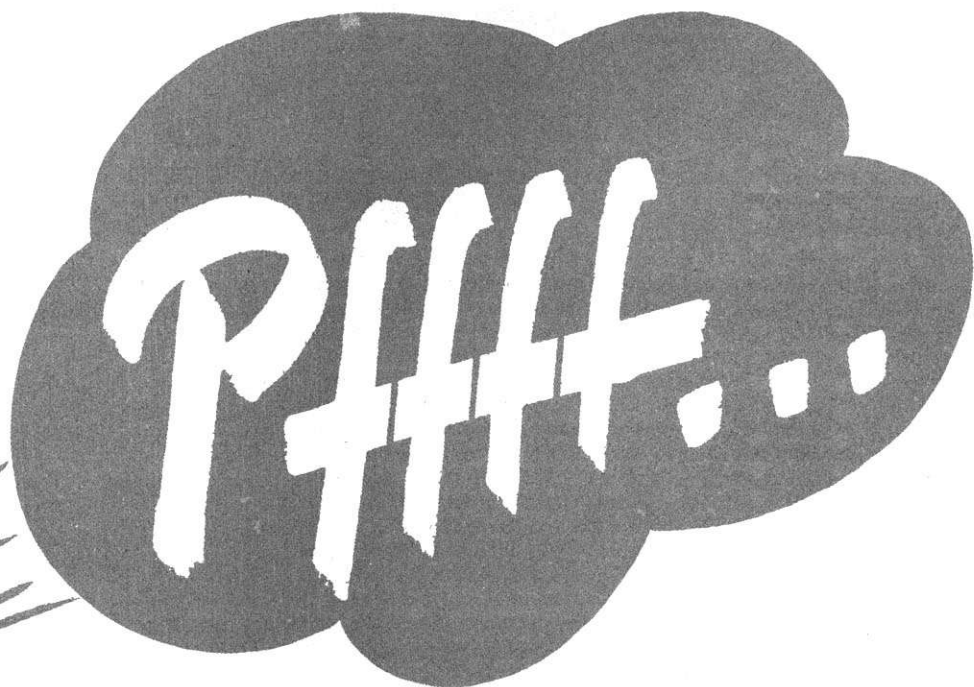
Hürde dar wie die Tatsache, daß sich alle Angaben auf die bereits überholte Betriebssystemversion 1.2 beziehen. Der darauf folgende Teil, eine ganz grob umrissene Einführung in die Hardware-Struktur des Amiga, ist für den Computer-Freak zu mangelhaft und für den Anwender uninteressant. Die anschließenden Kapitel schneiden nur kurz die beiden wichtigsten Themen AmigaBasic und AmigaDOS an, und sind als Einführung unzureichend. Bemerkenswert und für den Bastler sehr hilfreich, ist die Auflistung der Pinbelegung sämtlicher Schnittstellen des Amiga 500 und des Amiga 1000. Abschließend finden Sie noch eine Reihe von Tips und Tricks, eine Liste der AmigaDOS-Fehlermel-



PANNENHILFE

Noch ganz außer Atem, packen Sie Ihren neuen Amiga aus. Innerhalb von fünf Minuten steht er komplett installiert auf Ihrem Tisch. Es folgt der rasche Griff zum Hauptschalter. Gespannt warten Sie auf die ersten Lebenszeichen Ihres Amiga – und nichts passiert.





von Dietmar Inäbnit

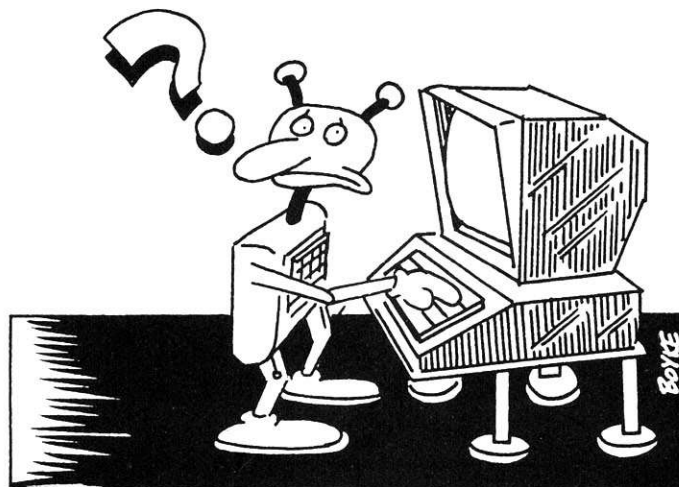
Häufig sind es nur kleine Ursachen, die einem Amiga-Besitzer das Leben schwermachen. Hier findet nun der Amiga-Neuling sowie der Amiga-Profi einige nützliche Hinweise, wie er die häufigsten Probleme auf einfachste Art und Weise lösen kann. Alle Ursachen sind nach Symptom, Ursache und Lösung aufgeteilt.

Natürlich können hier nicht alle Probleme bzw. Tips und Tricks erläutert werden. Diese Aufstellung kann nur ein paar Probleme aufzeigen. In erster Linie soll diese Aufstellung ein Anreiz sein, die eigenen Erfahrungen aufzuschreiben und sich so nach und nach sein ganz privates Amiga-Hilfelexikon aufzubauen. *pe*

Die Maus

Symptom:	Der Mauszeiger bewegt sich nicht.
Ursache 1:	Wenn Sie Ihren Amiga gerade erst erworben haben, und dies Ihre ersten Versuche sind, dann haben Sie vermutlich vergessen, die Transportsicherung der Maus zu entfernen. Bei dieser Transportsicherung handelt es sich um zwei kleine Plastikstreifen, die zur Fixierung der Rollkugel verwendet werden.
Lösung:	Nehmen Sie die Maus zur Hand. Drehen Sie nun die Maus auf die Rückseite. In der Mitte des Mausbodens sehen Sie eine kleine Abdeckung. Drücken Sie nun die Abdeckung in Pfeilrichtung (Die Pfeile befinden sich auf der Abdeckung). Nun können Sie die Abdeckung abnehmen und die beiden Plastikstreifen entfernen. Bitte beachten Sie, daß die kleine Gummikugel der Maus sauber bleiben muß. Nachdem Sie alles wieder zusammengesetzt haben, wird sich wie normalerweise erwartet der Mauszeiger bewegen.
Ursache 2:	Sie haben die Transportsicherung bereits entfernt und der Mauszeiger bewegt sich immer noch nicht.
Lösung:	Sie haben vermutlich die Maus am falschen Port eingesteckt. Die Maus muß beim Amiga immer beim Port 1 eingesteckt sein.
Symptom:	Der Mauszeiger ist völlig unkontrollierbar.
Ursache:	Der Untergrund, auf dem die Maus bewegt wird, ist zu rutschig. Für die Maus komplett ungeeignet sind glatte Oberflächen wie Glas und Marmor usw. Die Gummikugel rutscht dann über die Oberfläche und kann keine richtigen Informationen an die Abnehmerrädchen in der Maus liefern.
Lösung:	Die einfachste Lösung ist der Kauf eines Mouse-Pad.

Symptom:	Der Mauszeiger bewegt sich nur ruckartig und läßt sich nur sehr schwer positionieren.
Ursache:	Die Maus ist verschmutzt und muß gereinigt werden. Je schlechter die Unterlage ist, auf der Sie mit der Maus arbeiten, desto öfters müssen Sie Ihre Maus reinigen. Tun Sie Ihrer Maus etwas Gutes an und kaufen Sie ihr ein Mousepad. Die Ausgabe von ein paar Mark erspart Ihnen das lästige und sehr zeitaufwendige Reinigen der Maus. Es ist noch zu beachten, daß ein übertriebenes Reinigen der Maus eher schadet als nutzt.
Lösung:	Nehmen Sie die Abdeckung an der Unterseite der Maus ab. Entfernen Sie die kleine Gummikugel. Im Innern der Maus entdecken Sie drei kleine Rädchen. Immer wenn Sie die Maus bewegen, dreht sich die Kugel und überträgt diese Bewegung auf die Rädchen. Diese Rädchen senden bei Drehung elektrische Impulse zum Amiga, die ihm genau sagen, in welche Richtung und wie weit Sie die Maus bewegt haben. Wird eine Maus auf einer verschmutzten Unterlage bewegt, so transportiert die Kugel die Schmutzpartikel zu diesen kleinen Rädchen. Zum Reinigen dieser Rädchen verwenden wir Wattestäbchen. Bei starker Verschmutzung können die Wattestäbchen mit einem Reinigungsmittel getränkt werden. Als Reinigungsmittel sollten Sie Flüssigkeiten einsetzen, die auch zur Reinigung von Tonköpfen bei Kasettengeräten verwendet werden. Sehr zu empfehlen sind auch die diversen im Handel angebotenen Reinigungssets für Computer. Auf keinen Fall dürfen ätzende Flüssigkeiten oder sogenannte Universalreiniger verwendet werden. Lebensgefährlich für die Maus ist auch das Abkratzen der Rädchen mit einem scharfen Gegenstand. Die Gummikugel reinigen wir mit einem sauberen Tuch. Bei der Gummikugel muß darauf geachtet werden, daß die Oberfläche nicht verletzt wird. Sofern Sie alles sauber gereinigt und richtig zusammengesetzt haben, wird sich der Mauszeiger wieder ganz exakt positionieren lassen.



Workbench

Symptom:	Man muß mit der Maus über den halben Tisch fahren und am Bildschirm geht der Mauszeiger nur ein paar Zentimeter weiter.
Ursache:	In den Preferences ist die Übersetzung der Maus falsch eingestellt.
Lösung:	In den Preferences muß die Übersetzung der Maus anders eingestellt werden.
Symptom:	Der Doppelklick funktioniert nicht mehr.
Ursache:	Die Zeitspanne für die beiden Klicks ist zu kurz eingestellt.
Lösung:	Stellen Sie in den Preferences die Einstellung für den Doppelklick zurück.
Symptom:	Der Bildschirm ist nicht zentriert.
Ursache:	Es wurde in den Preferences die Einstellung des Bildschirms verändert.
Lösung:	Zentrieren Sie in den Preferences den Bildschirm wieder.
Symptom:	Mit der Maus kann nichts mehr angeklickt werden.
Ursache:	Beim Editieren des Mauszeigers wurde der sog. Hot Point verschoben.
Lösung:	In Edit Pointer muß der Hot Point wieder neu gesetzt werden.
Symptom:	Die Preferences müssen jedesmal wieder neu eingestellt werden.
Ursache:	Sie haben vergessen die Änderungen in den Preferences zu speichern.
Lösung:	Bei Einstellungen in den Preferences ist zu beachten, daß sie nur dauerhaft sind, wenn Sie die Option Save verwenden. Verlassen Sie hingegen die Preferences mit Use, Chancel oder schließen einfach das Fenster, dann sind die Einstellungen nur für eine Arbeitssitzung gültig. Das heißt, wenn der Rechner neu gebootet wird, haben Sie wieder Ihre alten Einstellungen.
Symptom:	Beim Starten eines Programms erscheint die Fehlermeldung: Error while opening Amiga-Basic.
Ursache:	Sie haben versucht ein Basic-Programm zu starten, ohne daß sich das Amiga-Basic auf dieser Diskette befindet.
Lösung:	Beim Starten eines Basic-Programms erwartet der Amiga immer, daß sich das Basic auf der gleichen Diskette befindet. Sie müssen also nichts weiter tun, als das Amiga-Basic auf diese Diskette zu kopieren. Das Amiga-Basic finden Sie auf der Extras-Diskette.
Symptom:	Der Amiga bootet nicht, obwohl eine Diskette eingelegt ist.
Ursache:	Sie haben eine nicht startfähige Diskette eingelegt.
Lösung:	Booten Sie den Amiga mit der Workbench und legen Sie nachher die andere Diskette ein. Um aus einer

Tastatur

Symptom:	Die Zeichen, die auf dem Bildschirm erscheinen, stimmen nicht mit denen auf der Tastatur überein.
Ursache:	Sie haben in der Startup-Sequence den falschen Tastaturtreiber. Mit dem Setmap-Befehl wird der gewünschte Tastaturtreiber eingestellt. Oft wird bei Public-Domain-Disketten automatisch der amerikanische Treiber geladen. Bei Programmen, die ins Deutsche übersetzt wurden, ist dies leider auch noch sehr oft der Fall.
Lösung:	Booten Sie den Amiga mit Ihrer Workbench. Wenn jetzt ein anderes Programm von der Workbench gestartet wird, ist immer noch der deutsche Treiber aktiv. Um direkt auf einer bootbaren Diskette den Treiber zu ändern, müssen Sie die Startup-Sequence ändern. In der Startup-Sequence steht z.B. »Setmap usa0«. Für den deutschen Treiber muß lediglich »usa0« durch ein »d« ersetzt werden.
Symptom:	Die Tastatur reagiert nicht.
Ursache:	Beim Amiga 1000 und 2000 könnte das Kabel nicht richtig eingesteckt sein.
Lösung:	Kabel kontrollieren und den richtigen Sitz der Stecker überprüfen.
Symptom:	Manche Tasten zeigen keine Reaktion.
Ursache:	Wenn dieses Symptom öfters auftritt, ist Ihr Amiga vermutlich defekt.
Lösung:	Geben Sie Ihren Amiga zur Reparatur.
Symptom:	Alle Buchstaben werden plötzlich groß geschrieben.
Ursache:	Auf der Tastatur wurde versehentlich die Taste Caps Lock gedrückt. Die kleine Kontrollampe leuchtet.
Lösung:	Ein zweiter Druck auf die gleiche Taste und alles ist wieder beim alten.

	nicht bootbaren Diskette eine bootbare zu machen, müssen Sie den Install-Befehl verwenden.
Symptom:	Nach dem Verschieben der Icons sind diese plötzlich verschwunden.
Ursache:	Sie haben vermutlich die Icons außerhalb eines Fensters positioniert und danach einen Reset durchgeführt.
Lösung:	Die Icons der Programme sind zwar gelöscht, aber die Programme sind noch auf der Diskette. Sie sollten nun lediglich den einzelnen Files neue Icons zuordnen.
Symptom:	Dauernde Fehlermeldungen, die auf irgendwelche Libraries verweisen.
Ursache:	Der Amiga verwendet immer die Libraries, die sich auf der Diskette befinden, mit der der Amiga gebootet wird.
Lösung:	Kopieren Sie die nötigen Libraries auf die Boot-Diskette in den Ordner Libs.
Symptom:	Der Trashcan (Mülleimer) läßt sich nicht löschen.
Ursache:	Der Trashcan wird automatisch vom Betriebssystem des Amiga erstellt und gehört auf jede Diskette.
Lösung:	Der Trashcan kann mit »Discard« nicht gelöscht werden.
Symptom:	Die Workbench läßt sich nicht mehr laden, nachdem die Ordner System und Utilities gelöscht wurden.
Ursache:	Beim Path-Befehl in der Startup-Sequence werden diese beiden Verzeichnisse erwähnt. Nachdem sie die Verzeichnisse gelöscht haben, findet der Path-Befehl sie nicht mehr und die Startup-Sequence wird abgebrochen.
Lösung:	Mit einem Editor die Path-Befehlszeile in der Startup-Sequence korrigieren.
Symptom:	Bei der Workbench 1.2 können Sie das Icon für das CLI nicht finden.
Ursache:	Das Icon wurde in den Preferences nicht aktiviert.
Lösung:	Gehen Sie in die Preferences und setzen Sie die Option CLI auf on und booten Sie den Amiga neu. Und siehe da – das CLI-Icon erscheint.
Symptom:	Beim Anklicken eines Icons erscheint die Fehlermeldung: File is not an Object Module.
Ursache:	Dieses Icon steht nicht für ein direkt ausführbares Programm. Das heißt, es ist vielleicht das Icon für eine ASCII-Datei oder ein Assembler-Listing usw.
Lösung:	Man sollte sich nur merken, daß nicht jedes Icon auch ein ausführbares Programm darstellen muß.
Symptom:	Der Rollbalken zeigt an, daß die Diskette voll ist, aber Sie sehen keine Icons.
Ursache:	Sie haben eine typische PD-Diskette vor sich. Aus Platzgründen wurde auf die Icons verzichtet. Das Icon ist nur ein optischer Zusatz für die Workbench. Jedes Programm ist auf dem Amiga auch ohne Icon lauffähig.
Lösung:	Disketten ohne Icons müssen mit dem CLI betrachtet werden.
Symptom:	Beim Starten eines Basic-Programms erscheint immer die Fehlermeldung: out of memory
Ursache:	Bei der Grundeinstellung des Amiga-Basic wird nur ein kleiner Teil des effektiven Speichers als Programmspeicher verwendet.
Lösung:	Bevor Sie Ihr Basic-Programm starten, müssen Sie den Befehl Clear eingeben. Zum Beispiel Clear 50000&.
Symptom:	Ihr Amiga hat vollkommen durchgedreht. Programme, die sonst immer gelaufen sind, stürzen dauernd ab. Disketten haben plötzlich read/write Errors. Ein Requester fordert Sie immer auf, den Schreibschutz zu entfernen.
Ursache:	Bingo - Ein netter kleiner Besucher, genannt Virus, hat sich in Ihrem Amiga eingenistet.
Lösung:	Hier hilft nur der Griff zum Viruskiller. Zum Beispiel das im AMIGA-Magazin 9/89 vorgestellte Programm ist ein idealer Viruskiller.

Monitor

Symptom:	Das Bild vom Monitor wird teilweise verzerrt.
Ursache:	Der Monitor wurde in der Nähe von einem starken Magnetfeld plaziert. Starke Magnetfelder können durch schlecht isolierte Stromleitungen entstehen. Die häufigsten Verursacher sind aber Lautsprecher, Stereoanlagen, Mikrowellenherd usw.
Lösung:	Stellen Sie den Monitor an einen anderen Platz. Wenn immer noch Verzerrungen zu sehen sind, muß der Monitor von einem Fachmann wieder neu justiert werden.
Symptom:	Das Bild des Monitors flimmert sehr stark.
Ursache:	Dieses Flimmern taucht beim Amiga immer dann auf, wenn er sich im sogenannten Interlace-Modus befindet. Der Interlace-Modus ist eine spezielle Bezeichnung für den Aufbau eines Monitorbildes. Im Interlace-Modus werden, sehr einfach ausgedrückt, immer zwei Halb-Bilder kurz hintereinander gezeigt. Bewirkt wird damit eine Erhöhung der grafischen Auflösung.
Lösung:	Kaufen Sie sich eine sog. Flicker-Fixer-Karte.
Symptom:	Der Monitor bleibt einfach dunkel, obwohl er eingeschaltet und alle Kabel richtig eingesteckt sind.
Ursache:	Der Monitor wurde auf den Video-Eingang umgeschaltet.
Lösung:	Öffnen Sie die Frontabdeckung des Monitors und drücken Sie die Taste (CVBS).
Symptom:	Der Monitor läuft, aber Sie haben keinen Ton.
Ursache:	Die beiden einzelnen Stecker am Monitorkabel sind nicht eingesteckt.
Lösung:	Stecken Sie die beiden Cinch-Stecker beim Amiga ein.
Symptom:	Der Monitor zeigt nur eine Farbe an. Zum Beispiel beim Einschaltbild ist alles rot oder grün.
Ursache:	Der Scart-Stecker beim Monitor ist nicht mehr richtig eingesteckt. Wenn der Monitor öfters bewegt wird, kann es schon einmal vorkommen, daß der Stecker leicht aus der Fassung rutscht und nicht mehr alle Kontakte verbunden sind.
Lösung:	Es muß lediglich der Scart-Stecker beim Monitor wieder richtig in die Fassung gedrückt werden.
Symptom:	Der Monitor arbeitet ganz normal. Plötzlich ist der ganze Bildschirm leer. Wenn Sie den Amiga ausschalten und neu booten, arbeitet der Monitor wieder wie gewohnt.
Ursache:	Gratuliere, Sie sind stolzer Besitzer eines Byte-Bandit-Virus. Dieser Virus schaltet einfach nach einer zufälligen Zeitspanne den Bildschirm ab.
Lösung:	Um den Bildschirm wieder zu reaktivieren, müssen Sie die Tasten (Alt + Amiga + Space + Amiga + Alt) gleichzeitig drücken. Jetzt können Sie ganz normal weiterarbeiten und Ihre angefangene Arbeit sichern. Um diesen hinterhältigen Virus wieder aus Ihrem System zu entfernen, benutzen Sie am besten eines der vielen Virus-Killer-Programme oder den Install-Befehl im CLI.
Symptom:	Die Regler für Color und Sharpness an der Vorderseite des Monitors reagieren nicht.
Ursache:	Es handelt sich hierbei nicht um einen Fehler. Die Regler haben nur eine Wirkung, wenn am Video-Eingang eine Signalquelle angeschlossen ist.
Lösung:	Keine Lösung.
Symptom:	Am Bildschirm haben Sie unten einen ca. 6 Zentimeter breiten Rand. Fenster lassen sich auch nicht auf die ganze Bildschirmseite vergrößern.
Ursache:	Beim Booten überprüft der Amiga anhand der Taktfrequenz des Netzes, ob er die PAL- oder NTSC-Auflösung nehmen soll. Bei starken Netzschwankungen kann es schon einmal vorkommen, daß der Amiga glaubt, er sei an ein amerikanisches Netz angeschlossen und aktiviert die NTSC-Auflösung.
Lösung:	Hier gibt es nur eine Hilfe. Ausschalten und neu booten.

Vollbremsung im Amiga

Kennen Sie das? Wieder einmal sitzen Sie vor dem brandneuen Spiele-Hit, der Highscore ist zum Greifen nah. Doch gerade im entscheidenden Moment klingelt das Telefon und noch während Sie den Hörer abheben, verlieren Sie Ihr letztes Raumschiff. Hätten Sie das Spiel anhalten können, wäre dies nicht geschehen. Auch beim Testen eigener Programme und bei Bildschirmfotos erweist sich eine Pausenfunktion als überaus nützlich.

Der »Soft-Stopper« ist ein kurzes Programm in Maschinensprache, das, einmal aufgerufen, unauffällig im Hintergrund auf ein Drücken der Enter-Taste (im Zahlenblock ganz rechts) lauert und sodann alle Amiga-Aktivitäten einfriert.

Die Bremse lauert im Hintergrund

Erst ein Druck auf die linke Maustaste setzt den Computer wieder in Gang.

Um bei unserem Beispiel mit dem Spielprogramm zu bleiben: Zum Anhalten des Spiels drücken Sie die Enter-Taste.

Highscore-Jäger und Programmtüftler aufgepaßt: Mit unserem »Soft-Stopper« genügt fortan ein Tastendruck, und schon frieren Sie Ihr Lieblingsspiel oder Ihr selbstgeschriebenes Programm ein. Ein Druck auf die Maustaste genügt, sofort geht der Spielespaß weiter.

Nun können Sie in aller Ruhe zu Mittag essen, Gäste empfangen oder tolle Bildschirmfotos schießen. Sie wollen weiterspielen? Ein Klick mit der linken Maustaste und schon geht die Punktehatz weiter.

Den Soft-Stopper geben Sie auf zweierlei Arten ein. Listing 1 ist der Assembler-Quellcode, den Besitzer des Seka-Assemblers unverändert eintippen und assemblieren können. Der Quellcode zeigt auch deutlich die prinzipielle Arbeitsweise des Soft-Stoppers, der alle laufenden Programme durch Sperren der Interrupts anhält. Das Stopper-Programm selbst liegt in einem Speicherbereich, der normalerweise nicht verwendet wird.

Verfügen Sie nicht über diesen Assembler, dann geben Sie einfach Listing 2 mit Amiga-Basic ein und starten

das Programm. Das Basic-Programm erzeugt aus den DATA-Werten den Soft-Stopper und schreibt diesen als ausführbares Programm auf Diskette.

Der richtige Aufruf

Nun müssen Sie nur noch das Programm einmal vor dem Start Ihres Spiels aufrufen. Tippen Sie dazu entweder im CLI

direkt »Soft-Stopper« ein oder speichern Sie das kurze Programm auf Ihre Spiele-Diskette und fügen Sie den Aufruf (»Soft-Stopper«) am Beginn der Startup-Sequence dieser Diskette ein. Die Startup-Sequence befindet sich im »S«-Ordner.

Der Soft-Stopper funktioniert auch bei »Interceptor«, »World Tour Golf« und anderen kommerziellen Spielen. Versuchen Sie Ihr Glück.

Fallen Ihnen noch Ergänzungen zu diesem Programm ein, so würden wir uns sehr über eine Zusendung freuen. Wie wäre es zum Beispiel mit einer Routine, die den Bildschirm dunkel schaltet, wenn einige Zeit kein Tastendruck und keine Mausbewegung registriert wurde? Oder vielleicht... (Ralf Dietrich/rs)

Programmname: SoftStopper

Computer: A500, A1000, A2000 mit Kickstart 1.2

Sprache: Assembler

```
*****
*      Soft-Stopper      *
*      von Ralf Dietrich *
*****
7  START:
8  EVEN
9  MOVEM.L D0-D7/A0-A6,-(A7)
10 LEA BSTART(PC),A0
11 LEA ENDE(PC),A1
12 MOVE.L #$100,A2
13 MOVE.L A2,A3
14
15 COPYSCHL:
16 MOVE.W (A0)+,(A2)+
17 CMP.L A0,A1
18 BNE.S COPYSCHL
19 JSR (A3)
20 MOVEM.L (A7)+,D0-D7/A0-A6
21 RTS
22
23 BSTART:
24 LEA.L IRBUFFER+2(PC),A0
25 MOVE.L $68,(A0)
26 LEA NEWIR(PC),A0
27 MOVE.L A0,$68
28 RTS
29
30 NEWIR:
31 MOVEM.L D0-D7/A0-A6,-(A7)
32 MOVE.B $BFEC01,D0
33 CMP.B #121,D0
34 BNE.S NOSPKEY
35 BSR PRGSTOP
36
37 NOSPKEY:
38 MOVEM.L (A7)+,D0-D7/A0-A6
39
40 IRBUFFER:
41 JMP $FFFFFFF
42
43 PRGSTOP:
44 MOVE.W SR,D0
45 MOVE.W # $2700,SR
46
47 PRGSTOPWAIT:
48 BTST #6,$BFEC01
49 BNE.S PRGSTOPWAIT
50 MOVE.B #120,$BFEC01
51
52 ERSTLOSLASSEN:
53 BTST #6,$BFEC01
54 BEQ.S ERSTLOSLASSEN
55 MOVE.W D0,SR
56 RTS
57
58 ENDE:
```

Listing 1. Der Seka-Quellcode des Soft-Stoppers

Programmname: SoftStopper_Gen

Computer: A500, A1000, A2000 mit Kickstart 1.2

Sprache: Amiga-Basic

Programmautor: Ralf Dietrich

```
1 pB0 REM Generiert lauffaehiges Programm
2 ag CLS
3 Br OPEN "df2:softstopper" FOR OUTPUT AS 1
4 BS READ anz
5 oa FOR i=1 TO anz
6 3n1 READ h$
7 yB2 wert1=ASC(LEFT$(h$,1))
8 bP IF wert1>64 THEN wert1=wert1-87 ELSE wert1=wert1-48
9 FI wert1=wert1*16
10 7c wert2=ASC(RIGHT$(h$,1))
11 wp IF wert2>64 THEN wert2=wert2-87 ELSE wert2=wert2-48
12 Pi wert=wert1+wert2
13 9G PRINT #1,CHR$(wert);
14 JOO NEXT
15 3n CLOSE 1
16 Ov END
17 yc Werte:
18 Kr DATA 188
19 ph DATA 00,00,03,f3,00,00,00,00,00,00
20 bk DATA 00,02,00,00,00,00,00,00,00,01
21 bk DATA 00,00,00,20,00,00,00,01,00,00
22 Yn DATA 03,e9,00,00,00,20,48,e7,ff,fe
23 Mx DATA 41,fa,00,1c,43,fa,00,72,24,7c
24 6t DATA 00,00,01,00,26,4a,34,d8,b3,c8
25 4N DATA 66,fa,4e,93,4c,df,7f,ff,4e,75
26 on DATA 41,fa,00,2e,20,b9,00,00,00,68
27 jP DATA 41,fa,00,0a,23,c8,00,00,00,68
28 nO DATA 4e,75,48,e7,ff,fe,10,39,00,bf
29 dz DATA ec,01,0c,00,00,79,66,04,61,00
30 IB DATA 00,0c,4c,df,7f,ff,4e,f9,ff,ff
31 XS DATA ff,ff,40,c0,46,fc,27,00,08,39
32 Ok DATA 00,06,00,bf,e0,01,66,f6,13,fc
33 1z DATA 00,78,00,bf,ec,01,08,39,00,06
34 id DATA 00,bf,e0,01,67,f6,46,c0,4e,75
35 me DATA 00,00,ff,ff,00,00,03,ec,00,00
36 D8 DATA 00,00,00,00,03,f2,00,00,03,eb
37 ja DATA 00,00,00,01,00,00,03,f2
(C) 1989 M&T
```

Listing 2. Kurz und bündig: das Stopper-Programm in DATA-Zeilen, einzugeben mit Amiga-Basic

Egal in welchem Programm Sie sich gerade befinden, »SniffIFF« speichert fast jeden Bildschirminhalt oder einen Ausschnitt davon. Ob Sie dieses Bild ausdrucken, mit einem Malprogramm nachbearbeiten oder einfach nur genauer betrachten wollen, SniffIFF macht's möglich. SniffIFF ist einfach zu installieren, und auch für die Aufrufe brauchen Sie sich nur wenig zu merken: Mit einer einfachen Tastenkombination <Amiga links-Shift links-s> speichern Sie den aktuellen Bildschirminhalt in einer »IFF«-Datei, mit <Amiga links-Shift links-b> einen Ausschnitt.

Sie starten das Programm mit »SniffIFF« vom CLI aus; darauf wird es installiert und bleibt solange resident, bis es mit <Amiga links-Shift links-q> wieder aus dem Speicher entfernt wird oder bis ein Reset ausgelöst wird. Sie brauchen SniffIFF nicht mit »Run« zu starten, denn es arbeitet automatisch im Hintergrund; Sie können nach der Installation normal weiterarbeiten.

Wenn Sie »SniffIFF« ohne Parameter aufrufen, werden alle Dateien unter dem Namen »picxx« in »df1:« gespeichert. Dabei ist »xx« eine aufsteigende Numerierung der Bilder. Wollen Sie die Bilder in einem anderen Laufwerk haben, dann sollten Sie bei der Installation das Verzeichnis als Parameter übergeben: »SniffIFF [Verzeichnis]«.

Standardmäßig werden IFF-Dateien komprimiert und auf

Der Bilderklau

Dieses Bild muß ich haben! Wie oft haben Sie sich schon gefragt, wie man an Grafiken professioneller Programme kommt? »SniffIFF« erledigt diese Aufgabe für Sie. Das Besondere an diesem Programm: Es belegt nur sehr wenig Speicher und läuft unauffällig im Hintergrund.

Wort-Grenze gespeichert. Falls Ihr Malprogramm eine derartige Datei nicht laden kann, verändern Sie mit den Parametern »-u« (für unkomprimiert) oder »-b« (für Speichern auf Byte-Grenze) diese Einstellungen. Beispielsweise sollten

Weiterbearbeiten mit DPaint

Sie die Option »-b« verwenden, wenn Sie mit »DPaint« arbeiten, da dieses Programm IFF-Dateien auf Byte-Grenze erwartet.

Bevor Sie SniffIFF starten können, geben Sie Listing 1 mit dem Checksummer ein.

Das Programm »SniffIFF__Gen« ist ein Basic-Programm, das Sie nur einmal starten. Bei diesem Durchlauf wird das ablauffähige Programm SniffIFF erzeugt und auf Diskette geschrieben.

Starten Sie nun Ihren »Grafikdieb« mit

SniffIFF [Verzeichnis]
[-u] [-b]

Hinterher können Sie eine beliebige Anwendung laden, von der die Grafik gespeichert werden soll. SniffIFF kann alle Bildschirme speichern außer denen, welche den »HAM«-Modus verwenden oder die Ta-

statur blockieren. Wenn Sie einen Bildschirmausschnitt speichern wollen (mit <Amiga links-Shift links-b>), können Sie mit der Maus einen Rahmen über den Bereich legen, den Sie speichern wollen.

Wenn Sie SniffIFF nicht mehr benötigen, entfernen Sie es mit der Tastenkombination <Amiga links-Shift links-q> aus dem Speicher.

Wir veröffentlichen SniffIFF als Data-Lader, damit auch Leser, die keinen Assembler haben, in den Genuß dieses Programms kommen.

Wer sich für die Programmierung interessiert, findet auf der Programmservice-Diskette ein Assembler-Source-File, das mit dem Public Domain-Assembler A68K von der Fish-Disk 110 assembliert werden kann. Sie benötigen dazu das Include-File »amiga.library« vom Lattice-C-Compiler und den Public Domain-Linker BLINK in der Version 7.1.

Für fortgeschrittene Assembler-Programmierer dürfte es kein Problem sein, dieses Programm beispielsweise auf den »DevPac« zu übertragen oder die eine oder andere Routine hinzuzufügen.

Übrigens läßt sich die Routine »ILBMSave« auch in eigenen Programmen verwenden, um IFF-Bilder zu speichern. Sie müssen als Parameter die Adresse des Viewport, die Startkoordinaten sowie Höhe und Breite angeben.

Sie können sich damit Ihre eigene »Grabbit«-Version bauen. Viel Spaß beim Tüfteln!

(so)

SniffIFF-Kurzreferenz

<Amiga links-Shift links-q>
<Amiga links-Shift links-s>
<Amiga links-Shift links-b>

Aufruf mit:
SniffIFF [Verz.][-u][-b]

beendet SniffIFF
speichert den Bildschirminhalt
speichert einen Bildschirmausschnitt

installiert SniffIFF

Tabelle 1. Der Aufruf und die Tastenkombinationen, mit denen Sie »SniffIFF« steuern

Programmname: SniffIFF__gen
Computer: A500, A1000, A2000 mit Kickstart 1.2
Sprache: Amiga-Basic

Programmautor: Werner Günther

```
1 Om0 REM Generiert lauffähiges Programm
2 ag CLS
3 AJ OPEN "df1:sniffiff/sniffiff.bas" FOR OUTPUT AS 1
4 BS READ anz
5 oa FOR i=1 TO anz
6 3n1 READ h$
7 yB2 wert1=ASC(LEFT$(h$,1))
8 bP IF wert1>64 THEN wert1=wert1-87 ELSE wert1=wert1-48
9 FI wert1=wert1*16
10 7c wert2=ASC(RIGHT$(h$,1))
11 wp IF wert2>64 THEN wert2=wert2-87 ELSE wert2=wert2-48
12 Pi wert=wert1+wert2
13 9G PRINT #1,CHR$(wert);
14 JOO NEXT
15 3n CLOSE 1
16 Ov END
17 yc Werte:
18 Gx DATA 5204
19 ph DATA 00,00,03,f3,00,00,00,00,00,00
20 7w DATA 00,0b,00,00,00,00,00,00,00,0a
21 sw DATA 00,00,00,63,00,00,01,58,00,00
```

```
22 d1 DATA 01,3f,00,00,00,29,00,00,00,40
23 5A DATA 00,00,00,40,00,00,00,05,00,00
24 Or DATA 00,37,00,00,00,09,00,00,00,1d
25 CD DATA 00,00,00,25,00,00,03,e9,00,00
26 CU DATA 00,63,42,81,34,3c,00,ff,43,f9
27 um DATA 00,00,04,f6,52,02,0c,30,00,20
28 XG DATA 20,00,67,f6,0c,30,00,2d,20,00
29 Jn DATA 66,20,52,02,0c,30,00,61,20,00
30 sK DATA 66,06,42,79,00,00,04,8e,0c,30
31 j4 DATA 00,75,20,00,66,d6,42,39,00,00
32 t1 DATA 05,5f,60,ce,0c,30,00,0a,20,00
33 St DATA 67,12,0c,30,00,20,00,67,be
34 rk DATA 12,f0,20,00,52,02,01,60,e6
35 Xt DATA 4a,81,66,02,72,08,23,c1,00,00
36 VG DATA 04,aa,30,3c,00,07,41,f9,00,00
37 i5 DATA 05,10,43,f9,00,00,04,f6,d3,e1
38 bw DATA 12,d8,51,c8,ff,fc,20,78,00,04
39 qu DATA 23,c8,00,00,04,92,20,68,01,7a
40 zA DATA 2f,08,43,f9,00,00,05,18,2c,78
41 ye DATA 00,04,4e,ae,fe,ec,23,c0,00,00
42 uK DATA 04,96,43,f9,00,00,05,2a,20,57
43 IV DATA 2c,78,00,04,4e,ae,fe,ec,23,c0
44 wp DATA 00,00,04,9a,43,f9,00,00,05,36
45 eT DATA 20,5f,2c,78,00,04,4e,ae,fe,ec
46 44 DATA 23,c0,00,00,04,a6,43,f9,00,00
47 nq DATA 05,54,2c,78,00,04,4e,ae,fe,da
48 hr DATA 4a,80,67,0e,24,3c,00,00,01,5a
49 hv DATA 26,3c,00,00,00,19,60,4c,2c,78
50 u2 DATA 00,04,4e,ae,ff,7c,41,fa,ff,0a
51 Os DATA 91,fc,00,00,00,04,26,28,00,00
```

Listing 1.
»SniffIFF__Gen«
geben Sie mit dem
Checksummer
auf Seite 159 ein.
Nach dem ersten
Programmlauf
wird das Programm
»SniffIFF« erzeugt.

52 at DATA 42,a8,00,00,23,c3,00,00,04,be
 53 w3 DATA 22,3c,00,00,05,54,74,32,28,3c
 54 v3 DATA 00,00,07,d0,2c,79,00,00,04,9a
 55 Hc DATA 4e,ae,ff,76,2c,78,00,00,04,ae
 56 1P DATA ff,76,24,3c,00,00,01,73,26,3c
 57 qH DATA 00,00,00,18,2c,79,00,00,04,9a
 58 BJ DATA 4e,ae,ff,c4,22,00,2c,79,00,00
 59 B1 DATA 04,9a,4e,ae,ff,d0,42,81,2c,79
 60 ez DATA 00,00,04,9a,4e,ee,ff,70,53,6e
 61 Mp DATA 69,70,49,66,66,20,61,6c,72,65
 62 yk DATA 61,64,79,20,6d,6f,75,6e,74,65
 63 1g DATA 64,2e,0a,53,6e,69,70,49,66,66
 64 4A DATA 20,56,31,2e,30,20,69,6e,73,74
 65 7n DATA 61,6c,6c,65,64,2e,0a,00,00,00
 66 WQ DATA 03,ec,00,00,00,02,00,00,00,00
 67 pU DATA 00,00,00,e0,00,00,01,2e,00,00
 68 AO DATA 00,13,00,00,00,01,00,00,00,08
 69 xF DATA 00,00,00,38,00,00,00,60,00,00
 70 GJ DATA 00,6a,00,00,00,70,00,00,00,82
 71 Mv DATA 00,00,00,8e,00,00,00,9c,00,00
 72 ta DATA 00,a2,00,00,00,b2,00,00,00,b8
 73 G2 DATA 00,00,00,c8,00,00,00,ce,00,00
 74 sM DATA 01,08,00,00,01,0e,00,00,01,1c
 75 pn DATA 00,00,01,3a,00,00,01,46,00,00
 76 5M DATA 01,52,00,00,00,01,00,00,00,02
 77 1w DATA 00,00,00,2a,00,00,00,00,00,00
 78 Lf DATA 03,f2,00,00,03,e9,00,00,01,58
 79 hJ DATA 48,78,00,00,48,78,00,00,4e,b9
 80 Ew DATA 00,00,00,00,4f,ef,00,08,4a,80
 81 GV DATA 67,00,02,d8,23,c0,00,00,04,9e
 82 ez DATA 2f,00,4e,b9,00,00,00,04,f,ef
 83 32 DATA 00,04,4a,80,67,00,02,c0,23,c0
 84 OI DATA 00,00,04,a2,22,40,41,fa,05,0d
 85 NH DATA 42,80,42,81,2c,7a,04,50,4e,ae
 86 gR DATA fe,44,4a,80,66,00,02,a2,23,c0
 87 Pz DATA 00,00,04,c6,22,7a,04,4c,23,7c
 88 1g DATA 00,00,04,de,00,28,33,7c,00,09
 89 sm DATA 00,1c,2c,7a,04,2a,4e,ae,fe,38
 90 37 DATA 4a,80,66,00,02,7c,23,c0,00,00
 91 YL DATA 04,c2,20,7a,04,16,22,68,01,14
 92 oo DATA 23,e9,00,00,04,b6,70,ff,2c,7a
 93 eH DATA 04,06,4e,ae,fe,b6,4a,80,6b,00
 94 gh DATA 02,58,42,81,01,c1,23,c1,00,00
 95 oR DATA 04,ba,43,fa,04,a3,2c,7a,03,ea
 96 Me DATA 4e,ae,fe,da,23,c0,00,00,04,ca
 97 hm DATA 20,3a,04,04,2c,7a,03,d8,4e,ae
 98 dn DATA fe,c2,10,3a,04,32,c0,00,00,10
 99 EQ DATA 67,00,02,24,2c,7a,03,c8,4e,ae
 100 oQ DATA fe,7a,28,7a,03,c0,28,6c,00,3c
 101 1G DATA 23,cc,00,00,04,ae,4b,ec,00,54
 102 xT DATA 26,6c,00,50,23,ed,00,00,04,b2
 103 tR DATA 10,3a,04,02,42,39,00,00,04,f4
 104 yJ DATA 0c,00,00,21,67,00,01,7e,13,fc
 105 bo DATA 00,01,00,00,04,f5,70,32,22,7a
 106 pa DATA 03,bc,2c,7a,03,80,4e,ae,fe,d4
 107 Va DATA 13,ed,00,19,00,00,05,5e,70,01
 108 hL DATA 22,4d,2c,7a,03,80,4e,ae,fe,aa
 109 uf DATA 13,ed,00,1c,00,00,05,5d,70,02
 110 sj DATA 22,4d,2c,7a,03,6c,4e,ae,fe,9e
 111 4i DATA 33,ed,00,20,00,00,04,da,3b,7c
 112 zu DATA 00,01,00,20,33,ed,00,22,00,00
 113 RP DATA 04,dc,3b,7c,ff,ff,00,22,08,39
 114 9q DATA 00,06,00,bf,e0,01,66,f6,61,00
 115 oH DATA 02,0a,4a,00,66,ae,33,ec,00,12
 116 R7 DATA 00,00,04,ee,33,ec,00,10,00,00
 117 mO DATA 04,d0,08,39,00,06,00,bf,e0,01
 118 VJ DATA 67,f6,20,3c,00,01,86,a0,51,c8
 119 GA DATA ff,fe,08,39,00,06,00,bf,e0,01
 120 4h DATA 67,e2,33,fa,03,30,00,00,04,d2
 121 XP DATA 33,fa,03,2a,00,00,04,d4,3a,2c
 122 sM DATA 00,12,3c,2c,00,10,33,c5,00,00
 123 ST DATA 04,d6,33,c6,00,00,04,d8,61,00
 124 Ym DATA 01,ce,10,39,00,bf,e0,01,02,00
 125 Jd DATA 00,40,13,c0,00,00,05,5e,67,14
 126 BC DATA 30,2c,00,12,b0,7a,02,fa,66,12
 127 D2 DATA 30,2c,00,10,b0,7a,02,f2,67,da
 128 nr DATA 61,00,01,86,4a,00,66,22,30,2c
 129 qY DATA 00,12,b0,7a,02,d6,6e,06,33,c0
 130 1V DATA 00,00,04,ee,30,2c,00,10,b0,7a
 131 Z2 DATA 02,c8,6e,06,33,c0,00,00,04,d0
 132 1Y DATA 3a,3a,02,c2,3c,3a,02,c0,61,00
 133 s6 DATA 01,74,4a,39,00,00,05,5c,66,00

134 ZQ DATA ff,76,42,80,10,3a,03,32,22,4d
 135 ep DATA 2c,7a,02,74,4e,ae,fe,aa,42,80
 136 WQ DATA 10,3a,03,21,22,4d,2c,7a,02,64
 137 Bx DATA 4e,ae,fe,9e,3b,7a,02,90,00,20
 138 PW DATA 3b,7a,02,8c,00,00,22,6c,00,50
 139 bF DATA 30,2b,00,08,d0,7a,02,70,32,2b
 140 8w DATA 00,0a,d2,7a,02,6a,34,3a,02,6c
 141 OM DATA 94,7a,02,60,52,42,36,3a,02,64
 142 30 DATA 96,7a,02,58,52,43,60,14,30,2b
 143 4K DATA 00,08,32,2b,00,0a,22,7a,02,26
 144 YB DATA 34,29,00,0c,36,29,00,0e,22,7a
 145 8X DATA 02,1a,43,e9,00,2c,41,fa,02,5a
 146 EG DATA 18,3a,02,bf,4a,39,00,00,04,f5
 147 IC DATA 67,1c,48,e7,f8,c0,70,14,22,7a
 148 4I DATA 02,18,2c,7a,01,dc,4e,ae,fe,44
 149 p5 DATA 42,39,00,00,04,f5,4c,d,f,03,1f
 150 UH DATA 4e,b9,00,00,00,1c,20,3a,01,dc
 151 B2 DATA 41,fa,02,24,12,30,00,01,52,01
 152 h0 DATA 0c,01,00,3a,66,06,72,2e,52,30
 153 1y DATA 00,00,11,81,00,01,60,00,fd,c8
 154 tU DATA 4a,b9,00,00,04,c2,66,1a,22,7a
 155 xu DATA 01,aa,23,7c,00,00,04,dc,00,28
 156 8r DATA 33,7c,00,0a,00,1c,2c,7a,01,88
 157 33 DATA 4e,ae,fe,38,4a,b9,00,00,04,c6
 158 aG DATA 66,0c,22,7a,01,88,2c,7a,01,74
 159 51 DATA 4e,ae,fe,3e,4a,b9,00,00,04,a2
 160 Ik DATA 67,10,2f,39,00,00,04,a2,4e,b9
 161 wP DATA 00,00,00,14,4f,ef,00,04,4a,b9
 162 Y7 DATA 00,00,04,9e,67,10,2f,39,00,00
 163 9c DATA 04,9e,4e,b9,00,00,00,96,4f,ef
 164 6n DATA 00,04,2c,7a,01,3c,4e,ae,ff,7c
 165 ax DATA 22,3a,01,60,2c,7a,01,38,4e,ae
 166 Hb DATA ff,64,42,81,2c,7a,01,2e,4e,ee
 167 dM DATA ff,70,70,01,32,2c,00,12,6b,14
 168 bv DATA b2,6c,00,0c,0e,0e,32,2c,00,10
 169 eu DATA 6b,08,b2,6c,00,0e,6e,02,42,00
 170 k8 DATA 4e,75,22,4d,26,6c,00,50,30,2b
 171 U7 DATA 00,08,d0,7a,01,36,32,2b,00,0a
 172 eF DATA d2,7a,01,30,2c,7a,00,fe,4e,ae
 173 Uv DATA ff,10,22,4d,26,6c,00,50,30,2b
 174 9q DATA 00,08,d0,45,32,2b,00,0a,d2,7a
 175 xX DATA 01,14,2c,7a,00,e2,4e,ae,ff,0a
 176 R8 DATA 22,4d,26,6c,00,50,30,2b,00,08
 177 Or DATA d0,45,32,2b,00,0a,d2,46,2c,7a
 178 pE DATA 00,c8,4e,ae,ff,0a,22,4d,26,6c
 179 uJ DATA 00,50,30,2b,00,08,d0,7a,00,e2
 180 U1 DATA 32,2b,00,0a,d2,46,2c,7a,00,ae
 181 xe DATA 4e,ae,ff,0a,22,4d,26,6c,00,50
 182 cC DATA 30,2b,00,08,d0,7a,00,c6,32,2b
 183 d0 DATA 00,0a,d2,7a,00,c0,2c,7a,00,8e
 184 AN DATA 4e,ee,ff,0a,10,28,00,04,0c,00
 185 hI DATA 00,01,66,42,30,28,00,08,02,40
 186 mT DATA 00,41,0c,40,00,41,66,52,30,28
 187 68 DATA 00,06,0c,00,00,21,67,0c,0e,00
 188 TU DATA 00,35,67,06,0c,00,00,10,66,38
 189 31 DATA 13,c0,00,00,04,f4,60,2e,2f,08
 190 ER DATA 22,7a,00,5e,20,3a,00,5e,2e,7a
 191 RI DATA 00,32,4e,ae,fe,bc,20,5f,60,16
 192 W0 DATA 0c,00,00,02,66,14,4a,39,00,00
 193 Wn DATA 04,f5,67,0c,0c,68,00,68,00,06
 194 bu DATA 66,04,42,28,00,00,04,20,8e,75
 195 2j DATA 4a,39,00,00,04,f4,66,c4,60,f2
 196 AQ DATA 00,00,00,00,00,00,00,00,00,00
 197 BR DATA 00,00,00,00,00,00,00,00,00,00
 198 i6 DATA 00,00,00,00,00,00,00,00,00,00
 199 DT DATA 00,00,00,00,00,00,00,00,00,00
 200 QT DATA 00,00,00,00,00,00,00,00,ff,ff
 201 RR DATA ff,ff,ff,ff,ff,ff,ff,ff,ff,ff
 202 GW DATA 00,00,00,00,00,00,00,00,00,00
 203 HX DATA 00,00,00,00,00,00,00,00,00,00
 204 IZ DATA 00,00,00,00,02,7f,00,00,00,00
 205 A2 DATA 00,00,00,00,00,00,04,1e,00,00
 206 FX DATA 73,61,76,65,69,66,66,2e,20,20
 207 dT DATA 20,20,20,20,20,20,20,20,20,20
 208 Tp DATA 20,20,20,20,20,20,30,30,2e,69
 209 Uk DATA 6c,62,6d,00,69,6e,74,75,69,74
 210 qg DATA 69,6f,6e,2e,6c,69,62,72,61,72
 211 or DATA 79,00,64,6f,73,2e,6c,69,62,72
 212 Ua DATA 61,72,79,00,67,72,61,70,68,69
 213 oJ DATA 63,73,2e,6c,69,62,72,61,72,79
 214 zL DATA 00,69,6e,70,75,74,2e,64,65,76
 215 3Q DATA 69,63,65,00,53,6e,69,49,66

216 eq DATA 66,00,00,00,00,01,00,00,03,ec
 217 zF DATA 00,00,00,27,00,00,00,01,00,00
 218 r8 DATA 00,1a,00,00,00,32,00,00,00,50
 219 KG DATA 00,00,00,5a,00,00,00,76,00,00
 220 Fb DATA 00,84,00,00,00,00,09,00,00,b0
 221 Ab DATA 00,00,00,de,00,00,00,ec,00,00
 222 6c DATA 00,f6,00,00,01,06,00,00,01,1c
 223 Ve DATA 00,00,01,30,00,00,01,44,00,00
 224 6P DATA 01,52,00,00,01,72,00,00,01,7a
 225 QO DATA 00,00,01,a0,00,00,01,a8,00,00
 226 rJ DATA 01,b6,00,00,01,bc,00,00,01,d0
 227 He DATA 00,00,01,fe,00,00,02,0e,00,00
 228 RL DATA 02,20,00,00,02,a4,00,00,02,be
 229 9W DATA 00,00,02,f0,00,00,02,fe,00,00
 230 EX DATA 03,12,00,00,03,26,00,00,03,2e
 231 nj DATA 00,00,03,3e,00,00,03,46,00,00
 232 dn DATA 04,4e,00,00,04,72,00,00,04,8a
 233 JA DATA 00,00,04,f0,00,00,00,01,00,00
 234 KT DATA 00,00,00,02,c8,00,00,00,02,02
 235 nr DATA 00,00,00,07,00,00,03,4c,00,00
 236 1Y DATA 00,0a,00,00,00,02,00,00,00,08
 237 fu DATA 00,00,03,34,00,00,00,22,00,00
 238 AQ DATA 00,00,00,00,03,f2,00,00,03,e9
 239 XI DATA 00,00,01,3f,48,e7,7f,fe,4c,ef
 240 1K DATA 03,00,00,3c,4c,ef,00,1f,00,44
 241 QJ DATA 4e,b9,00,00,00,0c,4c,d,f,7f,fe
 242 Oc DATA 4e,75,23,c8,00,00,04,3e,23,c9
 243 mJ DATA 00,00,04,7e,33,c0,00,00,04,86
 244 Ud DATA 33,c1,00,00,04,88,33,c2,00,00
 245 21 DATA 00,14,33,c2,00,00,00,24,33,c3
 246 6U DATA 00,00,00,16,33,c3,00,00,00,26
 247 mq DATA 13,c4,00,00,00,1e,42,b9,00,00
 248 1R DATA 04,82,20,7a,04,24,33,e8,00,20
 249 e5 DATA 00,00,00,32,22,68,00,04,30,29
 250 ZY DATA 00,02,48,c0,53,40,45,f9,00,00
 251 cD DATA 00,3e,22,69,00,04,42,81,32,19
 252 gm DATA 74,02,26,01,e8,4b,02,03,00,f0
 253 JE DATA 14,c3,e9,49,51,ca,ff,f2,51,c8
 254 xq DATA ff,ea,20,68,00,24,20,68,00,04
 255 af DATA 42,80,42,81,32,28,00,00,23,c1
 256 Hd DATA 00,00,04,66,30,39,00,00,00,14
 257 BE DATA 4a,79,00,00,04,8e,66,06,5e,40
 258 C7 DATA e6,48,60,08,06,40,00,0f,e8,48
 259 dP DATA e3,48,23,c0,00,00,04,6a,30,3a
 260 1F DATA 03,b8,80,fc,00,08,48,40,33,c0
 261 Vz DATA 00,00,04,8c,42,82,34,3a,03,a8
 262 PS DATA 04,c1,42,80,30,3a,03,9e,e8,48
 263 PD DATA d0,42,42,82,14,28,00,05,13,c2
 264 Jc DATA 00,00,00,1c,53,02,43,fa,03,44
 265 3N DATA 45,e8,00,f8,22,1a,d2,80,22,c1
 266 Om DATA 51,ca,ff,f8,42,a9,00,00,2c,79
 267 gc DATA 00,00,04,92,4e,ae,ff,88,2c,79
 268 ob DATA 00,00,04,92,4e,ae,ff,7c,33,fe
 269 uR DATA 40,00,00,d,f,0,9a,42,b9,00,00
 270 wA DATA 00,a0,42,79,00,00,04,8a,42,b9
 271 am DATA 00,00,04,6e,45,fa,03,00,22,6a
 272 UE DATA 00,00,42,81,42,82,47,f9,00,00
 273 PB DATA 00,00,10,19,34,03,36,67,0a
 274 Pe DATA e1,48,10,29,00,00,e5,68,e0,48
 275 nY DATA 17,80,10,00,52,41,b2,ba,02,fe
 276 92 DATA 66,e2,22,4b,41,f9,00,00,00,00
 277 OL DATA 4a,39,00,00,00,1e,66,14,20,3a
 278 qL DATA 02,e8,22,00,11,b1,10,00,10,00
 279 re DATA 51,c9,ff,f8,60,00,00,00,70,01
 280 dE DATA 42,39,00,00,04,cc,42,81,42,85
 281 kb DATA 42,84,14,31,10,00,4a,01,67,04
 282 Ne DATA b6,02,67,1a,16,02,52,04,11,82
 283 mE DATA 00,00,52,00,52,01,b2,ba,02,ae
 284 x7 DATA 66,e2,53,04,11,84,50,00,60,5a
 285 7F DATA 55,04,6a,04,10,05,60,06,11,84
 286 bz DATA 50,00,53,00,42,04,14,31,10,00
 287 zJ DATA b6,02,66,12,52,04,52,01,b2,ba
 288 sR DATA 02,84,66,ee,13,fc,00,01,00,00
 289 nY DATA 04,ce,44,04,11,84,00,00,11,83
 290 FJ DATA 00,01,42,04,54,00,1a,00,52,00
 291 dy DATA b2,ba,02,64,67,02,60,a2,53,00
 292 az DATA 4a,39,00,00,04,ce,66,0c,11,be
 293 9R DATA 00,00,50,00,11,82,00,01,54,00
 294 CD DATA 24,3a,02,42,d5,aa,00,00,45,ea
 295 2V DATA 00,04,4a,aa,00,00,66,0a,45,fa
 296 De DATA 02,0c,52,79,00,00,04,8a,42,82
 297 Ag DATA 14,00,5a,82,26,00,48,e7,7f,fe


```

298 y5 DATA 20,02,42,81,2c,79,00,00,04,92
299 X2 DATA 4e,ae,ff,3a,4c,df,7f,fe,4a,80
300 i1 DATA 67,00,01,14,4a,b9,00,00,04,6e
301 fp DATA 66,08,23,c0,00,00,04,6e,60,08
302 Ae DATA 20,7a,01,fe,21,40,00,00,23,c0
303 YP DATA 00,00,04,72,20,40,42,a8,00,00
304 rm DATA 11,43,00,04,d7,b9,00,00,00,a0
305 zs DATA 41,e8,00,05,43,f9,00,00,00,00
306 C4 DATA 53,03,10,d9,51,eb,ff,fc,34,39
307 KA DATA 00,00,04,8a,b4,79,00,00,00,16
308 xw DATA 66,00,fe,94,33,fc,c0,00,00,df
309 iJ DATA f0,9a,22,3a,01,82,24,3c,00,00
310 Qd DATA 03,ee,2c,79,00,00,04,9a,4e,ae
311 fn DATA ff,e2,23,c0,00,00,04,76,4a,80
312 gm DATA 67,00,00,e0,22,39,00,00,00,a0
313 pt DATA 06,81,00,00,00,00,9c,23,c1,00,00
314 LG DATA 00,04,22,00,26,3c,00,00,00,a4
315 Xe DATA 24,3c,00,00,00,00,2c,79,00,00
316 4k DATA 04,9a,4e,ae,ff,d0,20,7a,01,68
317 Tk DATA 23,e8,00,00,00,00,04,7a,42,83
318 Uh DATA 16,28,00,04,43,e8,00,05,24,09
319 eG DATA 22,3a,01,58,2f,03,2f,08,2c,79
320 Wf DATA 00,00,04,9a,4e,ae,ff,d0,20,5f
321 lo DATA 26,1f,b6,80,66,00,00,c6,61,00
322 bi DATA 00,d4,4a,b9,00,00,04,7a,67,08
323 EU DATA 20,79,00,00,04,7a,60,bc,22,3a
324 5L DATA 01,28,2c,79,00,00,04,9a,4e,ae
325 Zp DATA ff,dc,2c,79,00,00,04,92,4e,ae
326 b2 DATA ff,76,2c,79,00,00,04,92,4e,ae
327 kp DATA ff,82,20,3a,01,12,4e,75,33,fc
328 sH DATA c0,00,00,df,f0,9a,2c,79,00,00
329 r1 DATA 04,92,4e,ae,ff,76,2c,79,00,00
330 lT DATA 04,92,4e,ae,ff,82,4a,b9,00,00
331 qJ DATA 04,6e,67,10,20,7a,00,d4,23,e8
332 ac DATA 00,00,00,00,04,7a,61,00,00,7e
333 jH DATA 43,fa,00,e6,23,fc,00,00,00,01
334 9v DATA 00,00,04,82,60,20,00,00,b4
335 A7 DATA 23,e8,00,00,00,00,04,7a,61,00
336 Py DATA 00,5e,22,7c,00,00,04,a4,23,fc
337 Fp DATA 00,00,00,02,00,00,04,82,91,c8
338 fD DATA 45,fa,00,dc,26,4a,42,80,42,81
339 cw DATA 24,3c,00,00,00,fa,76,3c,2c,79
340 wM DATA 00,00,04,96,4e,ae,fe,a4,20,3a
341 Bo DATA 00,8a,4e,75,61,00,00,26,23,fc
342 6v DATA 00,00,00,03,00,00,04,82,60,00
343 9V DATA ff,40,42,80,10,28,00,04,5a,00
344 eE DATA 22,48,2c,79,00,00,04,92,4e,ae
345 IP DATA ff,2e,4e,75,61,e8,4a,b9,00,00
346 O8 DATA 04,7a,66,02,4e,75,20,7a,00,48
347 E7 DATA 23,e8,00,00,00,00,04,7a,60,e6
348 cs DATA 00,00,00,00,00,00,00,00,00,00
349 dt DATA 00,00,00,00,00,00,00,00,00,00
350 eu DATA 00,00,00,00,00,00,00,00,00,00
351 Fv DATA 00,00,00,00,00,00,00,00,00,00
352 gw DATA 00,00,00,00,00,00,00,00,00,00
353 hx DATA 00,00,00,00,00,00,00,00,00,00
354 iy DATA 00,00,00,00,00,00,00,00,00,00
355 jz DATA 00,00,00,00,00,00,00,00,00,00
356 Rb DATA 00,01,00,01,00,00,00,28,00,0a
357 yd DATA 00,00,00,00,00,00,04,ed,00,00
358 EA DATA 00,00,00,01,00,00,00,1e,00,0a
359 Bt DATA 00,00,00,00,00,00,04,df,00,00
360 Po DATA 00,00,00,01,00,00,00,05,00,03
361 1w DATA 00,00,00,00,00,00,04,f3,00,00
362 uY DATA 00,00,00,4e,6f,74,20,65,6e,6f
363 iZ DATA 75,67,68,20,6d,65,6d,6f,72,79
364 uw DATA 00,55,6e,61,62,6c,65,20,74,6f
365 Qg DATA 20,6f,70,65,6e,20,66,69,6c,65
366 dC DATA 00,43,41,4e,43,45,4c,00,4e,71
367 Fh DATA 00,00,03,ec,00,00,00,0d,00,00
368 OI DATA 00,01,00,00,03,ee,00,00,02,c6
369 jI DATA 00,00,02,fc,00,00,03,26,00,00
370 Od DATA 03,52,00,00,01,14,00,00,01,1e
371 Q6 DATA 00,00,02,50,00,00,03,5c,00,00
372 iQ DATA 03,66,00,00,03,7e,00,00,03,88
373 JS DATA 00,00,04,1a,00,00,00,24,00,00
374 hk DATA 00,02,00,00,00,12,00,00,00,1e
375 de DATA 00,00,00,24,00,00,00,2a,00,00
376 Je DATA 00,30,00,00,00,54,00,00,00,a6
377 yz DATA 00,00,00,b2,00,00,00,c8,00,00
378 sV DATA 00,d8,00,00,01,36,00,00,01,3c
379 Or DATA 00,00,01,98,00,00,01,ee,00,00

380 i5 DATA 02,10,00,00,02,3a,00,00,02,64
381 Kr DATA 00,00,02,6c,00,00,02,7c,00,00
382 eD DATA 02,a4,00,00,02,d0,00,00,03,0c
383 FB DATA 00,00,03,3e,00,00,03,46,00,00
384 Zc DATA 03,92,00,00,03,a0,00,00,03,b2
385 Gh DATA 00,00,03,c0,00,00,03,ca,00,00
386 Tk DATA 03,d4,00,00,04,06,00,00,04,28
387 A8 DATA 00,00,04,38,00,00,04,9c,00,00
388 io DATA 04,b0,00,00,04,c4,00,00,00,10
389 uM DATA 00,00,00,03,00,00,00,36,00,00
390 bu DATA 00,3c,00,00,00,42,00,00,00,48
391 he DATA 00,00,00,4e,00,00,00,60,00,00
392 6v DATA 00,72,00,00,00,ac,00,00,00,f6
393 ID DATA 00,00,01,30,00,00,01,7a,00,00
394 g7 DATA 02,8c,00,00,02,aa,00,00,02,dc
395 5v DATA 00,00,02,e8,00,00,02,f6,00,00
396 ay DATA 00,02,00,00,00,04,00,00,01,74
397 FU DATA 00,00,02,96,00,00,00,01,00,00
398 gY DATA 00,05,00,00,01,4e,00,00,00,00
399 wF DATA 00,00,03,f2,00,00,03,ea,00,00
400 Kp DATA 00,29,46,f4,52,4d,00,00,00,00
401 It DATA 49,4c,42,4d,42,4d,48,44,00,00
402 xy DATA 00,14,00,00,00,00,00,00,00,00
403 TN DATA 00,00,00,00,00,00,0a,0b,00,00
404 Zq DATA 00,00,43,41,4d,47,00,00,00,04
405 Lh DATA 00,00,00,00,43,4d,41,50,00,00
406 20 DATA 00,60,00,00,00,00,00,00,00,00
407 Zp DATA 00,00,00,00,00,00,00,00,00,00
408 aq DATA 00,00,00,00,00,00,00,00,00,00
409 br DATA 00,00,00,00,00,00,00,00,00,00
410 es DATA 00,00,00,00,00,00,00,00,00,00
411 dt DATA 00,00,00,00,00,00,00,00,00,00
412 eu DATA 00,00,00,00,00,00,00,00,00,00
413 fv DATA 00,00,00,00,00,00,00,00,00,00
414 gw DATA 00,00,00,00,00,00,00,00,00,00
415 lj DATA 00,00,00,00,00,00,00,00,42,4f
416 jt DATA 44,59,00,00,00,00,00,00,03,f2
417 re DATA 00,00,03,eb,00,00,00,40,00,00
418 at DATA 03,f2,00,00,03,eb,00,00,00,40
419 gL DATA 00,00,03,f2,00,00,03,e9,00,00
420 G1 DATA 00,05,20,6f,00,04,20,88,58,90
421 4v DATA 42,a8,00,04,21,48,00,08,4e,75
422 xa DATA 00,00,00,00,03,f0,00,00,00,02
423 1w DATA 5f,4e,65,77,4c,69,73,74,00,00
424 an DATA 00,00,00,00,00,00,00,00,03,f2
425 hj DATA 00,00,03,e9,00,00,00,3f,48,e7
426 ud DATA 3c,20,26,2f,00,18,14,2f,00,1f
427 yV DATA 7a,ff,2f,05,4e,b9,00,00,00,44
428 OI DATA 12,00,70,00,10,01,28,00,72,ff
429 db DATA b2,80,58,8f,66,04,70,00,60,66
430 7v DATA 2f,3c,00,01,00,01,48,78,00,22
431 li DATA 4e,b9,00,00,00,00,24,40,2a,0a
432 W7 DATA 50,8f,66,0e,2f,04,4e,b9,00,00
433 sF DATA 00,58,70,00,58,8f,60,40,25,43
434 WK DATA 00,0a,15,42,00,09,15,7c,00,04
435 ps DATA 00,08,42,2a,00,0e,15,44,00,0f
436 Oz DATA 42,a7,4e,b9,00,00,00,30,25,40
437 PF DATA 00,10,4a,83,58,8f,67,0a,2f,0a
438 xv DATA 4e,b9,00,00,00,6c,60,0a,48,6a
439 uZ DATA 00,14,4e,b9,00,00,00,58,8f
440 jt DATA 20,0a,4c,df,04,3c,4e,75,2f,0a
441 QG DATA 24,6f,00,08,4a,aa,00,0a,67,0a
442 i4 DATA 2f,0a,4e,b9,00,00,00,80,58,8f
443 Y7 DATA 15,7c,00,ff,00,08,70,ff,25,40
444 tT DATA 00,14,70,00,10,2a,00,0f,2f,00
445 Z2 DATA 4e,b9,00,00,00,58,48,78,00,22
446 p9 DATA 2f,0a,4e,b9,00,00,00,18,4f,ef
447 JS DATA 00,0c,24,5f,4e,75,00,00,00,00
448 76 DATA 03,ec,00,00,00,01,00,00,00,06
449 nn DATA 00,00,00,88,00,00,00,08,00,00
450 cu DATA 00,0a,00,00,00,36,00,00,00,12
451 2Y DATA 00,00,00,6a,00,00,00,c2,00,00
452 XI DATA 00,46,00,00,00,7c,00,00,00,a6
453 xm DATA 00,00,00,ce,00,00,00,00,00,00
454 Cp DATA 03,f0,00,00,00,01,2e,4c,38,00
455 SO DATA 00,00,02,2a,00,00,00,01,2e,4c
456 2Y DATA 37,00,00,00,00,50,00,00,00,01
457 JX DATA 2e,4c,36,00,00,00,00,82,00,00
458 tk DATA 00,01,2e,4c,31,00,00,00,90
459 BK DATA 00,00,00,01,2e,4c,31,34,00,00
460 oz DATA 00,ac,00,00,00,01,2e,4c,31,32
461 r9 DATA 00,00,00,8c,00,00,00,03,5f,44

462 fK DATA 65,6c,65,74,65,50,6f,72,74,00
463 Sw DATA 00,00,00,96,00,00,00,03,5f,43
464 sP DATA 72,65,61,74,65,50,6f,72,74,00
465 V1 DATA 00,00,00,00,00,00,00,00,00,00
466 KZ DATA 03,f2,00,00,03,e9,00,00,00,09
467 iq DATA 20,2f,00,04,48,78,00,30,2f,00
468 Ne DATA 4e,b9,00,00,00,00,50,8f,4e,75
469 5U DATA 20,2f,00,04,2f,00,4e,b9,00,00
470 Ob DATA 00,42,58,8f,4e,75,00,00,03,ee
471 Fg DATA 00,00,00,02,00,00,00,09,00,00
472 3r DATA 00,0c,00,00,00,1c,00,00,00,00
473 3b DATA 00,00,03,f0,00,00,00,03,5f,44
474 oz DATA 65,6c,65,74,65,53,74,64,49,4f
475 om DATA 00,00,00,14,00,00,00,03,5f,43
476 14 DATA 72,65,61,74,65,53,74,64,49,4f
477 hx DATA 00,00,00,00,00,00,00,00,00,00
478 30 DATA 03,f2,00,00,03,e9,00,00,00,1d
479 up DATA 48,e7,38,00,24,2f,00,10,26,2f
480 Ov DATA 00,14,4a,82,66,04,70,00,60,28
481 KQ DATA 2f,3c,00,01,00,01,2f,03,4e,b9
482 qW DATA 00,00,00,00,20,40,28,08,50,8f
483 p4 DATA 66,02,60,e4,11,7c,00,05,00,08
484 Qa DATA 31,43,00,12,21,42,00,0e,20,08
485 qH DATA 4c,df,00,1c,4e,75,20,6f,00,04
486 iX DATA 20,08,66,02,60,24,11,7c,00,ff
487 6p DATA 00,08,70,ff,21,40,00,14,70,ff
488 CO DATA 21,40,00,18,70,00,30,28,00,12
489 hu DATA 2f,00,2f,08,4e,b9,00,00,00,18
490 1H DATA 50,8f,4e,75,00,00,00,00,03,ee
491 9E DATA 00,00,00,02,00,00,00,0a,00,00
492 iS DATA 00,1e,00,00,00,6a,00,00,00,00
493 iA DATA 00,00,03,f0,00,00,00,01,2e,4c
494 c6 DATA 39,00,00,00,00,10,00,00,00,01
495 em DATA 2e,4c,35,00,00,00,00,14,00,00
496 P1 DATA 00,01,2e,4c,34,00,00,00,00,2c
497 OQ DATA 00,00,00,01,2e,4c,31,00,00,00
498 19 DATA 00,3c,00,00,00,01,2e,4c,31,32
499 Qs DATA 00,00,00,4c,00,00,00,01,2e,4c
500 vS DATA 31,30,00,00,00,70,00,00,00,03
501 oZ DATA 5f,44,65,6c,65,74,65,45,78,74
502 gs DATA 49,4f,00,00,00,00,42,00,00,03
503 nm DATA 5f,43,72,65,61,74,65,45,78,74
504 FI DATA 49,4f,00,00,00,00,00,00,00,00
505 4J DATA 00,00,03,f2,00,00,00,03,e9,00,00
506 sx DATA 00,25,2f,0e,2c,79,00,00,04,92
507 7K DATA 4c,ef,00,03,00,08,4e,ae,ff,3a
508 dF DATA 2c,5f,4e,75,00,00,2f,0e,2c,79
509 hP DATA 00,00,04,92,22,6f,00,08,20,2f
510 TA DATA 00,0c,4e,ae,ff,2e,2c,5f,4e,75
511 MR DATA 2f,0e,2c,79,00,00,04,92,22,6f
512 gs DATA 00,08,4e,ae,fe,da,2c,5f,4e,75
513 43 DATA 2f,0e,2c,79,00,00,04,92,20,2f
514 Z2 DATA 00,08,4e,ae,fe,b6,2c,5f,4e,75
515 65 DATA 2f,0e,2c,79,00,00,04,92,20,2f
516 Jg DATA 00,08,4e,ae,fe,b0,2c,5f,4e,75
517 SX DATA 2f,0e,2c,79,00,00,04,92,22,6f
518 a9 DATA 00,08,4e,ae,fe,9e,2c,5f,4e,75
519 UZ DATA 2f,0e,2c,79,00,00,04,92,22,6f
520 RH DATA 00,08,4e,ae,fe,98,2c,5f,4e,75
521 pe DATA 00,00,03,ec,00,00,00,07,00,00
522 Qv DATA 00,01,00,00,00,84,00,00,00,70
523 HE DATA 00,00,00,5c,00,00,00,48,00,00
524 ur DATA 00,34,00,00,00,1c,00,00,00,04
525 NJ DATA 00,00,00,00,00,00,03,f0,00,00
526 9r DATA 00,02,5f,52,65,6d,50,6f,72,74
527 yN DATA 00,00,00,80,00,00,00,02,5f,41
528 sw DATA 64,64,50,6f,72,74,00,00,00,6c
529 Po DATA 00,00,00,03,5f,46,72,65,65,53
530 8z DATA 69,67,6e,61,6c,00,00,00,00,58
531 TA DATA 00,00,00,03,5f,41,6c,6c,6f,63
532 zJ DATA 53,69,67,6e,61,6c,00,00,00,44
533 Dv DATA 00,00,00,03,5f,46,69,6e,64,54
534 Qf DATA 61,73,6b,00,00,00,00,00,00,30
535 MW DATA 00,00,00,02,5f,46,72,65,65,4d
536 Gd DATA 65,6d,00,00,00,18,00,00,00,03
537 yz DATA 5f,41,6c,6c,6f,63,4d,65,6d,00
538 gw DATA 00,00,00,00,00,00,00,00,00,00
539 5E DATA 00,00,03,f2
(C) 1988 M&T

```

Listing 1. (Schluß)

Öl, Geld und Bomben

Spielen Sie Manager

Es ist kaum zu glauben, mein 520 000-Dollar-Tanker liegt mit Motorschaden im Trockendock, der Bohrturm geht in Flammen auf, und dann auch noch 120 000 Dollar Strafe wegen dieser lächerlichen Beamtenbestechung, das treibt mich in den Ruin. Gott sei Dank liegt das Sümmchen aus meiner letzten Börsenmanipulation sicher auf meinem Schweizer Nummernkonto. So mörderisch ist das Leben eines Managers in »Oel-Fieber«. Denn wer hier nicht knallhart agiert, wird von seinem Gegner ausgetrickst und endet am Betelstab. Damit es nicht so weit

Wollten Sie nicht schon immer Boß eines Öl-Imperiums sein? An der Börse spekulieren? Oder à la J. R. Bohrtürme sprengen und der Konkurrenz zeigen, was eine Harke ist? Dann ist »Oel-Fieber« das richtige Spiel für Sie. Werden Sie zum Millionär.

kommt, geben wir Ihnen fast gratis einen Manager-Kursus.

Bevor Sie anfangen, das Listing abzutippen, erstellen Sie eine Diskette, auf welche man später das Programm abspeichert. Machen Sie dafür eine Kopie Ihrer Original-Workbench

und löschen Sie auf Ihrer Kopie (niemals auf dem Original!) die Schubladen Utilities, Expansion, Empty und Prefs mit Hilfe der »Discard«-Funktion aus dem Project-Menü. Dann kopieren Sie noch »AmigaBasic« von Ihrer Extras-Diskette auf

die soeben erstellte Kopie und benennen diese in »Oel-Fieber« um. Wie Sie dabei vorgehen müssen, erfahren Sie in unserem Kurs auf Seite 28. Aber nun zur harten Arbeit, tippen Sie das Listing mit Hilfe des Checksummer V1.1 ab (wenn Sie nicht wissen wie das geht, lesen Sie bitte den Artikel »Big Brother is Checking you« auf Seite 126). Bei der Frage des Checksummers nach dem Dateinamen geben Sie bitte »Oel-Fieber:Oel-Fieber« ein. Darauf folgt links oben im Bildschirm eine Aufforderung, die Diskette mit dem Namen »Oel-Fieber« einzulegen. Tun Sie das, und wenn Sie

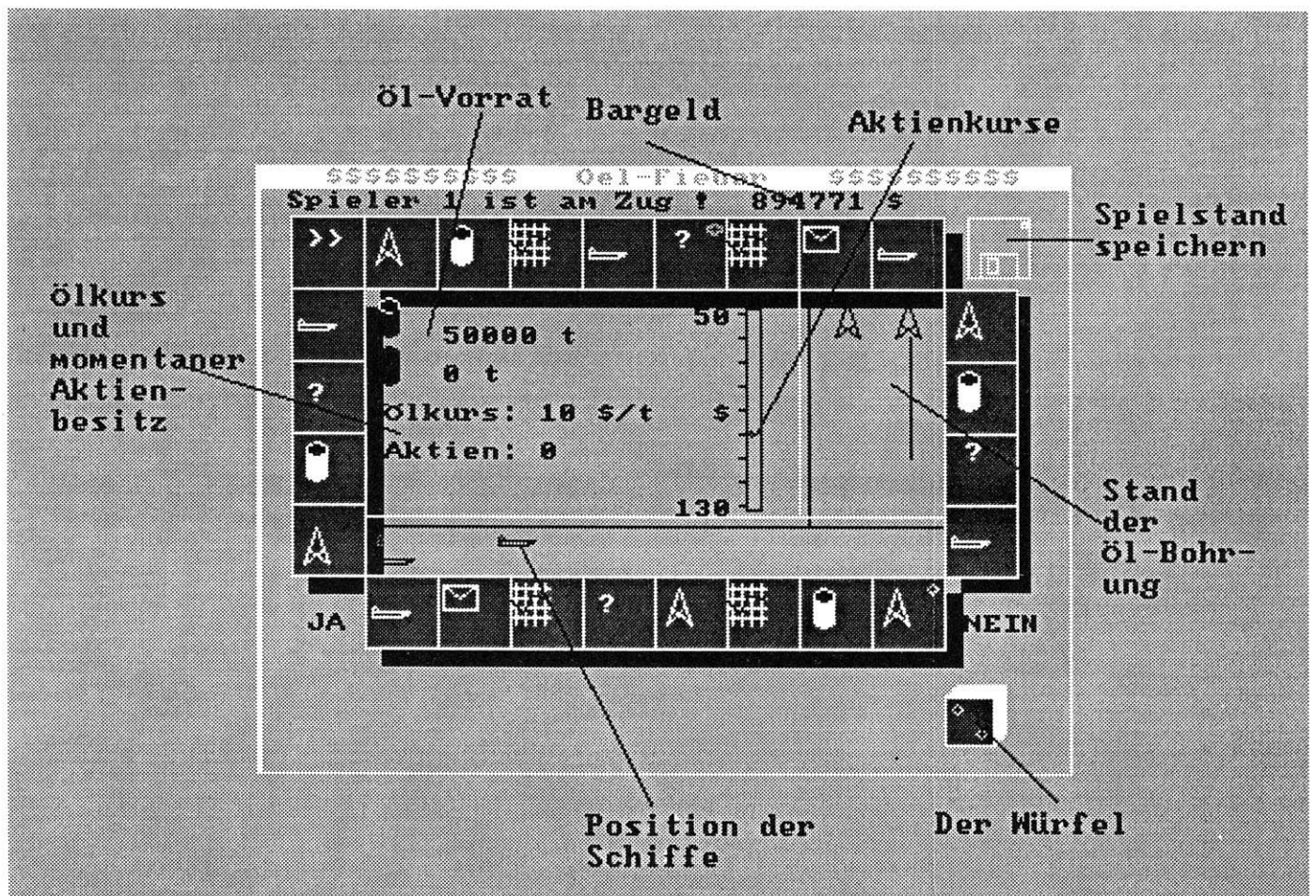


Bild 1. Die Spieloberfläche von »Oel-Fieber« auf einen Blick. Die erläuterten Felder sollten Sie stets unter Kontrolle behalten.

Felder-Übersicht

Startfeld:	Jeder Spieler erhält 30 000 \$ Gehalt beim Überqueren. Sollten Sie genau auf dieses Feld treffen, gibt es 80 000 \$ Erfolgsprämie.
Schiff:	Beim ersten Betreten des Feldes wird Ihnen ein Schiff zum Kauf angeboten. Sie müssen mit einem Klick auf »Ja« oder »Nein« eine Entscheidung treffen. Besitzen Sie bereits ein Schiff und es befinden sich mindestens 50 000 Tonnen Öl in Ihrem Bunker (siehe Bild 1 »Öl-Vorrat«), so fährt Ihr Schiff dem Zielhafen entgegen (Bild 1 »Position der Schiffe«).
Bohrturm:	Wenn Sie auf dieses Quadrat kommen, können Sie einen Bohrturm kaufen (wählen Sie »Ja« oder »Nein«). Nachdem Sie einen solchen Turm gekauft haben, wird bei jedem weiteren Besuch tiefer gebohrt (siehe Bild 1 »Stand der Öl-Bohrung«). Bis Sie auf Öl gestoßen sind. Ein Bohrturm kann maximal 100 000 t Öl fördern, dann ist das Öl-Feld leer und es muß ein neuer Bohrturm erstanden werden.
Öltonne:	Wenn Ihr Schiff im Zielhafen vor Anker liegt, können Sie die 50 000 t Öl an Bord zum Tageskurs (siehe Bild 1 »Ölkurs«) verkaufen.
Ereignisfeld:	Bei diesem Feld können sich die merkwürdigsten Dinge ereignen. Lassen Sie sich überraschen.
Sabotagefeld:	Hier kommt Ihre rabenschwarze Seele zum Vorschein. Sie können hier das Schiff und den Bohrturm des Gegners sabotieren. Folgende Möglichkeiten stehen zur Wahl:
Sabotage-Bohrturm:	Feuer: Der Mitspieler kann, bis er das Startfeld überschritten hat, nicht bohren und die Löscharbeiten kosten ihn 80 000 \$. - Explosion: Es kann zwei Runden lang nicht gebohrt werden und die Ersatzteile kosten 120 000 \$. - Nachschub: Der Gegner kann eine Runde lang nicht bohren und der beschleunigte Nachschubtransport kostet 20 000 \$.
Sabotage-Schiff:	Leck: Das Schiff muß zwei Runden die Motoren abschalten und die Reparatur kostet 200 000 \$. - Motor: Der Motor wird ausgebaut, dauert eine Runde, und die Kosten für die Instandsetzung belaufen sich auf 100 000 \$. - Verpflegung: Die Nahrungsmittel auf dem Schiff sind verdorben. Die Mannschaft schiebt Bummeldienst, kostet eine Runde an Zeit, und die Schadensersatzklagen der Seeleute verschlingen 20 000 \$.
Börsenfeld:	Hier können Sie Aktien zum Tageskurs (siehe Bild 1 »Aktienkurs«) erwerben oder abstoßen. Sollte die Börse schon geschlossen haben, Pech, so müssen Sie Ihre Aktivitäten auf das nächste Mal verschieben.

Tabelle 1. Schnellübersicht

Ihre vorbereitete Diskette richtig benannt haben, kopiert der Amiga das Programm automatisch auf diese Diskette. Jetzt ist Ihre »Oel-Fieber«-Diskette fertig und das Spiel kann beginnen.

Booten Sie neu von Ihrer erstellten Diskette und öffnen Sie die Diskette »Oel-Fieber« mit einem Doppelklick (bitte nicht mit dem Schraubendreher). Dann starten Sie das Programm »Oel-Fieber« und warten bis die neue Bildschirmmaske aufgebaut ist. Wenn alles gutgeht, erscheint die Frage »Altes Spiel?« auf dem Monitor. Drücken Sie einfach die Taste »Return«, um

ein neues Spiel zu starten oder geben »ja« und dann »Return« ein. Nun werden Sie aufgefordert, den Namen des alten Spiels einzugeben. Wenn Sie bereits einen Spielstand abgespeichert haben, geben Sie hier den Namen des Spiels ein und das Spiel beginnt bei der zuletzt gesicherten Spielsituation. Bei einem Neustart erscheint die Frage nach der gewünschten Rundenzahl, Sie geben damit an wie oft ein Spieler das Startfeld (siehe Bild 2) überqueren muß bis zum Spielende. Dabei zählt jedes Überschreiten des Startfeldes als ei-

ne halbe Runde. Geben Sie Ihre gewünschte Rundenzahl ein und bestätigen die Eingabe mit »Return«. Jetzt können Sie die Anzahl der Spieler eingeben, bei »2« spielen Sie gegen sich selbst als Training oder einen Konkurrenten, bei »1« spielen Sie gegen den Amiga. Nach dieser Eingabe und dem folgenden »Return« beginnt das eigentliche Spielgeschehen.

Jetzt sollte der Bildschirm etwa dem von Bild 1 entsprechen. Auf dem Startfeld blinkt ein Spielstein (ein kleiner grüner Kreis) und wartet darauf, daß Sie mit dem Mauszeiger den Würfel in der rechten unteren Ecke anklicken. Der Würfel bewegt sich und zeigt die erreichte Punktezahl an. Sogleich geht Ihr Spielstein um die gewürfelte Felderzahl nach vorne. Wenn Sie gegen den Amiga spielen, bewegt sich nun alles wie von Geisterhand und ein kleiner ro-

ter Kreis symbolisiert den zweiten Spielstein. Bei zwei Spielern erscheint oben eine Schrift, die Spieler 2 auffordert zu würfeln. So, jetzt geht der Streß los und Sie müssen Entscheidungen treffen um Ihr Kapital zu vergrößern. Denn am Ende entscheiden die nackten Zahlen über Sieg oder Niederlage. Wenn Sie das Spiel unterbrechen wollen, können Sie mit einem Klick auf das Diskettensymbol rechts oben den Spielstand speichern. Welche Bedeutung die einzelnen Felder und Anzeigen haben, erfahren Sie in Bild 1 und der folgenden Tabelle. Dann viel Spaß mit »Oel-Fieber« und Vorsicht, dieses Fieber ist ansteckend. *pe*

Manager-Kurs

Versuchen Sie schnellstens einen Bohrturm zu erstehen.

Spekulieren Sie mit Aktien, es lohnt sich. Aber Vorsicht, wenn Sie in die roten Zahlen geraten ist das Spiel aus.

Gehen Sie mit Bedacht an Sabotagen, auch wenn es Sie in den Fingern juckt, sie kosten oft mehr als sie dem Gegner schaden.

Kaufen Sie kein Schiff bevor Sie einen Bohrturm besitzen, sonst kann es geschehen, daß Sie sich keinen Bohrturm mehr leisten können.

Wenn es Ihnen zu mühselig ist, das Programm abzutippen, haben Sie die Möglichkeit, dieses packende Spiel mit unserer Programm-Service-Diskette zu erwerben (siehe Seite 33).

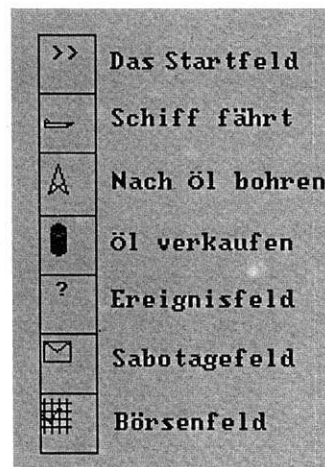


Bild 2. Felder-Übersicht

Programmname:	Oel-Fieber
Computer:	A500, A1000, A2000 mit Kickstart 1.2 oder 1.3
Sprache:	Amiga-Basic
Bemerkung:	Siehe Text

Programmautor: Thomas Schachtleben

```

1 Mt0 CLEAR, 40000%
2 MO SCREEN 1,330,255,4,1
3 eo WINDOW 2," $$$$$$$$ Oel-Fieber $$$$$$$$",(0,0)-(3
  21,240),0,1
4 Cq PALETTE 0,0,.3,0
5 XF PALETTE 1,0,.6,0
6 40 PALETTE 5,0,.25,0
7 ve PALETTE 6,.3,.3,.3
8 JS RANDOMIZE TIMER
9 wU COLOR 9,8
10 vH feldaufb:
11 3v REM felder
12 fY LINE(10,10)-(40,40),1,b:PAINT(12,12),8,1
13 uD FOR x=40 TO 270 STEP 30

```

Listing 1. »Oel-Fieber«


```

14 ze1 LINE(x+7,17)-(x+37,47),5,bf
15 Vt LINE(x,10)-(x+30,40),1,b:PAINT(x+2,12),8,1
16 zS LINE(x+7,167)-(x+37,197),5,bf
17 e6 LINE(x,160)-(x+30,190),1,b:PAINT(x+2,162),8,1
18 NSO NEXT
19 s4 FOR y=40 TO 150 STEP 30
20 QD1 LINE(10,y)-(40,y+30),1,b:PAINT(12,y+2),8,1
21 En LINE(280,y)-(310,y+30),1,b:PAINT(282,y+2),8,1
22 RWO NEXT
23 7o LINE(41,48)-(47,159),5,bf:LINE(17,161)-(39,167),5,bf
24 hF LINE(311,43)-(317,166),5,bf:LINE(287,161)-(310,166),5,bf
25 iU REM 81tonne
26 nr FOR a=1 TO 10
27 hf1 CIRCLE(80,17+a),4,6:CIRCLE(80,17),4,1
28 b0 CIRCLE(290,77+a),4,6:CIRCLE(290,77),4,1
29 n3 CIRCLE(230,168+a),4,6:CIRCLE(230,168),4,1
30 86 CIRCLE(20,107+a),4,6:CIRCLE(20,107),4,1
31 wG CIRCLE(50,47+a),4,14:CIRCLE(50,47),4,1
32 xc CIRCLE(50,67+a),4,9:CIRCLE(50,67),4,1
33 ch0 NEXT
34 w0 REM bohrturm
35 6V LINE(50,15)-(45,30),3:LINE(50,15)-(55,30),3
36 RL LINE(45,30)-(52,23),3:LINE(55,30)-(47,23),3
37 9J LINE(260,165)-(255,180),3:LINE(260,165)-(265,180),3
38 eu LINE(255,180)-(262,173),3:LINE(265,180)-(257,173),3
39 FP LINE(170,165)-(165,180),3:LINE(170,165)-(175,180),3
40 k0 LINE(165,180)-(172,173),3:LINE(175,180)-(167,173),3
41 sv LINE(20,140)-(15,155),3:LINE(20,140)-(25,155),3
42 PE LINE(15,155)-(22,150),3:LINE(25,155)-(18,150),3
43 OC LINE(290,45)-(285,60),3:LINE(290,45)-(295,60),3
44 sA LINE(285,60)-(292,55),3:LINE(295,60)-(288,55),3
45 N1 LINE(265,45)-(260,60),9:LINE(265,45)-(270,60),9
46 ez LINE(260,60)-(267,55),9:LINE(270,60)-(263,55),9
47 pc LINE(240,45)-(235,60),14:LINE(240,45)-(245,60),14
48 J9 LINE(235,60)-(242,55),14:LINE(245,60)-(238,55),14
49 hS REM aktien
50 AS FOR x=1 TO 15 STEP 4
51 eq1 LINE(x+101,13)-(x+101,30),1
52 v00 NEXT
53 JX FOR y=1 TO 15 STEP 4
54 le1 LINE(101,y+14)-(116,y+14),1
55 y30 NEXT
56 xo LINE(100,20)-(105,25),10:LINE(105,25)-(109,18),10
57 Kf LINE(109,18)-(115,19),10
58 Ia FOR x=1 TO 15 STEP 4
59 RN1 LINE(x+101,163)-(x+101,180),1
60 380 NEXT
61 Rf FOR y=1 TO 15 STEP 4
62 ko1 LINE(101,y+164)-(116,y+164),1
63 6B0 NEXT
64 Ds LINE(100,170)-(105,175),10:LINE(105,175)-(109,168),10
65 y1 LINE(109,168)-(115,169),10
66 Qi FOR x=1 TO 15 STEP 4
67 VG1 LINE(x+191,13)-(x+191,30),1
68 BGO NEXT
69 Zn FOR y=1 TO 15 STEP 4
70 RT1 LINE(191,y+14)-(206,y+14),1
71 EJO NEXT
72 f6 LINE(190,20)-(195,25),10:LINE(195,25)-(199,18),10
73 5U LINE(199,18)-(205,19),10
74 Yq FOR x=1 TO 15 STEP 4
75 iw1 LINE(x+191,163)-(x+191,180),1
76 JOO NEXT
77 hv FOR y=1 TO 15 STEP 4
78 Qd1 LINE(191,y+164)-(206,y+164),1
79 MRO NEXT
80 n2 LINE(190,170)-(195,175),10:LINE(195,175)-(199,168),10
81 eq LINE(199,168)-(205,169),10
82 kK REM sabotage
83 7r LINE(222,14)-(236,24),15,b:LINE(222,14)-(229,19),15:LINE(229,19)-(236,14),15
84 Vx LINE(72,164)-(86,174),15,b:LINE(72,164)-(79,169),15:LINE(79,169)-(86,164),15
85 UI REM schiff
86 43 CIRCLE(134,24),1,12
87 Oh LINE(133,25)-(145,27),12,b
88 3V LINE(145,25)-(148,25),12:LINE(148,25)-(145,27),12
89 LN CIRCLE(254,24),1,12
90 L3 LINE(253,25)-(265,27),12,b

```

```

91 gA LINE(265,25)-(268,25),12:LINE(268,25)-(265,27),12
92 bm CIRCLE(284,144),1,12
93 Pn LINE(283,145)-(295,147),12,b
94 Tv LINE(295,145)-(298,145),12:LINE(298,145)-(295,147),12
95 7q CIRCLE(44,174),1,12
96 OY LINE(43,175)-(55,177),12,b
97 Us LINE(55,175)-(58,175),12:LINE(58,175)-(55,177),12
98 fd CIRCLE(14,54),1,12
99 uN LINE(13,55)-(25,57),12,b
100 AU LINE(25,55)-(28,55),12:LINE(28,55)-(25,57),12
101 O8 CIRCLE(46,144),1,14
102 Hg LINE(45,145)-(57,147),14,b
103 Uz LINE(57,145)-(60,145),14:LINE(60,145)-(57,147),14
104 Gn CIRCLE(46,152),1,9
105 D7 LINE(45,153)-(57,155),9,b
106 Y9 LINE(57,153)-(60,153),9:LINE(60,153)-(57,155),9
107 m0 REM zufallsfeld
108 zW LOCATE 3,22:PRINT "?":LOCATE 22,18:PRINT "?"
109 QX LOCATE 14,37:PRINT "?":LOCATE 11,3:PRINT "?"
110 zD LOCATE 3,3:PRINT ">>"
111 ey COLOR 3,0
112 B9 REM wuerfelmalen:
113 Su LINE(280,210)-(300,230),1,b
114 i8 LINE(280,210)-(285,205),1
115 iD LINE(300,210)-(305,205),1
116 AQ LINE(300,230)-(305,225),1
117 um LINE(285,205)-(305,205),1:LINE(305,205)-(305,225),1
118 YY PAINT(282,212),6,1:PAINT(285,208),6,1:PAINT(304,223),6,1
119 R1 LOCATE 13,8:INPUT "Altes Spiel";j$
120 JR LINE(56,61)-(277,120),0,bf
121 5j IF j$="j" THEN altesSpiel
122 aw IF j$<>"j" THEN werteSetzen
123 F3 spielSichern:
124 bi LOCATE 27,1:INPUT "NAME:";spieler$
125 so IF spieler$="" THEN main
126 ly ON ERROR GOTO FEHLER
127 FR1 OPEN spieler$ FOR OUTPUT AS 1
128 iH0 FOR n=1 TO 2
129 LV PRINT #1,xfigur(n);yfigur(n);umf(n);farb(n);geld(n)
130 wv PRINT #1,x(n);weite(n);schiff(n);aktien(n);bohren(n)
131 TA PRINT #1,tiefe(n);leck(n);motor(n);verpfl(n);bunker(n)
132 nF PRINT #1,feuer(n);explo(n);nach(n)
133 EJ NEXT
134 jV PRINT #1,runden;spieler;amiga;oelkurs;aktkurs
135 Nw CLOSE 1:GOTO aus3
136 yH altesSpiel:
137 TB LOCATE 14,8:INPUT "Name";spieler$
138 Ch IF spieler$="" THEN werteSetzen
139 EB ON ERROR GOTO FEHLER
140 qb1 OPEN spieler$ FOR INPUT AS 1
141 i10 WHILE NOT EOF(1)
142 wv1 FOR n=1 TO 2
143 3I2 INPUT #1,xfigur(n),yfigur(n),umf(n),farb(n),geld(n)
144 e1 INPUT #1,x(n),weite(n),schiff(n),aktien(n),bohren(n)
145 w8 INPUT #1,tiefe(n),leck(n),motor(n),verpfl(n),bunker(n)
146 vw INPUT #1,feuer(n),explo(n),nach(n)
147 SX1 NEXT
148 RV INPUT #1,runden,spieler,amiga,oelkurs,aktkurs
149 Ga SOUND 800,4,,0:SOUND 850,4,,1
150 3E2 FOR x=1 TO 5000:NEXT
151 Wg1 CLOSE 1:GOTO losgehts
152 th0 WEND
153 TB werteSetzen:
154 X6 xfigur(1)=35:yfigur(1)=15:umf(1)=4:farb(1)=14:geld(1)=10000
00&
155 ij x(1)=241:weite(1)=60:aktien(1)=100
156 hY xfigur(2)=35:yfigur(2)=15:umf(2)=2:farb(2)=9:geld(2)=100000
0&
157 LW x(2)=266:weite(2)=60:aktien(2)=100
158 bT spieler:
159 8Y BEEP:runden=0
160 Ha LOCATE 13,22:PRINT " "
161 A5 LOCATE 13,8:INPUT "Wieviel Runden";runden
162 zF IF runden<1 OR runden>99999& THEN spieler
163 nU LOCATE 14,8:PRINT "OK !"
164 bX LOCATE 9,8:INPUT "Wieviel Spieler(max.2)";spieler
165 3Z IF spieler<1 OR spieler>2 THEN spieler
166 ck IF spieler=1 THEN
167 Oz1 LOCATE 11,8:PRINT "OK! AMIGA spielt mit Dir"

```



```

168 eX0 END IF
169 Ir IF spieler=2 THEN
170 Au1 LOCATE 11,8:PRINT "Viel Spass !"
171 ha0 END IF
172 Se FOR x=1 TO 6000:NEXT
173 fT losgehts:
174 vJ LINE(56,61)-(277,120),0,bf
175 WP LINE(198,48)-(204,132),1,b
176 CL FOR y=50 TO 130 STEP 10
177 Pr1 LINE(195,y)-(197,y),1
178 x20 NEXT
179 Bv LOCATE 7,23:PRINT "50":LOCATE 12,24:PRINT "$":LOCATE 17,22:
PRINT "130"
180 L3 LINE(41,139)-(279,139),5:LINE(224,41)-(224,139),5
181 s8 LINE(41,135)-(279,135),1:LINE(220,41)-(220,135),1
182 mo SOUND 300,5,50
183 Ol LOCATE 23,3:PRINT "JA"
184 hh LOCATE 23,37:PRINT "NEIN"
185 JO sp=1:n=1
186 9C main:
187 zw FOR zoegern=1 TO 200:NEXT
188 Zq LINE(0,200)-(278,240),0,bf
189 Gs IF geld(n)<1 THEN aus2
190 5y jn$=""
191 NV IF aktkurs<50 THEN aktkurs=50
192 vE IF aktkurs>130 THEN aktkurs=130
193 li IF oelkurs<1 THEN oelkurs=1
194 Oq IF oelkurs>12 THEN oelkurs=12
195 mK CIRCLE(xfigur(n),yfigur(n)),umf(n),8
196 Yx LINE(x(1),60)-(x(1),tiefe(1)+60),farb(1)
197 pI LINE(x(2),60)-(x(2),tiefe(2)+60),farb(2)
198 Yl IF amiga=1 THEN
199 Mm1 COLOR 9,0
200 s9 LOCATE 1,2:PRINT "AMIGA ist am Zug ! ";geld(n);"$"
"
201 B40 END IF
202 2o IF sp=1 OR sp=2 THEN
203 Rh2 IF n=1 THEN COLOR 14,0
204 Ir IF n=2 THEN COLOR 9,0
205 MX1 LOCATE 1,2:PRINT "Spieler";sp;"ist am Zug ! ";geld(n);"$"
"
206 G90 END IF
207 BH LOCATE 12,7:PRINT "Ölkurs:";oelkurs;"$t "
208 kT LOCATE 8,9:PRINT bunker(1);"t";" "
209 A3 LOCATE 10,9:PRINT bunker(2);"t";" "
210 kI LOCATE 14,7:PRINT "Aktien:";aktien(n)
211 RO dx=-1:dy=-1
212 UH WHILE (MOUSE(0)<0)
213 q51 PAINT(199,55),0,1
214 Kh CIRCLE(200,aktkurs),1,10
215 Go2 IF bohren(1)=0 THEN
216 pK3 LINE(x(1),60)-(x(1),130),0
217 RK2 END IF
218 db IF bohren(2)=0 THEN
219 OX3 LINE(x(2),60)-(x(2),130),0
220 UN2 END IF
221 ut LINE(48,141)-(279,159),0,bf
222 5F CIRCLE(weite(1)-14,144),1,14
223 sU LINE(weite(1)-15,145)-(weite(1)-3,147),14,b
224 5J LINE(weite(1)-3,145)-(weite(1),145),14
225 zQ LINE(weite(1),145)-(weite(1)-3,147),14
226 GE CIRCLE(weite(2)-14,152),1,9
227 md LINE(weite(2)-15,153)-(weite(2)-3,155),9;b
228 Bd LINE(weite(2)-3,153)-(weite(2),153),9
229 5K LINE(weite(2),153)-(weite(2)-3,155),9
230 X51 dx=MOUSE(1):dy=MOUSE(2)
231 Ay0 WEND
232 Lx CIRCLE(xfigur(n),yfigur(n)),umf(n),farb(n)
233 7K IF amiga=1 THEN
234 GO1 FOR x=1 TO 2000:NEXT
235 oS dx=290:dy=220
236 kd0 END IF
237 P1 mus=mus+1
238 gt IF mus=20 THEN
239 od2 FOR st=40 TO 10 STEP -8
240 X73 SOUND WAIT
241 9a SOUND 200,2,st,1:SOUND 250,2,st,0
242 un SOUND 300,2,st,2:SOUND 350,2,st,3
243 ZF SOUND 200,2,st-5,1:SOUND 200,2,st-10,2
244 Yt4 FOR x=1 TO 2

```

```

245 7o5 SOUND 200,2,5,st-15,3:SOUND 200,2,st-15,0
246 384 NEXT
247 on3 SOUND RESUME
248 5A2 NEXT
249 rF1 mus=0
250 yr0 END IF
251 4r IF dx=-1 THEN main
252 uI mus=0
253 7M IF dx>280 AND dx<300 AND dy>210 AND dy<230 THEN
254 Oq1 LOCATE 3,37:PRINT " "
255 BF LINE(290,10)-(315,35),1,b
256 DP LINE(295,25)-(310,35),1,b:LINE(298,27)-(302,33),1,b
257 km LINE(312,12)-(314,14),1,b
258 VW GOTO wuerfel
259 700 END IF
260 Lx IF dx>288 AND dx<320 AND dy>15 AND dy<34 THEN spielSich
ern
261 p8 LOCATE 3,37:PRINT "Th.S"
262 XF GOTO main
263 a2 wuerfel:
264 UW SOUND 261,1,40,1:SOUND 329,2,40,2:SOUND 392,1,40,3
265 xG FOR x=1 TO 500:NEXT
266 9Z FOR x=1 TO 7
267 tg1 SOUND x*250,.5,30,RND
268 JS fb=RND*13+2
269 Nn PAINT(282,212),fb,1:PAINT(285,208),fb,1:PAINT(304,223),fb,
1
270 RWO NEXT
271 t7 PAINT(282,212),10,1:PAINT(285,208),11,1:PAINT(304,223),3,1
272 8L wurf=INT(RND*6)+1
273 n0 IF wurf=1 THEN
274 O71 oelkurs=oelkurs-1:aktkurs=aktkurs-10
275 YB CIRCLE(290,220),2,6
276 Jg GOTO zug
277 PIO END IF
278 xB IF wurf=2 THEN
279 ZE1 oelkurs=oelkurs-1:aktkurs=aktkurs-20
280 tx CIRCLE(285,215),2,6:CIRCLE(295,225),2,6
281 ol GOTO zug
282 UNO END IF
283 7M IF wurf=3 THEN
284 hN1 oelkurs=oelkurs-2:aktkurs=aktkurs-20
285 be CIRCLE(285,215),2,6:CIRCLE(290,220),2,6:CIRCLE(295,225),2,
6
286 tq GOTO zug
287 ZSO END IF
288 HX IF wurf=4 THEN
289 P41 oelkurs=oelkurs+1:aktkurs=aktkurs+10
290 37 CIRCLE(285,215),2,6:CIRCLE(295,225),2,6
291 48 CIRCLE(295,215),2,6:CIRCLE(285,225),2,6
292 zw GOTO zug
293 fYO END IF
294 SJ IF wurf=5 THEN
295 YE1 oelkurs=oelkurs+2:aktkurs=aktkurs+10
296 9D CIRCLE(285,215),2,6:CIRCLE(295,225),2,6
297 uX CIRCLE(290,220),2,6
298 BF CIRCLE(295,215),2,6:CIRCLE(285,225),2,6
299 63 GOTO zug
300 mf0 END IF
301 ew IF wurf=6 THEN
302 lN1 oelkurs=oelkurs+2:aktkurs=aktkurs+20
303 GK CIRCLE(285,215),2,6:CIRCLE(295,225),2,6
304 XS CIRCLE(285,220),2,6:CIRCLE(295,220),2,6
305 IM CIRCLE(295,215),2,6:CIRCLE(285,225),2,6
306 DA GOTO zug
307 tm0 END IF
308 Hz GOTO main
309 g4 zug:
310 d1 FOR leis=150 TO 1 STEP -8
311 8m1 SOUND 261*2,.5,leis,1:SOUND 329*2,.5,leis,2:SOUND 392*2,.5
,leis,3
312 7CO NEXT
313 aj FOR x=1 TO 3000:NEXT
314 hF CIRCLE(xfigur(n),yfigur(n)),umf(n),8
315 Ue FOR weiter=1 TO wurf
316 DA2 IF xfigur(n)=35 AND yfigur(n)=15 THEN
317 KQ3 runden=runden-.5

```

Listing 1. (Fortsetzung)


```

318 DG IF runden=-1 THEN
319 g34 aus2:
320 C3 geld(1)=geld(1)+(aktien(1)*60)
321 QP geld(2)=geld(2)+(aktien(2)*60)
322 RI bunt=RND*9
323 EF COLOR 10,bunt
324 rF5 FOR y=300 TO 0 STEP -.5
325 Gq5 LINE(0,y)-(330,y),bunt
326 LQ5 NEXT
327 K54 LOCATE 23,3:PRINT "JA"
328 11 LOCATE 23,37:PRINT "NEIN"
329 jē5 IF geld(1)>geld(2) THEN
330 6y6 LOCATE 15,1:PRINT "SPIELER 1 HAT GEWONNEN: ";geld(1);
"$"
331 rr PRINT "Spieler 2 hat verloren: ";geld(2);"$"
332 IB5 END IF
333 p14 IF geld(2)>geld(1) THEN
334 K95 LOCATE 15,1:PRINT "SPIELER 2 HAT GEWONNEN: ";geld(2);
"$"
335 pn PRINT "Spieler 1 hat verloren: ";geld(1);"$"
336 MF4 END IF
337 ph IF geld(1)=geld(2) THEN
338 Jv5 LOCATE 15,1:PRINT "!! SPIEL ENDE !!"
339 5b PRINT " ! GLEICHSTAND !"
340 QJ4 END IF
341 7V0 aus3:
342 rt3 LOCATE 27,1:PRINT " NOCH MAL ??? "
343 n9 GOSUB jainein
344 81 IF jn$="j" THEN RUN
345 Jd SCREEN CLOSE 1:WINDOW CLOSE 2:END
346 WP2 END IF
347 LQ IF feuer(n)>.5 THEN feuer(n)=feuer(n)-1
348 Q7 IF explo(n)>.5 THEN explo(n)=explo(n)-1
349 EN IF nach(n)>.5 THEN nach(n)=nach(n)-1
350 2G IF leck(n)>.5 THEN leck(n)=leck(n)-1
351 RP IF motor(n)>.5 THEN motor(n)=motor(n)-1
352 JR IF verpfl(n)>.5 THEN verpfl(n)=verpfl(n)-1
353 YG FOR x=1 TO 30000 STEP 500
354 qd3 geld(n)=geld(n)+500
355 ER LOCATE 1,25:PRINT geld(n)
356 1C SOUND x/10,.3
357 qv2 NEXT
358 1b1 END IF
359 6Z SOUND 220,1:SOUND 261,1,,1:SOUND 293,1,,2
360 Wn IF yfigur(n)=15 AND weiter<>0 THEN
361 Ru2 xfigur(n)=xfigur(n)+30
362 mf1 END IF
363 mm IF xfigur(n)=305 THEN
364 Y32 yfigur(n)=yfigur(n)+30
365 p11 END IF
366 Kv IF yfigur(n)=165 THEN
367 J42 xfigur(n)=xfigur(n)-30
368 s11 END IF
369 ZB IF xfigur(n)=35 THEN
370 qD2 yfigur(n)=yfigur(n)-30
371 vo1 END IF
372 bD CIRCLE(xfigur(n),yfigur(n)),umf(n),farb(n)
373 SZ FOR x=1 TO 1000:NEXT
374 fD CIRCLE(xfigur(n),yfigur(n)),umf(n),8
375 550 NEXT weiter
376 fH CIRCLE(xfigur(n),yfigur(n)),umf(n),farb(n)
377 CF SOUND 392,1,,3
378 dE IF yfigur(n)=15 THEN
379 3a1 IF xfigur(n)=35 THEN kontrolle
380 1G IF xfigur(n)=65 THEN bohren
381 ff IF xfigur(n)=95 THEN bunker
382 SY IF xfigur(n)=125 OR xfigur(n)=215 THEN aktien
383 PG IF xfigur(n)=155 THEN schiff
384 cq IF xfigur(n)=245 THEN sabotage
385 r7 IF xfigur(n)=185 THEN zufall
386 JT IF xfigur(n)=275 THEN schiff
387 B40 END IF
388 BB IF xfigur(n)=305 THEN
389 5J1 IF yfigur(n)=45 THEN bohren
390 j1 IF yfigur(n)=75 THEN bunker
391 G3 IF yfigur(n)=105 THEN zufall
392 RR IF yfigur(n)=135 THEN schiff
393 HAO END IF
394 mN IF yfigur(n)=165 THEN
395 VQ1 IF xfigur(n)=275 THEN bohren
396 K6 IF xfigur(n)=245 THEN bunker

```

```

397 hs IF xfigur(n)=215 OR xfigur(n)=125 THEN aktien
398 ZP IF xfigur(n)=185 THEN bohren
399 1R IF xfigur(n)=95 THEN sabotage
400 oG IF xfigur(n)=155 THEN zufall
401 By IF xfigur(n)=65 THEN schiff
402 QJ0 END IF
403 7J IF xfigur(n)=35 THEN
404 GR1 IF yfigur(n)=135 THEN bohren
405 57 IF yfigur(n)=105 THEN bunker
406 TN IF yfigur(n)=75 THEN zufall
407 Cy IF yfigur(n)=45 THEN schiff
408 WPO END IF
409 o9 weiter:
410 CM FOR x=1 TO 4000:NEXT
411 Zh IF spieler=1 THEN
412 Xa1 IF sp=1 THEN ami
413 FN IF sp=0 THEN spi
414 eV0 END IF
415 gp IF spieler=2 THEN
416 De1 IF sp=1 THEN sp12
417 Ef IF sp=2 THEN sp11
418 gZ0 END IF
419 4m GOTO main
420 PI ami:
421 bw n=2:sp=0:amiga=1
422 7p GOTO main
423 BP spi:
424 cw n=1:sp=1:amiga=0
425 As GOTO main
426 1Q sp1:
427 7D n=1:sp=1
428 Dv GOTO main
429 9Z sp12:
430 IQ n=2:sp=2
431 Cy GOTO main
432 A8 kontrolle:
433 zf FOR x=1 TO 80000& STEP 500
434 8v1 geld(n)=geld(n)+500
435 WJ LOCATE 1,25:PRINT geld(n)
436 Ks SOUND x/50,.3
437 8D0 NEXT
438 hb GOTO weiter
439 kn bohren:
440 Jn IF feuer(n)>.5 OR explo(n)>.5 OR nach(n)>.5 THEN
441 X01 LOCATE 27,1:PRINT "S A B O T A G E !"
442 4x0 END IF
443 uR IF bohren(n)=0 THEN
444 of1 tiefe(n)=0:turmpreis=300000&+(aktien(n)*25)+(aktkurs*1000)
445 1d LOCATE 27,1:PRINT "Bohrturm zu verkaufen: ";turmpreis;"$"
446 vh PRINT "Möchten Sie kaufen?"
447 q24 IF amiga=1 AND geld(n)>turmpreis+50000& THEN
448 oP5 LOCATE 28,1:PRINT " AMIGA kauft ! ":dx=20:dy=17
0
449 s3 FOR x=1 TO 5000:NEXT
450 C54 END IF
451 mw IF amiga=1 AND geld(n)<turmpreis+50000& THEN
452 KE5 dx=290:dy=170
453 F84 END IF
454 aw1 GOSUB jainein
455 NY2 IF jn$="j" THEN
456 tA3 LINE(0,200)-(278,240),0,bf
457 Hq bohren(n)=1
458 1H SOUND 500,5,,1:SOUND 550,5,,0
459 n0 geld(n)=geld(n)-turmpreis
460 v8 LOCATE 1,25:PRINT geld(n)
461 wk aktien(n)=aktien(n)+1500
462 OH2 END IF
463 PIO END IF
464 b8 IF bohren(n)=1 AND feuer(n)=0 AND explo(n)=0 AND nach(n)=0
THEN
465 E11 tiefe(n)=tiefe(n)+(wurf*5)
466 Rs2 IF tiefe(n)+60>130 OR tiefe(n)+60=130 THEN
467 ov3 IF bunker(n)<50001& THEN
468 WJ4 bunker(n)=bunker(n)+50000&
469 Zt LOCATE 27,1:PRINT "Fündig geworden !"
470 WP3 END IF
471 JP tiefe(n)=70
472 YR2 END IF
473 qJ1 LINE(x(n),60)-(x(n),130),0
474 m7 LINE(x(n),60)-(x(n),tiefe(n)+60),farb(n)

```



```

475 bu0 END IF
476 JD GOTO weiter
477 bn bunker:
478 OE LOCATE 8,9:PRINT bunker(1);"t"
479 vJ LOCATE 10,9:PRINT bunker(2);"t"
480 Ov IF oelkurs<1 THEN oelkurs=1
481 tT IF oelkurs>12 THEN oelkurs=12
482 ag LOCATE 12,7:PRINT "Ölkurs: ";oelkurs;"$/t"
483 xF1 IF angekommen(n)=1 THEN
484 Th2 IF bunker(n)>1 THEN
485 da3 LOCATE 27,1:PRINT "Tageskurs: ";oelkurs;"$/t"
486 5T LOCATE 28,1:PRINT "Verkaufen Sie 50000t"
487 DQ5 IF amiga=1 THEN
488 c86 LOCATE 28,1:PRINT "NEIN !":dx=290:dy=170
489 gw7 IF oelkurs>3 THEN
490 kt8 LOCATE 28,1:PRINT "AMIGA verkauft 50000 t !"
491 co dx=20:dy=170
492 Zk FOR x=1 TO 5000:NEXT
493 tm7 END IF
494 un5 END IF
495 Fb3 GOSUB janein
496 gW IF jn$="j" OR jn$="J" THEN
497 bp5 IF bunker(n)=100000& THEN bohren(n)=0
498 CH4 bunker(n)=bunker(n)-50000&
499 RJ geld(n)=geld(n)+500000&*oelkurs
500 Zm LOCATE 1,25:PRINT geld(n)
501 bx weite(n)=60:angekommen(n)=0
502 v23 END IF
503 3w2 END IF
504 4x1 END IF
505 mg0 GOTO weiter
506 ef aktien:
507 X5 LOCATE 14,7:PRINT "Aktien: ";aktien(n)
508 bq PAINT(199,55),0,1
509 Vd IF aktkurs<50 THEN aktkurs=50
510 3M IF aktkurs>130 THEN aktkurs=130
511 7U CIRCLE(200,aktkurs),1,10
512 Ua IF aktkurs>50 THEN
513 OG1 LOCATE 27,1:PRINT "Aktienankauf"
514 er3 IF amiga=1 THEN
515 8e5 IF aktkurs>81 THEN akt2.1
516 gK IF geld(n)>89001& AND aktkurs<81 THEN
517 306 ak=INT((geld(n)*.15)/aktkurs)
518 uG LOCATE 27,1:PRINT "AMIGA kauft ";ak;"Aktien !"
519 gt aktien(n)=aktien(n)+INT(ak)
520 VM geld(n)=geld(n)-(ak*aktkurs)
521 u7 LOCATE 1,25:PRINT geld(n)
522 3E FOR x=1 TO 5000:NEXT
523 NG5 END IF
524 5z4 GOTO weiter
525 PI3 END IF
526 k61 GOSUB janein
527 X12 IF jn$="j" THEN
528 tI3 INPUT "Wieviel ";ak
529 q3 aktien(n)=aktien(n)+INT(ak)
530 FW geld(n)=geld(n)-(ak*aktkurs)
531 4H LOCATE 1,25:PRINT geld(n)
532 WP2 END IF
533 M30 akt2.1:
534 9Q1 LINE(0,200)-(278,240),0,bf
535 OP LOCATE 27,1:PRINT "Aktien-verkauf"
536 OD4 IF amiga=1 THEN
537 2b5 IF aktkurs>81 AND aktien(n)>10 THEN
538 Pn6 av=INT(aktien(n)*.8)
539 UT LOCATE 27,1:PRINT "AMIGA verkauft ";av;"Aktien !"
540 Zp aktien(n)=aktien(n)-INT(av)
541 fF geld(n)=geld(n)+(av*aktkurs)
542 FS LOCATE 1,25:PRINT geld(n)
543 OZ FOR x=1 TO 5000:NEXT
544 lb5 END IF
545 QK GOTO weiter
546 kd4 END IF
547 5R1 GOSUB janein
548 s3 IF jn$="j" THEN
549 eM0 akt2.2:
550 Ip2 LOCATE 28,1:INPUT " Wieviel ";av
551 c43 IF av>aktien(n) THEN
552 3o4 av=0:GOTO akt2.2
553 rk3 END IF
554 n31 aktien(n)=aktien(n)-INT(av)

```

```

555 tt geld(n)=geld(n)+(av*aktkurs)
556 Tg LOCATE 1,25:PRINT geld(n)
557 vo END IF
558 wp0 END IF
559 kL LOCATE 14,14:PRINT aktien(n)
560 fQ FOR x=1 TO 5000:NEXT
561 x8 LOCATE 14,14:PRINT " "
562 hb GOTO weiter
563 km sabotage:
564 Gy IF n=1 THEN nn=2
565 Hz IF n=2 THEN nn=1
566 SB IF bohren(nn)=1 THEN
567 bq4 IF amiga=1 AND bunker(1)<60000& AND (yfigur(nn)=15 OR
xfigur(nn)=305) THEN
568 Oh5 IF geld(n)>1200000& AND geld(n)<1500000& AND feuer(n)
n)<.5 THEN
569 e5 LOCATE 27,1:PRINT "AMIGA setzt Deinen"
570 5Q PRINT "Bohrturm in Flammen !":jn$="j":GOTO sab1
571 92 END IF
572 rd IF geld(n)>1500000& AND explo(nn)<.5 THEN
573 Ee LOCATE 27,1:PRINT "AMIGA schickt einen"
574 yo PRINT "Bombenexperten !":jn$="j":GOTO sab2
575 D6 END IF
576 W1 IF geld(n)>800000& AND geld(n)<1200000& AND nach(nn)
<.5 THEN
577 SM LOCATE 27,1:PRINT "AMIGA unterbricht Deine"
578 sb PRINT "Nachschubversorgung !":jn$="j":GOTO sab3
579 HA END IF
580 IB4 END IF
581 Xt IF amiga=1 THEN amischiff
582 BZ1 LOCATE 27,1:PRINT "Sabotage am Bohrturm"
583 f1 GOSUB janein
584 Sd2 IF jn$="j" THEN
585 yF3 LINE(0,200)-(278,240),0,bf
586 P1 LOCATE 27,1:PRINT "Feuer(500000 $)"
587 j5 GOSUB janein
588 hE4 IF jn$="j" THEN sab1
589 4b3 PRINT "Explosion(700000& $)"
590 m8 GOSUB janein
591 qJ4 IF jn$="j" THEN sab2
592 co3 PRINT "Nachschubstop(200000 $)"
593 pB GOSUB janein
594 zO4 IF jn$="j" THEN sab3
595 Ze0 sab1:
596 ep2 IF jn$="j" THEN
597 503 feuer(nn)=1:geld(nn)=geld(nn)-80000&
598 4T geld(n)=geld(n)-500000&
599 AN LOCATE 1,25:PRINT geld(n)
600 JD GOTO weiter
601 dW2 END IF
602 lp0 sab2:
603 lw2 IF jn$="j" THEN
604 YZ3 explo(nn)=2:geld(nn)=geld(nn)-120000&
605 H1 geld(n)=geld(n)-700000&
606 6a LOCATE 1,25:PRINT geld(n)
607 QK GOTO weiter
608 kd2 END IF
609 x20 sab3:
610 s32 IF jn$="j" THEN
611 Ng3 nach(nn)=1:geld(nn)=geld(nn)-20000
612 9V geld(n)=geld(n)-200000&
613 WQ GOTO weiter
614 qJ2 END IF
615 rk1 END IF
616 s10 END IF
617 Cb amischiff:
618 Vm LINE(0,200)-(278,240),0,bf
619 V1 IF schiff(nn)=1 THEN
620 O94 IF amiga=1 AND (yfigur(nn)=15 OR xfigur(nn)=305) THEN
621 mN5 IF geld(n)>1600000& AND leck(nn)<.5 THEN
622 wb LOCATE 27,1 PRINT "AMIGA schlägt Deinen"
623 58 PRINT "Tanker leck !":jn$="j":GOTO sab4
624 Ot END IF
625 qa IF geld(n)>1200000& AND geld(n)<1600000& AND motor(n)
n)<.5 THEN
626 7i LOCATE 27,1:PRINT "AMIGA sabotiert Deinen"
627 Fu PRINT "Tankermotor !":jn$="j":GOTO sab5

```

Listing 1. (Fortsetzung)


```

628 4x      END IF
629 Kv      IF geld(n)>800000& AND geld(n)<1200000& AND verpfl(n)
           <.5 THEN
630 Io      LOCATE 27,1:PRINT "AMIGA verdirbt Deine"
631 Cp      PRINT "Verpflegung an Bord !":jn$="j":GOTO sab6
632 81      END IF
633 924     END IF
634 SN      IF amiga=1 THEN weiter
635 Te0     LOCATE 27,1:PRINT "Sabotage am Schiff"
636 Ws      GOSUB janein
637 JU1     IF jn$="j" THEN
638 p62     LINE(0,200)-(278,240),0,bf
639 7N      LOCATE 27,1:PRINT "Leck(900000 $)"
640 aw      GOSUB janein
641 qB3     IF jn$="j" THEN sab4
642 2d2     PRINT "Motor(800000 $)"
643 dz      GOSUB janein
644 zG3     IF jn$="j" THEN sab5
645 dD2     PRINT "Verpflegung(100000 $)"
646 g2      GOSUB janein
647 8L3     IF jn$="j" THEN sab6
648 f10     sab4:
649 Vg2     IF jn$="j" THEN
650 Db3     leck(nn)=2:geld(nn)=geld(nn)-200000&
651 7a      geld(n)=geld(n)-900000&
652 1E      LOCATE 1,25:PRINT geld(n)
653 A4      GOTO weiter
654 UN2     END IF
655 ry0     sab5:
656 cn2     IF jn$="j" THEN
657 t13     motor(nn)=1:geld(nn)=geld(nn)-100000&
658 Bd      geld(n)=geld(n)-800000&
659 GA      GOTO weiter
660 aT2     END IF
661 2A0     sab6:
662 it2     IF jn$="j" THEN
663 tf3     verpfl(nn)=1:geld(nn)=geld(nn)-10000
664 wH      geld(n)=geld(n)-100000&
665 MG      GOTO weiter
666 gZ2     END IF
667 ha1     END IF
668 ib0     END IF
669 QK      GOTO weiter
670 e0      zufall:
671 Tq      zuff=INT(RND*25)+1
672 p7      ON zuff GOTO 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,1
           9,20,21,22,23,24,25
673 5m      1 :
674 xz      IF weite(n)>60 THEN
675 aq1     x$="MOTORSCHADEN am Schiff !":motor(n)=1:GOTO anzeige
676 qJ0     END IF
677 JR      x$="Aktien fallen auf 50 $":aktkurs=50:GOTO anzeige
678 Cu      2 :
679 Oe      IF xfigur(n)>140 THEN
680 Xd1     x$="Aktentasche vergessen":y$="zurück zum Bohrturm"
681 cA      CIRCLE(xfigur(n),yfigur(n)),umf(n),8
682 ad      xfigur(n)=65:yfigur(n)=15:GOTO anzeige
683 xQ0     END IF
684 Ie      x$="Aktien steigen auf 130 $":aktkurs=130:GOTO anzeige
685 I4      3 :
686 uS      IF bohren(n)=1 THEN
687 yK1     x$="FEUER am Bohrturm !":feuer(n)=1:GOTO anzeige
688 2v0     END IF
689 90      x$="Ölkurs fällt auf 1 $":oelkurs=1:GOTO anzeige
690 SC      4 :
691 xp      x$="Ölkurs steigt auf 12 $":oelkurs=12:GOTO anzeige
692 WH      5 :
693 Jr      IF aktien(n)>300 THEN
694 tb1     aakk=aktien(n)*40
695 G9      x$="Aktienausschüttung ":y$="plus"
696 WH      geld(n)=geld(n)+aakk:GOTO anzeige
697 B40     END IF
698 3D      x$="Zu kleines Aktienpaket":y$="Kein Mitspracherecht, -180
           00 $"
699 4R      geld(n)=geld(n)-18000:GOTO anzeige
700 gS      6 :
701 OQ      IF weite(n)>60 THEN
702 tU1     x$="Reedergebühren sind fällig"
703 UR      y$="minus 15000 $":geld(n)=geld(n)-15000:GOTO anzeige
704 IBO     END IF

```

```

705 NU      IF aktien(n)>1000 THEN
706 ia1     x$="Börsenkrach !":y$="minus 500 aktien"
707 EE      aktien(n)=aktien(n)-500:GOTO anzeige
708 MFO     END IF
709 16      x$="Weihnachtsgeschenke gekauft":y$="minus 12004 $"
710 zK      geld(n)=geld(n)-12004:GOTO anzeige
711 tg      7 :
712 Ks      IF bohren(n)=1 THEN
713 z61     x$="Arbeiter erringen"
714 xp      y$="Gehaltserhöhung, minus 51100 $"
715 hw      geld(n)=geld(n)-51100&:GOTO anzeige
716 UNO     END IF
717 zJ      x$="Arbeiter werden ausgesperrt":y$="plus 63200 $"
718 wD      geld(n)=geld(n)+63200&:GOTO anzeige
719 3r      8 :
720 I5      x$="Geschäftsorganisatorischer-"
721 e2      y$="Gewinn, plus 84300 $":geld(n)=geld(n)+84300&:GOTO anzei
           ge
722 8x      9 :
723 85      x$="Arbeiter erhalten Urlaubsgeld"
724 DA      y$="minus 60000 $":geld(n)=geld(n)-60000&:GOTO anzeige
725 74      10 :
726 p1      x$="Steuerrückzahlung plus 45200 $"
727 70      geld(n)=geld(n)+45200&:GOTO anzeige
728 DB      11 :
729 qI      x$="Steuer-nachzahlung minus 53010 $"
730 5H      geld(n)=geld(n)-53010&:GOTO anzeige
731 JI      12 :
732 Z6      IF bohren(n)=0 THEN
733 WA1     x$="Probebohrung war erfolgreich"
734 W1      y$="plus 18000 $":geld(n)=geld(n)+18000:GOTO anzeige
735 ng0     END IF
736 eS      x$="Bohrloch verschüttet !":y$="minus 39299 $"
737 id      geld(n)=geld(n)-39299&:GOTO anzeige
738 TT      13 :
739 sj      x$="Mitarbeiter veruntreut Gelder"
740 7C      y$="minus 33310 $":geld(n)=geld(n)-33310&:GOTO anzeige
741 Za      14 :
742 eJ      IF geld(n)<800000& THEN
743 mf1     x$="Abschluss einer Firmenübernahme"
744 dG      y$="plus 330500 $":geld(n)=geld(n)+330500&:GOTO anzeige
745 xQ0     END IF
746 mT      IF geld(n)>800000& THEN
747 qJ      x$="Abschluss einer Firmenübernahme"
748 mv      y$="plus 102017 $":geld(n)=geld(n)+102017&:GOTO anzeige
749 Iu      END IF
750 In      15 :
751 IY      IF geld(n)>800000& AND amiga=0 THEN
752 x41     LOCATE 27,1:PRINT "Ein Glücksspiel (evtl.+500000$)?"
753 P1      GOSUB janein
754 CN      IF jn$="j" THEN
755 BR3     glueck=INT(RND*8)+1:gl=1:glueckzeig=1
756 nc      gluecksdm=INT(RND*400001&)+100000&
757 5L      LOCATE 27,1:PRINT "Stopst Du bei ";glueck;" , + ";gluecks
           dm;" $"
758 sj4     WHILE MOUSE(0)<>-1
759 Pq5     SOUND glueckzeig*250,.5,,RND*3
760 q0      LOCATE 28,gl:PRINT glueckzeig
761 5P6     FOR x=1 TO 150:NEXT
762 825     LOCATE 28,gl:PRINT "XXX"
763 k9      glueckzeig=glueckzeig+1
764 W5      gl=gl+3
765 FR6     IF glueckzeig>10 THEN glueckzeig=1:gl=1
766 nb4     WEND
767 xV      LOCATE 28,gl:PRINT glueckzeig
768 HO      IF glueckzeig=glueck THEN geld(n)=geld(n)+gluecksdm
769 st5     IF glueckzeig<>glueck THEN
770 Sc      pechdm=INT(RND*100001&)
771 zc6     LOCATE 27,1:PRINT "Pech gehabt , minus";pechdm;" $"
772 v4      geld(n)=geld(n)-pechdm
773 PI5     END IF
774 4E      FOR x=1 TO 4000:NEXT
775 824     GOTO weiter
776 SL1     END IF
777 aG      geld(n)=geld(n)-50000&:GOTO weiter
778 UNO     END IF
779 cF      x$="Beamtenbestechung minus 120000 $"
780 TR      geld(n)=geld(n)-120000&:GOTO anzeige
781 JM      16 :

```



```

782 2c x$="Steuerhinterziehung plus 4568 $"
783 hS geld(n)=geld(n)+4568:GOTO anzeige
784 PT 17 :
785 Gu IF geld(n)>500000& THEN
786 161 x$="Unfall gebaut,minus 23100 $":geld(n)=geld(n)-23100
787 J1 GOTO anzeige
788 eX0 END IF
789 t0 x$="Für zu schnelles Fahren"
790 TO y$="minus 217 $":geld(n)=geld(n)-217:GOTO anzeige
791 Ze 18 :
792 NB x$="Eine Golduhr zum Geburtstag"
793 d2 y$="plus 854 $":geld(n)=geld(n)+854:GOTO anzeige
794 f1 19 :
795 2u x$="Flugzeug im Hangar minus 83000 $"
796 CV geld(n)=geld(n)-83000&:GOTO anzeige
797 JH 20 :
798 1J x$="Firma abgestossen plus 187300 $"
799 xu geld(n)=geld(n)+187300&:GOTO anzeige
800 P0 21 :
801 PW IF geld(n)>1500000& THEN
802 PW1 x$="Feuer im Büro"
803 aA y$="minus 502000& $"
804 57 geld(n)=geld(n)-502000&:GOTO anzeige
805 v00 END IF
806 ax x$="Auto gestohlen, minus 50000 $"
807 2F geld(n)=geld(n)-50000&:GOTO anzeige
808 aa 22 :
809 Xe IF geld(n)>1500000& THEN
810 J51 x$="Vierus im Computerverbund"
811 P3 y$="minus 400300 $"
812 I5 geld(n)=geld(n)-400300&:GOTO anzeige
813 Jw0 END IF
814 sE x$="Scheck und Karte geklaut"
815 NJ y$="minus 18312 $"
816 Mk geld(n)=geld(n)-18312:GOTO anzeige
817 mn 23 :
818 2a IF bohren(n)=1 THEN
819 ho2 IF geld(n)>1500000& THEN
820 Fk3 x$="Bohrgestänge gebrochen"
821 mV y$="minus 290100 $"
822 km geld(n)=geld(n)-290100&:GOTO anzeige
823 D62 END IF
824 x11 x$="Neuer Bohrkopf fällig"
825 m7 y$="minus 40000 $"
826 IU0 geld(n)=geld(n)-40000&:GOTO anzeige
827 HA END IF
828 M8 x$="Konkurrenz kauf Claim":y$="minus 5912 $"
829 NY geld(n)=geld(n)-5912&:GOTO anzeige
830 24 24 :
831 t0 IF geld(n)>1500000& THEN
832 w11 x$="EHEPARTNER ENTFÜHRT !"
833 sg y$="Lösegeld: 1 300.000 $"
834 8V geld(n)=geld(n)-1300000&:GOTO anzeige
835 P10 END IF
836 5J IF geld(n)>500000& THEN
837 CY1 x$="EHEPARTNER ENTFÜHRT !"
838 XL y$="Lösegeld 350 000 $"
839 i1 geld(n)=geld(n)-350000&:GOTO anzeige
840 UN0 END IF
841 zr x$="Wagen ausgeraubt"
842 zI y$="minus 20000 $"
843 Td geld(n)=geld(n)-20000&:GOTO anzeige
844 JM 25 :
845 7E IF geld(n)>1500000& THEN
846 kv1 x$="Tochterunternehmen pleite"
847 8r y$="minus 523002 $"
848 6D geld(n)=geld(n)-523002&:GOTO anzeige
849 dW0 END IF
850 CD x$="Differenz bei Buchführung"
851 GN y$="fehlender Betrag ungewiss"
852 Co geld(n)=geld(n)-INT(RND*50000&)+1000:GOTO anzeige
853 ND anzeige:
854 MZ SOUND 500,1
855 hG LOCATE 27,1:PRINT x$:LOCATE 28,1:PRINT y$
856 Uh IF aakk>0 THEN PRINT aakk
857 te x$="":y$="":aakk=0
858 69 FOR x=1 TO 10000:NEXT
859 Bf LOCATE 1,25:PRINT geld(n)
860 VP GOTO weiter
861 dQ schiff:

```

```

862 rB IF schiff(n)=0 THEN
863 RK1 schiffpreis=440000&+(aktkurs*1000)
864 g6 LOCATE 27,1:PRINT "Tankschiff zu verkaufen(";schiffpreis;"
$)!"
865 ev PRINT "Möchten Sie kaufen"
866 U64 IF amiga=1 AND geld(n)>schiffpreis+50000& THEN
867 nx LOCATE 28,1:PRINT "AMIGA kauft ein Schiff !"
868 M1 FOR x=1 TO 5000:NEXT:dx=20:dy=170
869 xq END IF
870 Q0 IF amiga=1 AND geld(n)<schiffpreis+50000& THEN
871 5z dx=290:dy=170
872 Ot END IF
873 Lh1 GOSUB janein
874 8J2 IF jn$="j" THEN
875 ev3 LINE(0,200)-(278,240),0,bf
876 7Q schiff(n)=1
877 m2 SOUND 500,5,,1:SOUND 550,5,,0
878 hT geld(n)=geld(n)-schiffpreis
879 gt LOCATE 1,25:PRINT geld(n)
880 812 END IF
881 920 END IF
882 o4 IF leck(n)>.5 OR motor(n)>.5 OR verpfl(n)>.5 THEN
883 f81 LOCATE 27,1:PRINT "S A B O T A G E !"
884 C50 END IF
885 k1 IF schiff(n)=1 AND leck(n)<.5 AND motor(n)<.5 AND verpfl(
n)<.5 THEN
886 nm1 IF bunker(n)>1 AND angekommen(n)=0 THEN
887 7k2 weite(n)=weite(n)+(wurf*25)
888 xi3 IF weite(n)>279 THEN
889 ty4 weite(n)=280
890 6W LOCATE 27,1:PRINT "Schiff ist eingelaufen!"
891 mD PRINT "Liegegebühr 15000 $"
892 mq geld(n)=geld(n)-15000
893 u7 LOCATE 1,25:PRINT geld(n)
894 e7 angekommen(n)=1
895 NG3 END IF
896 nm2 LINE(48,141)-(279,159),0,bf
897 y8 CIRCLE(weite(1)-14,144),1,14
898 lN LINE(weite(1)-15,145)-(weite(1)-3,147),14,b
899 yc LINE(weite(1)-3,145)-(weite(1),145),14
900 sJ LINE(weite(1),145)-(weite(1)-3,147),14
901 97 CIRCLE(weite(2)-14,152),1,9
902 fW LINE(weite(2)-15,153)-(weite(2)-3,155),9,b
903 4W LINE(weite(2)-3,153)-(weite(2),153),9
904 yD LINE(weite(2),153)-(weite(2)-3,155),9
905 XQ1 END IF
906 YR0 END IF
907 GA GOTO weiter
908 oC janein:
909 eY IF n=1 THEN dx=-1
910 EP IF sp=2 THEN dx=-1
911 lY WHILE (MOUSE(0)<0)
912 X5 dx=MOUSE(1):dy=MOUSE(2)
913 Ay WEND
914 F5 LINE(18,185)-(32,188),1,bf:LINE(290,185)-(316,188),1,bf
915 9x LINE(18,185)-(32,188),0,bf:LINE(290,185)-(316,188),0,bf
916 bf IF dx=-1 THEN janein
917 j7 IF dx>10 AND dx<40 AND dy>160 AND dy<190 THEN
918 is1 jn$="j":RETURN
919 le0 END IF
920 SX IF dx>280 AND dx<310 AND dy>160 AND dy<190 THEN
921 TE1 jn$="":RETURN
922 oh0 END IF
923 HQ GOTO janein
924 Vu FEHLER:
925 4a BEEP
926 7t IF ERR=7 THEN LOCATE 15,20:PRINT " Speicher reicht nicht
aus !"
927 R2 IF ERR=53 THEN LOCATE 15,20:PRINT " Datei nicht gefunden !
"
928 PR v=ERR
929 8a PRINT "FEHLER NUMMER :";v
930 GJ FOR x=1 TO 10000:NEXT
931 Va SCREEN CLOSE 1:WINDOW CLOSE 2
932 Ba RUN
933 ni END
(C) 1989 M&T

```

Listing 1. (Schluß)

S P E R R

Ein Labyrinth ist immer wieder faszinierend. Wenn es mit so vielen Variationen arbeitet wie »Maze«, ist es ein Dauerbrenner.

von Hans Waldhäusel

Maze gehört zu den »Labyrinth-Spielen«. Aber Maze ist mehr als nur ein Irrgarten. Maze enthält Elemente aus »Emerald Mine« und »Marble Madness«. Maze ist leicht erweiterbar – es ist in Amiga-Basic geschrieben. Maze verwaltet mehr als 5000 Level. Und Maze stellt Ihnen einen äußerst komfortablen Level-Editor zur Verfügung – damit können Sie Ihre Freunde mit immer neuen Levels »quälen«.

Zur Einstimmung haben wir für Sie ein Bildschirmfoto des ersten Levels abgebildet (Bild 1). Sie sehen darauf alle »Gemeinheiten«, die Ihnen Maze auf der Suche nach dem Ausgang vor die Beine werfen kann – und auch wirft, wenn Sie es zulassen. Um die Erläuterungen besser zuordnen zu können, haben wir, wie Sie sehen, im Bild 1 eingekreiste Zahlen eingeblendet, die mit den Zahlen der Erläuterungen korrespondieren. Eine kurze Erklärung der einzelnen Felder finden Sie in Tabelle 1.

Ihre Aufgabe besteht darin, unter Beachtung des Zeitlimits alle im aktuellen Level vorhandenen Goldstücke einzusammeln, möglichst viele Punkte zu hamstern und sich nicht selbst den Weg zum Ausgang in den nächsten Level zu versperren. Letzteres passiert schneller, als Sie denken. In diesem Fall haben Sie nur drei Möglichkeiten: Möglichkeit 1 (genannt »Geh zurück im Zorn«) ist der berühmte Drei-Finger-Griff; Möglichkeit 2 ist stoisches Warten auf das Ablaufen der Spielzeit; und last but not least – wie die Germanen zu sagen pflegten – Möglichkeit 3 ist ein Druck auf die linke Maustaste, im Spiel auch als »Suicide«-Selbstmord-Taste bezeichnet.

Die Steuerung von »Blinky« erfolgt mittels Joystick (rauf-runter-links-rechts, wie gehabt). Auch eine »Pause«-Funktion wurde vorgesehen: durch

Druck auf den Feuerknopf. Ein nochmaliger Knopfdruck läßt Zeit und Spiel weiterlaufen. Schaffen Sie es, den Ausgang aus dem Level mit allen zu sammelnden Items (schon wieder so eine Eindeutigkeit) innerhalb der zur Verfügung stehenden Zeit zu erreichen, so kommt es zur großen Abrechnung. Nicht nur die in jedem Geldsack steckenden 1000 Punkte werden Ihrem Punktekonto gutgeschrieben. Auch für die verbleibende Zeit bekommen Sie noch Punkte, und zwar für je zehn Sekunden Restzeit ebenso viele Punkte. Haben Sie sich genug an Ihrem Konto-stand ergötzt, drücken Sie den Feuerknopf Ihres Joysticks, um in den nächsten Level eingeladen zu werden.

Maze starten Sie, wie es bei Basic-Programmen üblich ist; also entweder durch Anklicken des »MAZE«-Icons von der Workbench aus oder durch

AmigaBasic MAZE

wenn Sie einen Start aus dem

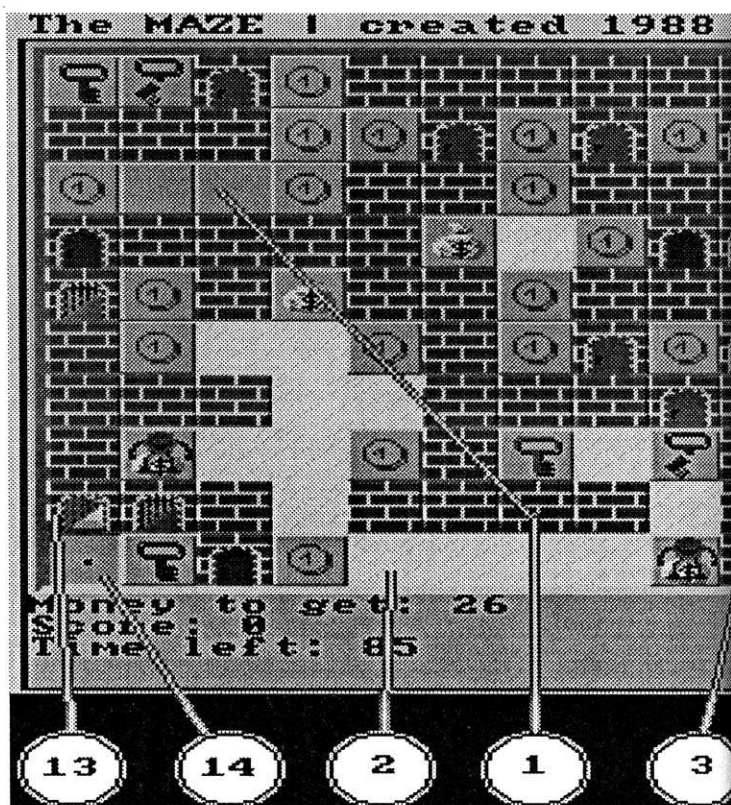


Bild 1. Damit Sie sich in »Maze« nicht völlig verlaufen, finden

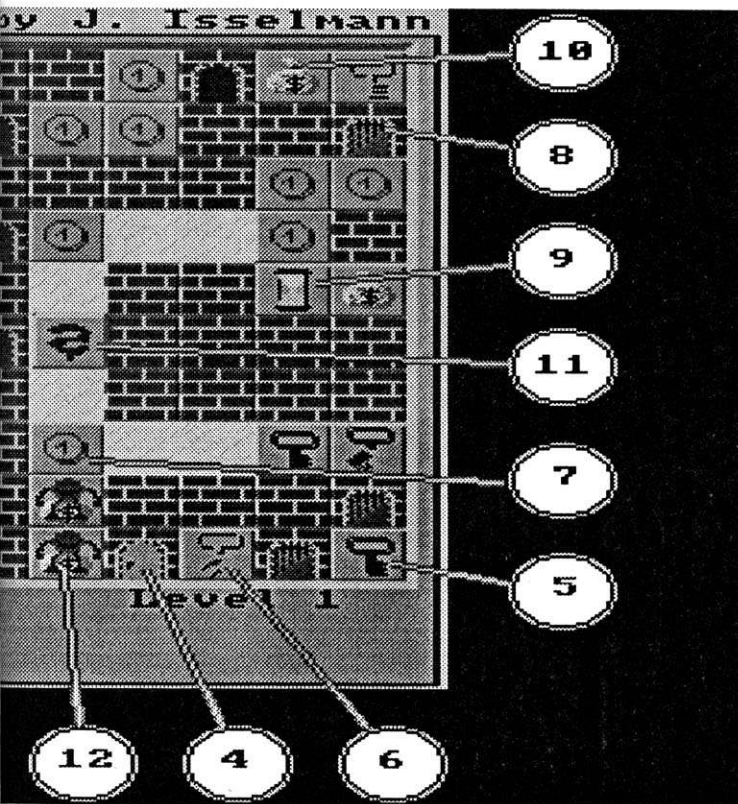
CLI vorziehen. Eine Besonderheit des Programms, man kann es auch als kleines Manko be-

zeichnen, muß an dieser Stelle erwähnt werden: die wenigen Diskettenzugriffe, die nötig

Das Spielfeld

Nr. Element	Effekt	Nr. Element	Effekt
1 »normaler« Boden	volle Manövrierfähigkeit für Ihre Spielfigur	9 Eieruhr	gitter nach unten. Und da bleibt es auch. spendiert Ihnen etwas mehr Zeit; zufallsgesteuert zwischen 20 und 60 Sekunden.
2 »vereister« Boden	hier gibt es kein Halten mehr	10 Dagoberts Liebling	bringt 1000 Punkte und ist das einzige, womit man den »Raubritter« bestechen kann
3 Wand	kein Durchkommen	11 nette Überraschung	ein Aufsammeln dieser Figur bringt Ihnen entweder mehr Gold oder mehr Zeit – der Zufall entscheidet
4 Tür	kein Durchgang ohne den gleichfarbigen Schlüssel. Vorsicht – schließt selbsttätig. Gestrichen in den Farben Rot, Gelb, Grün und Blau. öffnet gleichfarbige Türen. Ein aufgelesener Schlüssel bleibt so lange im Besitz des Spielers, bis dieser ein Feld »6« überquert	12 der Wächter	geben Sie ihm 1000 Punkte, und er gibt Ihnen den Weg frei. Soviel haben Sie nicht? Dann ist er stur wie eine Wand!
5 Schlüssel	Ein aufgelesener Schlüssel bleibt so lange im Besitz des Spielers, bis dieser ein Feld »6« überquert	13 Ausgang	nicht das Tor zu einer besseren Welt, aber wenigstens der Durchgang zum nächsten Level.
6 Schlüsselknacker	auch wieder in Rot, Grün, Gelb oder Blau anzutreffen. Klaut Ihnen den Schlüssel der entsprechenden Farbe. damit erkaufen Sie sich den Passierschein in den nächsten Level.	14 Blinky	Vorausgesetzt, Sie haben brav gesammelt das sind Sie!
7 Goldmünze	Vorsicht!, Einbahnstraße! Sie können nur einmal durch – hinter Ihnen rasselt das Fall-		
8 Burgtor			

G E B I E T



Sie hier alle Symbole und ihre Bedeutung

sind, wurden explizit auf »df0:« festgelegt. Wenn Sie mit dieser Zuweisung nicht glücklich werden können, müssen Sie entsprechende Pfadangaben in den Unterrouinen »Loading:«, »BuildChess:«, »Menu:«, »Saving:« und »Killing:« anstelle von »df0:« einfügen.

Nach dem Start befinden Sie sich in einem kleinen Auswahlmenü. Entscheiden Sie sich bitte, ob Sie gleich spielen oder lieber zuerst die Spielregeln lesen wollen, ob Ihnen der Sinn nach dem Kreieren neuer Levels steht, oder ob Sie gleich wieder den Hut nehmen möchten. Um zu spielen, reicht ein Druck auf den Feuerknopf des Joysticks; für die Spielregeln bewegen Sie den Joystick-Hebel in irgendeine Richtung und den Hut werfen Sie mit <Escape>. Ein Klick der linken Maustaste aktiviert den Level-Editor, dem die nächsten Zeilen und Bild 2 gewidmet sind. Sie meinen, 5000 Level wären doch mehr als genug – und wer braucht da noch einen Level-Editor? Naja... jetzt müssen wir wohl Farbe bekennen.

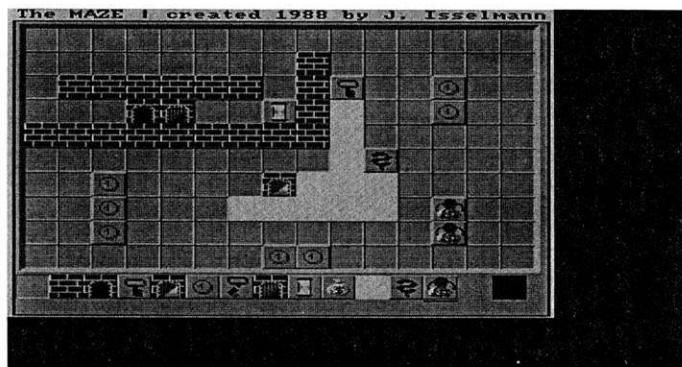


Bild 2. Bastelstube – der Level-Editor von »Maze«

Zu Beginn dieses Artikels haben wir Ihnen den Mund wäbrig gemacht mit dem Hinweis, Maze wäre in der Lage, 5000 Level zu verwalten. Diese Aussage war natürlich richtig, aber Sie werden verstehen (hoffen wir), daß weder Jörg Isselmann, der Autor dieses Programms, noch die Sonderheft-Redaktion die schönsten Jahre ihres Lebens mit dem Entwerfen von 5000 Levels zubringen wollten. Also muß ein Level-Editor her. Voilà – da ist er! In Bild 2 sehen Sie ein im Aufbau begriffenes Level-Bild

und unterhalb des Spielrasters die zur Verfügung stehenden Symbole, über deren Bedeutung ja bereits ausführlich referiert wurde.

Für alle Items, die in verschiedenen Farben auftreten können (Sie erinnern sich: Rot, Gelb, Grün und Blau), ist das Quadrat rechts außen wichtig. Durch Anklicken dieses Feldes wechseln Sie zyklisch die zur Anwendung kommende Bezugsfarbe. Die Auswahl der Spielelemente geschieht durch einfaches Anklicken des gewünschten Symbols in der Symbol-Leiste. Halten Sie die linke Maustaste beim Anklicken so lange gedrückt, bis das gewählte Symbol unterstrichen wird – das ist die Bestätigung des Editors, Ihrem Wunsch entsprochen zu haben. Danach positionieren Sie den Mauszeiger über jener Stelle des Spielrasters, an der Sie das selektierte Symbol platzieren wollen und drücken wieder die linke Maustaste – das war's. Wenn Sie bei gedrückter linker Maustaste die Maus bewegen, zie-

Neben der Symbol-Leiste stellt Ihnen der Level-Editor auch noch eine Menü-Leiste nach gewohnter Amiga-Manier zur Verfügung, in der sich die File-Kommandos und weitere Editier-Routinen anwählen lassen. Nach Anwahl der Menü-Leiste mit der rechten Maustaste sehen Sie ganz links das Menü »Project« und rechts davon das Menü »Do«. »Project« erlaubt mit seinen Unterpunkten »load«, »save« und »kill« das Laden von bereits vorhandenen Level-Bildern, das Abspeichern Ihrer eigenen Kreationen und das Löschen vorhandener Definitions-Files. In allen Fällen müssen Sie die gewünschte Level-Nummer eingeben, wobei bei »load« und »kill« eine Plausibilitätsprüfung auf Vorhandensein der angegebenen Level-Nummer erfolgt und Sie auf eventuelle Irrtümer aufmerksam gemacht werden. »NEW« im Hauptmenü »Project« löscht den Inhalt des Editierfeldes. Durch Anwahl von »RETURN« gelangen Sie zurück in das Spiel-Auswahlmenü, allerdings ohne zum Speichern des Editor-Schirms aufgefordert zu werden. Ein »RETURN« löscht aber den Inhalt des Editier-Feldes – also Vorsicht.

Im Menü »Do« finden Sie »set Start-Point«. Bei Anwahl dieses Punktes definieren Sie mit dem Mauszeiger und Anklicken mit der linken Maustaste die Position von »Blinky« zu Beginn des Levels. »set Time« fordert von Ihnen die Eingabe der zulässigen Spielzeit in Sekunden, wobei jeder Wert zwischen 1 und 2000 zulässig ist. Blicke noch die Funktion »Fill« zu erwähnen. Mit »Fill« füllen Sie den gesamten Spielraster mit jenem Symbol, das in der Symbol-Leiste durch Unterstreichung als aktiv gekennzeichnet ist. Eventuell bereits gesetzte Elemente anderen Typs werden dabei allerdings gnadenlos überschrieben. Sind Sie mit Ihrem Werk zufrieden, so speichern Sie die Level-Definitionen durch Anwahl von »Project« / »save« unter der gewünschten Level-Nummer.

Zum Abschluß der Beschreibung des Level-Editors noch

haben Sie eine ganze Kette von Items, was besonders bei den »Wänden« äußerst praktisch ist. Es gibt keinerlei Beschränkung bezüglich der Anzahl der Symbole je Typ. Die »Goldstücke«, die ja nach den Spielregeln alle einzusammeln sind, werden beim Setzen vom Editor mitgezählt und die Gesamtanzahl wird dann dem Hauptprogramm in der Level-Description übergeben. Sollten Sie zu den Raffzähnen gehören, steht es Ihnen frei, sovieler Geldsäcke zu platzieren, wie Sie wollen.

zwei kleine Anmerkungen: Es macht sicher am meisten Spaß, fremde Level zu knacken. Ein Level-File ist nur 155 Byte groß. Was liegt also näher, als Level-Definitionen unter gleichgesinnten Irrgärtnern auszutauschen. Anmerkung 2 betrifft eine denkbare Erweiterung des Level-Editors. Vorstellbar wäre z.B., nur Start- und Endpunkt einzugeben, evtl. auch noch den Lösungsweg in Form einer verschlungenen Kurve, und der Level-Editor setzt dann die Hindernisse nach dem Zufallsprinzip selbst. Hier gilt: Besonders gelungene Lösungen werden wir gerne veröffentlichen.

Irrgarten zu tauschen

Wie kommen Sie in den Genuß von Maze? Wie (fast) immer gibt es zwei Möglichkeiten. Zuerst die Variante für all jene, denen seitenlanges Listing-Abtippen ein Greuel ist:

Schritt 1: Sie kaufen sich die natürlich auch zu diesem Sonderheft erschienene Leserservice-Diskette (siehe Seite 33).

Schritt 2: kopieren Sie den Inhalt der Schublade Maze mit

```
makeDir ram:MAZE
copy df0:maze to ram:
maze all
```

in die RAM-Disk, legen Sie Ihre Basic-Spiele-Diskette in df0: ein und kopieren Sie nun mit

```
copy ram:MAZE to df0:
all
```

die notwendigen Dateien und Subdirectories in das ROOT-Verzeichnis Ihrer Basic-Spiele-Diskette. Mehr ist nicht zu tun.

Achtung! Das Programm ist so ausgelegt, daß es alle Dateien im Laufwerk df0: und die Level-Informationen im Unterverzeichnis »df0:level« sucht!

Nun zu jenen Lesern, die das Abtippen dem Kauf der Leserservice-Diskette vorziehen (gleich aus welchen Gründen):

Schritt 1: das Hauptprogramm Maze (Listing 1) abtippen und unter beliebigem Namen auf Ihrer Basic-Spiele-Diskette speichern (achten Sie auf die oben angeführten Directory-Bedingungen).

Schritt 2: Object-Generator (Listing 2) abtippen und als »ObjectGenerator« im Hauptverzeichnis abspeichern.

Schritt 3: Level-Daten für die Levels 1 bis 9 (Listing 3, 4, 5, 6, 7, 8, 9, 10 und 11) abtippen, jedes Programm einmal durchlaufen lassen (es handelt sich jeweils um ein kurzes Programm, das die entsprechende Leveldatei erzeugt) und das Ergebnis unter den File-Namen »1« bis »9« im Subdirectory »Level« abspeichern. Beachten Sie bitte das Leerzeichen im File-Namen der Levels (jeweils vor der Ziffer).

Schritt 4: Amiga-Basic und den Objekt-Generator laden und starten. Der Objekt-Generator schreibt nun das File »Object« auf die Diskette im internen Laufwerk (für A2000-Besitzer: gemeint ist »df0:« !).

Schritt 5: MAZE laden, starten und Spaß haben.

Wir wünschen Ihnen viel Spaß mit Maze – und denken Sie immer daran: Verlaufen Sie sich nicht.

so

Programmname:	Maze
Computer:	A500, A1000, A2000 mit Kickstart 1.2 oder 1.3
Sprache:	Amiga-Basic
Programmautor: Jrg Isselmann ----- 1 PSO SCREEN 2,320,256,4,1 2 zM WINDOW 2,"The MAZE I created 1988 by J. Isselmann",,0,2 3 W2 CLEAR ,30000& 4 dc GOSUB BuildChess 5 EQ DIM Level(14,9):Level=0 6 gm DIM Back#(3) 7 79 ON TIMER(1) GOSUB Countdown:TIMER OFF 8 pC HauptMenu: 9 gt MENU OFF:COLOR 0,2:CLS:FOR i=1 TO 4:MENU i,0,0,"":NEXT i 10 bg Center\$="Hello,welcome to the famous MAZE.":GOSUB Center:PRINT 11 uS Center\$="What do you want to do?":GOSUB Center:PRINT 12 mV Center\$="I'll play MAZE - FIRE BUTTON":GOSUB Center 13 yt Center\$="I'll edit MAZE - LEFT MOUSE BUTTON":GOSUB Center	

```

14 y2 Center$="I need help - JOYS. IN ANY DIRECTION":GOSUB Center:PRINT
15 6H Center$="I want to quit this silly programm,":GOSUB Center
16 zR Center$="I'll press the ESCape key !":GOSUB Center
17 xe WHILE STRIG(2)<>0 OR MOUSE(0)<>0 OR INKEY$<>"" :WEND
18 SN WHILE STRIG(2)=0
19 OV IF INKEY$=CHR$(27) THEN Quit
20 ws IF MOUSE(0)<>0 THEN GOTO Editor
21 IX IF STICK(2)<>0 OR STICK(3)<>0 THEN GOTO Help
22 WM WEND:WHILE STRIG(2)<>0:WEND
23 IZ Level=0:Score=0:LevelBonus=0
24 wx PlayLoop:RANDOMIZE TIMER
25 qA Level=Level+1
26 M5 COLOR 0,2:CLS
27 gB LINE(5,5)-(304,204),0,bf
28 l7 FOR i=2 TO 4
29 Ro LINE(i,i)-(309-i,209-i),3,b
30 f6 LINE(309-i,i)-(309-i,209-i),1
31 GK LINE-(i,209-i),1
32 xD NEXT i
33 gm GOSUB Loading
34 G6 Center$="Prepare for level"+STR$(Level)+".":LOCATE 28:GOSUB Center
35 mU WHILE STRIG(2)<>0:WEND
36 a3 PutX=StartX*20+5:PutY=StartY*20+5:PutIcon=0
37 lg WHILE STRIG(2)=0
38 5K FOR i=0 TO 1000:NEXT i
39 G3 LINE(PutX,PutY)-(PutX+19,PutY+19),0,bf
40 7M FOR i=0 TO 1000:NEXT i
41 cx GOSUB PutIcon
42 7v WEND
43 81 LINE(0,208)-(320,256),2,bf
44 5J LOCATE 27,30:PRINT "Level"Level;
45 9X LOCATE 27,1:PRINT "Money to get: ";Money
46 ps PRINT "Score: ";Score
47 OF PRINT "Time left: ";Zeit;
48 r2 x=StartX*20+15:y=StartY*20+15:FeldX=StartX:FeldY=StartY
49 oD FOR i=0 TO 3:Key(i)=0:NEXT i
50 xF GET(x,y)-(x+1,y+1),Back#
51 Fv LINE(x,y)-(x+1,y+1),0,bf
52 TZ TIMER ON
53 uG WHILE MOUSE(0)<>0 OR STRIG(2)<>0:WEND
54 RW WHILE MOUSE(0)=0
55 HF IF STRIG(2)<>0 THEN
56 5f2 TIMER OFF
57 8q WHILE STRIG(2)<>0:WEND
58 sX DIM Paused#(75)
59 OX GET(111,103)-(200,112),Paused#
60 Me LINE(111,103)-(200,112),3,bf:LINE(111,103)-(199,111),1,bf
61 WC LOCATE 14,15:PRINT "GAME PAUSED";
62 L4 GOSUB Joystick
63 pI PUT(111,103),Paused#,PSET
64 NC ERASE Paused#:TIMER ON
65 zs0 END IF
66 VJ IF Level((x-5)\20,(y-5)\20)<>19 THEN
67 Ez2 xx=STICK(2)*2:yy=STICK(3)*2:IF xx<>0 THEN yy=0
68 2v0 END IF
69 dV IF x+xx<5 OR x+xx>304 THEN xx=0
70 21 IF y+yy<5 OR y+yy>204 THEN yy=0
71 AV NextFeld=Level((x+xx-5)\20,(y+yy-5)\20)
72 bo IF NextFeld>0 THEN GOSUB LockColor
73 Cm IF Level((x-5)\20,(y-5)\20)=16 THEN
74 nc2 IF (x-5)\20<>(x+xx-5)\20 OR (y-5)\20<>(y+yy-5)\20 THEN
75 LE4 PUT((x-5)\20*20+5,((y-5)\20*20+5),Gitter#,PSET
76 Yz Level((x-5)\20,(y-5)\20)=1
77 Og GET(x,y)-(x+1,y+1),Back#
78 gM LINE(x,y)-(x+1,y+1),0,bf
79 D62 END IF
80 E70 END IF
81 Us IF xx<>0 OR yy<>0 THEN
82 G52 x=x+xx:y=y+yy:FeldX=(x-5)\20:FeldY=(y-5)\20
83 Bm PUT(x-xx,y-yy),Back#,PSET
84 Vn GET(x,y)-(x+1,y+1),Back#
85 nT LINE(x,y)-(x+1,y+1),0,bf
86 KDO END IF
87 XF WEND:TIMER OFF
88 ZV GOTO Suicide
89 tK LockColor:
90 Pf IF NextFeld=1 THEN xx=0:yy=0:RETURN
91 15 IF NextFeld>5 AND NextFeld<10 THEN
92 d22 Key(NextFeld-6)=1:Level((x+xx-5)\20,(y+yy-5)\20)=0

```



```

93 04 PUT((x+xx-5)\20)*20+5,((y+yy-5)\20)*20+5,Floor#,PSET
94 sU RETURN
95 TMO END IF
96 mk IF NextFeld>11 AND NextFeld<16 THEN
97 352 Key(NextFeld-12)=0:Level((x+xx-5)\20,(y+yy-5)\20)=0
98 t9 PUT((x+xx-5)\20)*20+5,((y+yy-5)\20)*20+5,Floor#,PSET
99 xZ RETURN
100 YRO END IF
101 Kx IF NextFeld>1 AND NextFeld<6 THEN
102 yD2 IF Key(NextFeld-2)=1 THEN RETURN ELSE xx=0:yy=0:RETURN
103 bUO END IF
104 HN IF NextFeld=11 THEN
105 oh2 Money=Money-1:Level((x+xx-5)\20,(y+yy-5)\20)=0
106 SO LOCATE 27,14:PRINT Money;
107 ZI PUT((x+xx-5)\20)*20+5,((y+yy-5)\20)*20+5,Floor#,PSET
108 61 RETURN
109 haO END IF
110 IN IF NextFeld=10 THEN
111 FV2 IF Money=0 THEN
112 114 TIMER OFF:GOTO LevelEnd
113 Q92 ELSE
114 xR4 xx=0:yy=0
115 ng2 END IF
116 ohO END IF
117 Yf IF NextFeld=21 THEN
118 yJ2 IF Score>999 THEN
119 X14 Score=Score-1000
120 L1 Level((x+xx-5)\20,(y+yy-5)\20)=0
121 00 LOCATE 28,7:PRINT Score;" ";
122 HX PUT((x+xx-5)\20)*20+5,((y+yy-5)\20)*20+5,Floor#,PSET
123 Lx RETURN
124 bK2 ELSE
125 OQ4 xx=0:yy=0:RETURN
126 yr2 END IF
127 zsO END IF
128 ek IF NextFeld=20 THEN
129 1X2 IF INT(2*RND(1))=0 THEN NextFeld=18 ELSE NextFeld=17
130 2vO END IF
131 CO IF NextFeld=17 THEN
132 902 Zeit=Zeit+INT(40*RND(1))+21
133 YE Level((x+xx-5)\20,(y+yy-5)\20)=0
134 yn LOCATE 29,11:PRINT Zeit;
135 Uk PUT((x+xx-5)\20)*20+5,((y+yy-5)\20)*20+5,Floor#,PSET
136 YA RETURN
137 920 END IF
138 Ob IF NextFeld=18 THEN
139 1D2 Score=Score+1000
140 fL Level((x+xx-5)\20,(y+yy-5)\20)=0
141 Ds LOCATE 28,7:PRINT Score;
142 br PUT((x+xx-5)\20)*20+5,((y+yy-5)\20)*20+5,Floor#,PSET
143 fH RETURN
144 G9O END IF
145 hJ RETURN
146 bs LevelEnd:CLS:PRINT
147 QC TimeBonus=INT(Zeit/10)*10
148 kh Center$="Your score is:"+STR$(Score):GOSUB Center:PRINT
149 X5 Center$="Bonus for "+STR$(Zeit)+" seconds left:"+STR$(TimeBo-
nus):GOSUB Center:PRINT
150 L8 Score=Score+TimeBonus
151 RB Center$="Your total score is:"+STR$(Score):GOSUB Center:PRI-
NT:PRINT:PRINT
152 8n Center$="Please press the joystick button.":GOSUB Center
153 s1 GOSUB Joystick:CLS
154 ZW RETURN PlayLoop
155 3p TimeOut:LINE(0,208)-(320,256),2,bf:LOCATE 27
156 5M Center$="Sorry,your time is out.":GOSUB Center
157 11 Center$="Your score is:"+STR$(Score):GOSUB Center
158 Et Center$="Please press the joystick button.":GOSUB Center
159 ud GOSUB Joystick
160 JU RETURN HauptMenue
161 WB Suicide:LINE(0,208)-(320,256),2,bf:LOCATE 27
162 Be Center$="You've comitted suicide.":GOSUB Center
163 77 Center$="Your score is:"+STR$(Score):GOSUB Center
164 Kz Center$="Please press the joystick button.":GOSUB Center
165 OJ GOSUB Joystick
166 D9 GOTO HauptMenue
167 1d GameOver:LINE(0,208)-(320,256),2,bf:LOCATE 5
168 Wp COLOR 2,0
169 iq Center$="Congratulations !!!":GOSUB Center
170 bV Center$="You've completed all":GOSUB Center
171 KV Center$="the "+STR$(Level-1)+" levels of MAZE !":GOSUB Cente
r

```

```

172 GG Center$="Your score is:"+STR$(Score):GOSUB Center
173 Mv FOR i=1 TO Level-1:LevelBonus=LevelBonus+i*1000:NEXT i
174 ZC Center$="Bonus for completing"+STR$(Level-1)+" levels:"+STR
$(LevelBonus):GOSUB Center
175 OC Score=Score+LevelBonus
176 NR Center$="Your total score is:"+STR$(Score):GOSUB Center
177 XC Center$="Please press the joystick button.":GOSUB Center
178 k3 COLOR 0,2
179 Ex GOSUB Joystick
180 RN GOTO HauptMenue
181 cd Quit:
182 7h TIMER OFF
183 JO CLS:PRINT"OK ! BYE BYE !"
184 3P ERASE Level
185 nw ERASE Floor#,Wall#
186 hU ERASE DoorR#,DoorG#,DoorB#,DoorY#
187 aM ERASE KeyR#,KeyY#,KeyG#,KeyB#
188 5s ERASE KeyExR#,KeyExY#,KeyExG#,KeyExB#
189 IW ERASE Goal#,Money#
190 aR ERASE FallGitter#,Gitter#
191 mB ERASE MoreTime#,Gold#,Ice#,Unknown#
192 sb ERASE Sentinel#
193 9P ERASE Back#
194 KO WINDOW CLOSE 2
195 06 SCREEN CLOSE 2
196 o2 LIBRARY CLOSE
197 vq END
198 Tf Joystick:
199 yn WHILE STRIG(2)<>0:WEND:WHILE STRIG(2)=0:WEND
200 R9 WHILE STRIG(2)<>0:WEND
201 bD RETURN
202 YM Maus:
203 Gh WHILE MOUSE(0)<>0:WEND:WHILE MOUSE(0)=0:WEND
204 NK WHILE MOUSE(0)<>0:WEND:RETURN
205 By PutIcon:
206 KV IF PutIcon=0 THEN PUT(PutX,PutY),Floor#,PSET:RETURN
207 Rg IF PutIcon=1 THEN PUT(PutX,PutY),Wall#,PSET:RETURN
208 fB IF PutIcon=2 THEN PUT(PutX,PutY),DoorR#,PSET:RETURN
209 1N IF PutIcon=3 THEN PUT(PutX,PutY),DoorG#,PSET:RETURN
210 13 IF PutIcon=4 THEN PUT(PutX,PutY),DoorB#,PSET:RETURN
211 Jz IF PutIcon=5 THEN PUT(PutX,PutY),DoorY#,PSET:RETURN
212 LO IF PutIcon=6 THEN PUT(PutX,PutY),KeyR#,PSET:RETURN
213 sN IF PutIcon=7 THEN PUT(PutX,PutY),KeyG#,PSET:RETURN
214 h8 IF PutIcon=8 THEN PUT(PutX,PutY),KeyB#,PSET:RETURN
215 sh IF PutIcon=9 THEN PUT(PutX,PutY),KeyY#,PSET:RETURN
216 al IF PutIcon=10 THEN PUT(PutX,PutY),Goal#,PSET:RETURN
217 qZ IF PutIcon=11 THEN PUT(PutX,PutY),Money#,PSET:RETURN
218 Gv IF PutIcon=12 THEN PUT(PutX,PutY),KeyExR#,PSET:RETURN
219 Hf IF PutIcon=13 THEN PUT(PutX,PutY),KeyExG#,PSET:RETURN
220 sb IF PutIcon=14 THEN PUT(PutX,PutY),KeyExB#,PSET:RETURN
221 BR IF PutIcon=15 THEN PUT(PutX,PutY),KeyExY#,PSET:RETURN
222 L5 IF PutIcon=16 THEN PUT(PutX,PutY),FallGitter#,PSET:RETURN
223 SU IF PutIcon=17 THEN PUT(PutX,PutY),MoreTime#,PSET:RETURN
224 Fb IF PutIcon=18 THEN PUT(PutX,PutY),Gold#,PSET:RETURN
225 gO IF PutIcon=19 THEN PUT(PutX,PutY),Ice#,PSET:RETURN
226 4S IF PutIcon=20 THEN PUT(PutX,PutY),Unknown#,PSET:RETURN
227 T3 IF PutIcon=21 THEN PUT(PutX,PutY),Sentinel#,PSET:RETURN
228 2e RETURN
229 xq Countdown:
230 FM Zeit=Zeit-1:LOCATE 29,11:PRINT Zeit;" ";IF Zeit=0 THEN TI-
MER OFF:GOTO TimeOut
231 5h RETURN
232 Mb Loading:
233 MX a$=STR$(Level)
234 VY OPEN"df0:Level/"+a$ FOR APPEND AS #1
235 gN IF LOF(1)=0 THEN CLOSE #1:KILL"df0:Level/"+a$:RETURN GameO-
ver
236 2p CLOSE #1
237 z2 OPEN"df0:Level/"+a$ FOR INPUT AS #1
238 90 FOR y=0 TO 9:FOR x=0 TO 14
239 Ep Level(x,y)=ASC(INPUT$(1,#1))
240 Pa PutIcon=Level(x,y):PutX=x*20+5:PutY=y*20+5:GOSUB PutIcon
241 JW NEXT x:NEXT y:StartX=ASC(INPUT$(1,#1)):StartY=ASC(INPUT$(1,
#1))
242 CP Money=ASC(INPUT$(1,#1))
243 kq Zeit=CVI(INPUT$(2,#1))
244 Ax CLOSE #1
245 Jv RETURN
246 EP BuildChess:
247 gQ OPEN"df0:Object" FOR INPUT AS #1

```

Listing 1. Maze, das Hauptprogramm


```

248 Q2 FOR i=0 TO 15
249 Ry PALETTE 1,ASC(INPUT$(1,1))/16,ASC(INPUT$(1,1))/16,ASC(INPUT
$(1,1))/16
250 TJ NEXT i
251 qk i=0
252 vk DIM Loading#(19)
253 WW WHILE NOT EOF(1)
254 E5 FOR o=0 TO 19:Loading#(o)=CVD(INPUT$(8,1)):NEXT o
255 oM PUT(0,1),Loading#:i=i+1
256 WI WEND:ERASE Loading#
257 NA CLOSE #1
258 CU DIM Floor#(50)
259 TB GET(0,0)-(19,19),Floor#
260 IU DIM Wall#(50)
261 aV GET(20,0)-(39,19),Wall#
262 mm DIM DoorR#(50)
263 Ox GET(40,0)-(59,19),DoorR#
264 NU DIM DoorY#(50)
265 Ak GET(60,0)-(79,19),DoorY#
266 xm DIM DoorG#(50)
267 KW GET(80,0)-(99,19),DoorG#
268 aK DIM DoorB#(50)
269 k3 GET(100,0)-(119,19),DoorB#
270 NC DIM KeyR#(50)
271 id GET(120,0)-(139,19),KeyR#
272 rn DIM KeyY#(50)
273 Xd GET(140,0)-(159,19),KeyY#
274 jN DIM KeyG#(50)
275 LD GET(160,0)-(179,19),KeyG#
276 RO DIM KeyB#(50)
277 C3 GET(180,0)-(199,19),KeyB#
278 YH DIM KeyExR#(50)
279 ig GET(200,0)-(219,19),KeyExR#
280 GX DIM KeyExY#(50)
281 CL GET(220,0)-(239,19),KeyExY#
282 Yz DIM KeyExG#(50)
283 sn GET(240,0)-(259,19),KeyExG#
284 6r DIM KeyExB#(50)
285 ys GET(260,0)-(279,19),KeyExB#
286 5M DIM Goal#(50)
287 iu GET(280,0)-(299,19),Goal#
288 K9 DIM Money#(50)
289 AN GET(0,20)-(19,39),Money#
290 SD DIM FallGitter#(50)
291 Bs GET(20,20)-(39,39),FallGitter#
292 sE DIM Gitter#(50)
293 3y GET(40,20)-(59,39),Gitter#
294 lV DIM MoreTime#(50)
295 ni GET(60,20)-(79,39),MoreTime#
296 Ga DIM Gold#(50)
297 Bn GET(80,20)-(99,39),Gold#
298 9t DIM Ice#(50)
299 5O GET(100,20)-(119,39),Ice#
300 hW DIM Unknown#(50)
301 RO GET(120,20)-(139,39),Unknown#
302 RY DIM Sentinel#(50)
303 qe GET(140,20)-(159,39),Sentinel#
304 Gs RETURN
305 mM Center:
306 08 POKEW WINDOW(8)+36,WINDOW(2)/2-LEN(Center$)*4
307 Ky PRINT Center$
308 Kw RETURN
309 2J Help:CLS
310 LB PUT(0,0),Floor#,PSET
311 sv LOCATE 1,4:PRINT"On this fields can you go."
312 82 PUT(0,20),Wall#,PSET
313 XX LOCATE 4,4:PRINT"This is a wall,isn't it?"
314 gu PUT(0,40),DoorR#,PSET
315 tN LOCATE 6,4:PRINT"You can only open such a door"
316 GJ LOCATE 7,4:PRINT"with a same-coloured key."
317 WJ PUT(0,60),KeyY#,PSET
318 JT LOCATE 9,4:PRINT"This is the key for the yellow doors."
319 wf PUT(0,80),KeyExG#,PSET
320 v2 LOCATE 11,4:PRINT"This icon grabbs all the green keys"
321 G1 LOCATE 12,4:PRINT"from your inventory."
322 C1 PUT(0,100),Money#,PSET
323 V8 LOCATE 14,4:PRINT"This is a piece of gold named money."
324 X4 PUT(0,120),Goal#,PSET
325 wA LOCATE 16,4:PRINT"You can only use this exit if you"
326 ex LOCATE 17,4:PRINT"have collected all the money."
327 FO PUT(0,140),FallGitter#,PSET

```

```

328 um LOCATE 19,4:PRINT"Attention.This is a special gate."
329 72 LOCATE 20,4:PRINT"If you pass it,it'll close."
330 m7 PUT(0,160),MoreTime#,PSET
331 TG LOCATE 21,4:PRINT"This magic object makes the time"
332 LE LOCATE 22,4:PRINT"grow a little bit."
333 BV PUT(0,180),Gold#,PSET
334 vI LOCATE 24,4:PRINT"This sack of money is made for"
335 IQ LOCATE 25,4:PRINT"you to get rich."
336 6y PUT(0,200),Ice#,PSET
337 xG LOCATE 26,4:PRINT"This is ice.It's horrible."
338 ok LOCATE 27,4:PRINT"You can't control your moves on it."
339 S3 PUT(0,220),Sentinel#,PSET
340 Ks LOCATE 29,4:PRINT"This is a sentinel.He disappears if"
341 XW LOCATE 30,4:PRINT"you give him 1000 points of score.";
342 v4 GOSUB Joystick:CLS
343 OZ Center$="Move your little man up,down,left":GOSUB Center
344 kR Center$="and right with the joystick.":GOSUB Center
345 rq Center$="You can stop the game any time by":GOSUB Center
346 fB Center$="pressing the fire button.":GOSUB Center
347 6R Center$="Sometimes you're hindered in doing":GOSUB Center
348 ZG Center$="something,because your silly moves.":GOSUB Center
349 gf Center$="Then you can wait till the time is":GOSUB Center
350 id Center$="over,but you can press the suicide-":GOSUB Center
351 rz Center$="button (left mouse button),too.":GOSUB Center
352 1k GOSUB Joystick
353 EA GOTO HauptMenue
354 E6 Editor:
355 V9 COLOR 0,2:CLS:LINE(5,5)-(304,204),0,bf
356 3P FOR i=2 TO 4
357 j6 LINE(i,1)-(309-i,209-i),3,b
358 xO LINE(309-i,1)-(309-i,209-i),1
359 Yc LINE-(i,209-i),1
360 FV NEXT i
361 8G GOSUB NewLevel
362 SW FOR i=1 TO 4:MENU i,0,0,"":NEXT i
363 Bo MENU 1,0,1,"Project":MENU 2,0,1,"Do"
364 7I MENU 1,1,1,"Load ":MENU 2,1,1,"Set Startpoint"
365 L8 MENU 1,2,1,"Save ":MENU 2,2,1,"Set Time "
366 jm MENU 1,3,1,"Kill ":MENU 2,3,1,"Fill "
367 O5 MENU 1,4,1,"NEW "
368 2B MENU 1,5,0,""
369 cO MENU 1,6,1,"RETURN "
370 DA ON MENU GOSUB Menue:MENU ON
371 Y1 EditorSchleife:
372 zQ WHILE MOUSE(0)<>0:WEND:WHILE MOUSE(0)=0:WEND
373 Kb WHILE MOUSE(0)<>0:x=MOUSE(1):y=MOUSE(2)
374 AC IF y>209 AND y<230 THEN
375 R2z GOSUB NewTool
376 fO0 ELSE
377 ZL2 x=(x-5)\20:y=(y-5)\20
378 wQ IF x>0 AND x<=14 AND y>0 AND y<=9 THEN
379 C24 IF x<>StartX OR y<>StartY OR PointSet=0 THEN
380 wR6 IF Level(x,y)=11 THEN Money=Money-1
381 Ph IF Level(x,y)=10 THEN ExitSet=0
382 l1 IF Tool=11 THEN Money=Money+1
383 bN IF Tool=10 THEN ExitSet=1
384 N4 PutX=x*20+5:PutY=y*20+5:PutIcon=Tool
385 Bu Level(x,y)=Tool:GOSUB PutIcon
386 A34 END IF
387 B42 END IF
388 C50 END IF
389 lW WEND
390 19 GOTO EditorSchleife
391 h1 Menue:MENU OFF
392 fP Men0=MENU(0):Men1=MENU(1)
393 3g IF Men0=1 THEN
394 7L2 IF Men1=1 THEN
395 oR4 LINE(0,208)-(320,256),2,bf
396 xv LOCATE 27,1
397 Yy INPUT"Please input the levelnumber: ",Level:a$=STR$(Leve
l)
398 9C OPEN"df0:Level/"+a$ FOR APPEND AS #1
399 6R IF LOF(1)=0 THEN
400 gT6 CLOSE #1
401 vp KILL"df0:Level/"+a$
402 m7 PRINT"Level doesn't exist.Click mouse.";
403 9o GOSUB Maus
404 7q4 ELSE
405 lY6 CLOSE #1
406 hn GOSUB Loading
407 Vy PointSet=1:TimeSet=1:ExitSet=1
408 WP4 END IF

```



```

409 vo      GOSUB BuildMenue
410 YR2     END IF
411 Tx      IF Men1=2 THEN GOSUB Saving
412 wG      IF Men1=3 THEN GOSUB Killing
413 yn      IF Men1=4 THEN GOSUB NewLevel
414 N2      IF Men1=6 THEN GOTO HauptMenue
415 I10     ELSE
416 5r2     IF Men1=1 THEN GOSUB SetPoint
417 DD      IF Men1=2 THEN GOSUB SetTime
418 5k      IF Men1=3 THEN GOSUB Fill
419 ha0     END IF
420 OQ      MENU ON
421 91      RETURN
422 sh      BuildMenue:Col=0
423 tS      LINE(0,208)-(320,256),2,bf:LINE(280,210)-(299,229),6,bf
424 Bo      PUT(0,210),Floor#,PSET:PUT(20,210),Wall#,PSET
425 Mg      PUT(40,210),DoorR#,PSET:PUT(60,210),KeyR#,PSET
426 YI      PUT(80,210),Goal#,PSET:PUT(100,210),Money#,PSET
427 Qn      PUT(120,210),KeyExR#,PSET
428 2S      PUT(140,210),FallGitter#,PSET
429 Nr      PUT(160,210),MoreTime#,PSET
430 yM      PUT(180,210),Gold#,PSET
431 9h      PUT(200,210),Ice#,PSET
432 bn      PUT(220,210),Unknown#,PSET
433 n9      PUT(240,210),Sentinel#,PSET
434 Ow      GOSUB NewTool
435 Nz      RETURN
436 D8      NewTool:
437 Hg      xx=x\20
438 sY      IF xx<13 THEN
439 fT2     IF xx<2 THEN Tool=xx
440 TC      IF xx=2 THEN Tool=2+Col
441 rf      IF xx=3 THEN Tool=6+Col
442 9V      IF xx=4 THEN Tool=10
443 Jc      IF xx=5 THEN Tool=11
444 kF      IF xx=6 THEN Tool=12+Col
445 lC      IF xx>6 AND xx<12 THEN Tool=9+xx
446 ua      IF xx=12 THEN Tool=21
447 xJ      LINE(0,231)-(320,231),2
448 7u      LINE(xx*20,231)-(xx*20+19,231),0
449 B40     END IF
450 As      IF xx=14 THEN
451 IC2     Col=Col+1:IF Col>3 THEN Col=0
452 4x      IF Col=0 THEN Col1=6
453 L7      IF Col=1 THEN Col1=8
454 WF      IF Col=2 THEN Col1=9
455 PH      IF Col=3 THEN Col1=7
456 Tk      LINE(280,210)-(299,229),Col1,bf
457 WK      IF Tool>1 AND Tool<6 THEN Tool=2+Col
458 Nv      IF Tool>5 AND Tool<10 THEN Tool=6+Col
459 u2      IF Tool>11 AND Tool<16 THEN Tool=12+Col
460 2u      WHILE MOUSE(0)<>0:WEND
461 NG0     END IF
462 oQ      RETURN
463 aF      Saving:LINE(0,208)-(320,256),2,bf:LOCATE 27,1
464 jY      IF PointSet=0 THEN
465 qe2     PRINT"No startpoint set.Click mouse."
466 gO      GOSUB Maus:GOSUB BuildMenue:RETURN
467 TMO     END IF
468 oN      IF ExitSet=0 THEN
469 4y2     PRINT"No exit set.Click mouse."
470 k4      GOSUB Maus:GOSUB BuildMenue:RETURN
471 XQ0     END IF
472 2d      IF TimeSet=0 THEN
473 Kq2     PRINT"No time set.Click mouse."
474 o8      GOSUB Maus:GOSUB BuildMenue:RETURN
475 bu0     END IF
476 QX      LINE INPUT"Please input the levelnumber:","a$:a$=STR$(VAL(a$
477 xV      OPEN"df0:Level/"+"a$ FOR OUTPUT AS #1
478 lS      FOR y=0 TO 9:FOR x=0 TO 14
479 vZ      PRINT #1,CHR$(Level(x,y));
480 lD      NEXT x:NEXT y
481 aO      PRINT #1,CHR$(StartX);CHR$(StartY);CHR$(Money);
482 6C      PRINT #1,MKT$(Zeit);
483 lO      CLOSE #1
484 Ez      PRINT"Level N"a$ saved.Click mouse.";
485 W1      GOSUB Maus:GOSUB BuildMenue
486 Co      RETURN
487 zD      Killing:LINE(0,208)-(320,256),2,bf:LOCATE 27,1
488 eJ      LINE INPUT"Please input the levelnumber:","a$:a$=STR$(VAL(a$
  ))

```

```

489 cf      OPEN"df0:Level/"+"a$ FOR APPEND AS #1
490 Zu      IF LOF(1)=0 THEN
491 9w2     CLOSE #1
492 OI      KILL"df0:Level/"+"a$
493 Fa      PRINT"Level doesn't exist.Click mouse.";
494 8S      GOSUB Maus:GOSUB BuildMenue:RETURN
495 aJ0     ELSE
496 E12     CLOSE #1
497 TN      KILL"df0:Level/"+"a$
498 93      PRINT"Level N"a$ killed.Click Mouse.";
499 kz      GOSUB Maus:GOSUB BuildMenue
500 Ot0     END IF
501 R3      RETURN
502 my      NewLevel:LINE(0,208)-(320,256),2,bf
503 L1      ExitSet=0:TimeSet=0:Tool=0:PointSet=0:Money=0:Col=0:GOSUB F
  ill
504 SL      GOSUB BuildMenue
505 V7      RETURN
506 Ok      SetPoint:LINE(0,208)-(320,256),2,bf:LOCATE 27,1
507 MI      PRINT"Please click mouse at startpoint."
508 oW      GOSUB Maus:x=MOUSE(1):y=MOUSE(2):x=(x-5)\20:y=(y-5)\20
509 pA      IF x<0 OR x>14 OR y<0 OR y>9 THEN BEEP:GOTO SetPoint
510 OU      IF Level(x,y)<>0 THEN
511 P22     PRINT"You can only start on the floor."
512 eN      PRINT"Click mouse.";GOSUB Maus:GOSUB BuildMenue:RETURN
513 D60     END IF
514 ki      StartX=x:StartY=y:PointSet=1:PRINT"OK ! Click mouse.";
515 OF      GOSUB Maus:GOSUB BuildMenue
516 gI      RETURN
517 zy      SetTime:LINE(0,208)-(320,256),2,bf:LOCATE 27,1
518 pN      INPUT"How much time ";Zeit:IF Zeit<1 OR Zeit>2000 THEN Se
  tTime
519 hx      PRINT"OK ! Click mouse.":TimeSet=1
520 5K      GOSUB Maus:GOSUB BuildMenue
521 lN      RETURN
522 Bb      Fill:LOCATE 28,1:PointSet=0:TimeSet=0:ExitSet=0
523 kb      FOR y=0 TO 9:FOR x=0 TO 14
524 IG      Level(x,y)=Tool
525 T9      PutX=20*x+5:PutY=20*y+5
526 3u      PutIcon=Tool:GOSUB PutIcon
527 TO      NEXT x:NEXT y
528 wI      IF Tool=10 THEN ExitSet=1
529 uB      IF Tool=11 THEN Money=150 ELSE Money=0
530 uW      RETURN
  (c) 1990 M&T

```

Listing 1. (Schluß)

Programmname:	Objekt-Generator
Computer:	A500, A1000, A2000 mit Kickstart 1.2 oder 1.3
Sprache:	Amiga-Basic
Programmator: Jrg Isselmann ----- 1 PS0 SCREEN 2,320,256,4,1 2 HA WINDOW 2,"Object Generator",,0,2 3 pP LINE(0,0)-(19,19),3,bf 4 mM LINE(1,1)-(19,19),1,bf 5 mL LINE(1,1)-(18,18),2,bf 6 ZM FOR y=0 TO 19:FOR x=20 TO 39 7 ON READ d:PSET(x,y),d:NEXT x:NEXT y 8 lO DATA 13,11,11,11,11,11,11,11,11,11,13,11,11,11,11,11,11,11,11,11 9 f1 DATA 13,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,12 10 gJ DATA 13,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,12 11 hK DATA 13,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,12 12 sz DATA 13,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,14 13 lK DATA 11,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,12 14 j1 DATA 11,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,12 15 kM DATA 11,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,12 16 lN DATA 11,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,12 17 x4 DATA 13,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,14 18 oR DATA 13,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,12 19 pS DATA 13,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,12	

Listing 2. Der Object-Generator wird nur einmal gebraucht, aber ohne ihn geht es leider nicht


```

20 qt DATA 13,4,4,4,4,4,4,4,4,4,5,4,4,4,4,4,4,4,12
21 ru DATA 13,4,4,4,4,4,4,4,4,4,4,5,4,4,4,4,4,4,12
22 np DATA 13,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,14
23 su DATA 11,4,4,4,4,5,4,4,4,4,4,4,4,4,4,4,4,12
24 tv DATA 11,4,4,4,4,5,4,4,4,4,4,4,4,4,4,4,5,12
25 uw DATA 11,4,4,4,4,5,4,4,4,4,4,4,4,4,4,4,5,12
26 vx DATA 11,4,4,4,4,5,4,4,4,4,4,4,4,4,4,4,5,12
27 hx DATA 14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14
28 sw FOR y=0 TO 19:FOR x=40 TO 59
29 mj READ d:PSET(x,y),d:NEXT x:NEXT y
30 ey DATA 13,11,11,11,11,11,11,11,11,11,13,11,11,11,11,11,11,11
31 la DATA 13,4,4,4,4,4,4,4,4,4,4,4,5,4,4,4,4,4,12
32 ts DATA 13,4,4,4,4,4,4,4,4,4,4,4,5,4,4,4,4,4,12
33 qe DATA 13,4,4,4,4,4,4,4,4,4,4,4,5,4,4,4,4,4,12
34 el DATA 13,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,14
35 tm DATA 11,4,4,4,4,5,4,5,6,6,6,6,6,6,4,5,4,5,12
36 wx DATA 11,4,4,4,5,5,5,6,6,6,6,6,6,6,4,5,4,4,12
37 ii DATA 11,4,4,4,5,4,6,6,6,6,6,6,6,6,6,5,5,4,12
38 gg DATA 11,4,4,5,4,4,6,6,6,6,6,6,6,6,6,6,4,5,12
39 oy DATA 13,5,5,5,4,6,6,6,6,6,6,6,6,6,6,6,5,5,14
40 hd DATA 13,4,4,5,5,6,6,6,6,6,6,6,6,6,6,6,5,4,12
41 dz DATA 13,4,4,5,4,6,6,6,6,6,6,6,6,6,6,6,4,5,12
42 ae DATA 13,4,4,5,4,6,6,6,6,6,6,6,6,6,6,6,4,5,12
43 tt DATA 13,4,4,5,5,6,0,0,6,6,6,6,6,6,6,5,4,12
44 pt DATA 13,5,5,5,4,6,0,6,6,6,6,6,6,6,6,6,4,5,14
45 zs DATA 11,4,4,4,5,4,6,0,6,6,6,6,6,6,6,6,4,5,12
46 hg DATA 11,4,4,5,5,6,6,6,6,6,6,6,6,6,6,6,5,4,12
47 zb DATA 11,4,4,5,4,6,6,6,6,6,6,6,6,6,6,6,4,5,12
48 ac DATA 11,4,4,5,4,6,6,6,6,6,6,6,6,6,6,6,4,4,12
49 xd DATA 14,14,14,14,14,14,6,6,6,6,6,6,6,6,6,6,14,14,14,14
50 ap GetX=2:GetY=0:PutX=3:PutY=0:GOSUB MoveTo
51 cj GetX=3:Which=6:ToWhich=7:GOSUB Switch
52 ec GetX=3:PutX=4:GOSUB MoveTo
53 qo GetX=4:Which=7:ToWhich=8:GOSUB Switch
54 km GetX=4:PutX=5:GOSUB MoveTo
55 eh GetX=5:Which=8:ToWhich=9:GOSUB Switch
56 ni FOR y=0 TO 19:FOR x=120 TO 139
57 ob READ d:PSET(x,y),d:NEXT x:NEXT y
58 ye DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
59 ka DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3
60 lb DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3
61 wc DATA 1,2,2,2,2,6,6,6,6,6,6,6,6,6,6,6,2,2,3
62 bi DATA 1,2,2,2,6,2,0,0,0,0,0,0,0,0,0,6,2,2,3
63 ey DATA 1,2,2,6,0,2,2,2,2,2,2,2,2,2,2,2,6,0,2,3
64 fz DATA 1,2,2,2,6,0,2,2,2,2,2,2,2,2,2,2,6,0,2,3
65 go DATA 1,2,2,2,6,0,2,2,2,2,2,2,2,2,2,2,6,0,2,3
66 t7 DATA 1,2,2,2,2,6,6,6,6,6,6,6,6,6,6,6,2,0,2,3
67 qi DATA 1,2,2,2,2,2,0,0,0,6,6,0,0,0,0,6,2,2,3
68 na DATA 1,2,2,2,2,2,2,2,2,2,6,6,0,2,2,2,2,2,3
69 oa DATA 1,2,2,2,2,2,2,2,2,6,6,0,2,2,2,2,2,2,3
70 n3 DATA 1,2,2,2,2,2,2,2,2,6,6,6,6,6,2,2,2,2,3
71 oo DATA 1,2,2,2,2,2,2,2,2,6,6,0,0,0,0,2,2,2,3
72 ps DATA 1,2,2,2,2,2,2,2,2,6,6,6,6,6,2,2,2,2,3
73 qq DATA 1,2,2,2,2,2,2,2,2,6,6,0,0,0,0,2,2,2,3
74 r7 DATA 1,2,2,2,2,2,2,2,2,6,6,6,6,6,2,2,2,2,3
75 wc DATA 1,2,2,2,2,2,2,2,2,2,0,0,0,0,0,2,2,2,3
76 br DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3
77 r7 DATA 3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3
78 ko GetX=6:PutX=7:GOSUB MoveTo
79 mn GetX=7:Which=6:ToWhich=7:GOSUB Switch
80 sa GetX=7:PutX=8:GOSUB MoveTo
81 oe GetX=8:Which=7:ToWhich=8:GOSUB Switch
82 ak GetX=8:PutX=9:GOSUB MoveTo
83 ev GetX=9:Which=8:ToWhich=9:GOSUB Switch
84 sy FOR y=0 TO 19:FOR x=200 TO 219
85 gd READ d:PSET(x,y),d:NEXT x:NEXT y
86 qg DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
87 me DATA 1,2,2,2,2,6,6,6,6,6,6,6,6,6,6,6,2,2,3
88 lr DATA 1,2,2,2,6,2,0,0,0,0,0,0,0,0,0,6,2,2,3
89 ao DATA 1,2,2,2,6,0,2,2,2,2,2,2,2,2,2,6,0,2,3
90 sp DATA 1,2,2,2,6,0,2,2,2,2,2,2,2,2,2,6,0,2,3
91 bq DATA 1,2,2,2,6,0,2,2,2,2,2,2,2,2,2,6,0,2,3
92 jx DATA 1,2,2,2,6,6,6,6,6,6,6,6,6,6,6,2,0,2,3
93 gi DATA 1,2,2,2,2,2,0,0,0,6,6,0,0,0,0,2,2,2,3
94 dz DATA 1,2,2,2,2,2,2,2,2,6,6,0,2,2,2,2,2,2,3
95 iu DATA 1,2,2,2,2,2,2,2,2,2,0,0,2,2,2,2,2,2,3
96 tr DATA 1,2,2,2,2,2,2,2,2,6,2,2,2,2,2,2,2,2,3
97 qy DATA 1,2,2,2,2,2,6,6,6,2,2,2,2,2,2,2,2,2,3
98 zd DATA 1,2,2,2,2,2,6,6,6,0,0,2,2,2,2,2,2,2,3

```

```

99  YY DATA 1,2,2,2,2,6,6,6,0,0,2,2,2,2,2,2,2,3
100 hL DATA 1,2,2,2,6,6,6,0,6,2,2,2,2,2,2,2,2,3
101 io DATA 1,2,2,2,2,6,0,6,2,6,2,2,2,2,2,2,2,3
102 P1 DATA 1,2,2,2,2,6,2,6,2,0,2,2,2,2,2,2,2,3
103 GE DATA 1,2,2,2,2,2,2,6,2,0,2,2,2,2,2,2,2,3
104 xB DATA 1,2,2,2,2,2,2,2,0,2,2,2,2,2,2,2,2,3
105 JZ DATA 3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3
106 zh GetX=10:PutX=11:GOSUB MoveTo
107 xo GetX=11:Which=6:ToWhich=7:GOSUB Switch
108 Ap GetX=11:PutX=12:GOSUB MoveTo
109 Ey GetX=12:Which=7:ToWhich=8:GOSUB Switch
110 Lx GetX=12:PutX=13:GOSUB MoveTo
111 V8 GetX=13:Which=8:ToWhich=9:GOSUB Switch
112 Z1 FOR y=0 TO 19:FOR x=280 TO 299
113 i5 READ d:PSET(x,y),d:NEXT x:NEXT y
114 OK DATA 13,11,11,11,11,11,11,11,11,11,13,11,11,11,11,11,11,
    11,11
115 NQ DATA 13,4,4,4,4,4,4,4,4,4,5,4,4,4,4,4,4,12
116 OR DATA 13,4,4,4,4,4,4,4,4,4,5,4,4,4,4,4,4,12
117 PS DATA 13,4,4,4,4,4,4,4,4,4,5,4,4,4,4,4,4,12
118 ah DATA 13,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,14
119 aa DATA 11,4,4,4,4,5,4,4,5,4,5,4,5,4,5,4,5,12
120 JZ DATA 11,4,4,4,5,5,5,11,14,11,14,5,5,4,4,5,4,4,12
121 sS DATA 11,4,4,4,5,4,14,14,14,11,14,6,14,5,5,5,4,4,12
122 tX DATA 11,4,4,5,4,4,14,11,14,14,14,6,14,6,4,4,5,4,12
123 Pa DATA 13,5,5,5,4,14,14,11,14,11,14,14,14,6,5,4,5,5,14
124 LO DATA 13,4,4,5,5,11,14,14,14,11,14,6,14,14,15,5,4,4,12
125 JB DATA 13,4,4,5,4,11,14,11,14,14,14,6,14,15,15,4,5,4,12
126 Dx DATA 13,4,4,5,4,14,14,11,14,6,14,14,15,15,15,4,5,4,12
127 41 DATA 13,4,4,5,5,11,14,14,14,6,14,1,1,1,1,5,4,4,12
128 26 DATA 13,5,5,5,4,11,14,11,14,14,1,1,1,1,4,5,5,14
129 X8 DATA 11,4,4,5,4,14,14,11,14,2,2,2,2,2,2,4,5,4,12
130 7J DATA 11,4,4,5,5,11,14,14,2,2,2,2,2,2,2,5,4,4,12
131 cZ DATA 11,4,4,5,4,11,14,2,2,2,2,2,2,2,2,4,5,4,12
132 CC DATA 11,4,4,5,4,4,3,3,3,3,3,3,3,3,4,5,4,12
133 Nd DATA 14,14,14,14,14,3,3,3,3,3,3,3,3,14,14,14,14
134 mN FOR y=20 TO 39:FOR x=0 TO 19
135 4R READ d:PSET(x,y),d:NEXT x:NEXT y
136 EU DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
137 aq DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3
138 br DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3
139 es DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3
140 TJ DATA 1,2,2,2,2,2,2,2,10,10,10,10,2,2,2,2,2,2,3
141 hx DATA 1,2,2,2,2,2,10,7,7,7,7,7,10,2,2,2,2,2,3
142 Hh DATA 1,2,2,2,2,10,7,7,7,7,7,7,7,10,2,2,2,2,3
143 rs DATA 1,2,2,2,10,7,7,7,7,10,7,7,7,7,10,2,2,2,3
144 i8 DATA 1,2,2,2,10,7,7,7,10,10,7,7,7,7,10,2,2,2,3
145 yO DATA 1,2,2,2,10,7,7,10,7,10,7,7,7,7,10,2,2,2,3
146 uK DATA 1,2,2,2,10,7,7,7,10,7,7,7,7,10,2,2,2,3
147 vL DATA 1,2,2,2,10,7,7,7,7,10,7,7,7,7,10,2,2,2,3
148 I2 DATA 1,2,2,2,2,10,7,7,7,7,7,7,7,10,7,10,2,2,3
149 zZ DATA 1,2,2,2,2,10,7,7,7,7,7,7,7,10,2,2,2,3
150 bF DATA 1,2,2,2,2,2,2,10,10,10,10,10,10,7,10,2,2,2,3
151 eu DATA 1,2,2,2,2,2,2,10,10,10,10,10,10,2,2,2,2,3
152 p5 DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3
153 q6 DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3
154 r7 DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3
155 7N DATA 3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3
156 sO FOR y=20 TO 39:FOR x=20 TO 39
157 Qn READ d:PSET(x,y),d:NEXT x:NEXT y
158 i2 DATA 13,11,11,11,11,11,11,11,11,11,13,11,11,11,11,11,11,
    11,11
159 58 DATA 13,4,4,4,4,4,4,4,4,4,5,4,4,4,4,4,4,12
160 69 DATA 13,4,4,4,4,4,4,4,4,4,5,4,4,4,4,4,4,12
161 7A DATA 13,4,4,4,4,4,4,4,4,4,5,4,4,4,4,4,4,12
162 IP DATA 13,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,14
163 uA DATA 11,4,4,4,4,5,4,4,5,0,5,0,5,4,5,4,5,12
164 xi DATA 11,4,4,4,5,5,5,0,14,0,14,0,5,4,4,5,4,4,12
165 M4 DATA 11,4,4,4,5,4,0,0,0,0,0,0,0,5,5,4,4,12
166 SL DATA 11,4,4,4,5,4,4,14,0,14,0,14,0,14,0,4,4,12
167 ud DATA 13,5,5,5,4,0,14,0,14,0,14,0,14,0,5,4,5,14
168 SA DATA 13,4,4,5,5,12,14,14,14,12,14,12,14,14,3,5,5,4,12
169 iT DATA 13,4,4,5,4,12,14,12,14,14,14,12,14,3,4,5,4,12
170 Wc DATA 13,4,4,5,4,14,14,12,14,12,14,14,14,3,3,4,5,4,12
171 Ze DATA 13,4,4,5,5,12,14,14,12,14,3,3,3,3,5,5,4,12
172 XD DATA 13,5,5,5,4,12,14,12,14,14,3,3,3,3,5,5,5,14
173 JM DATA 11,4,4,5,4,14,14,12,14,3,3,3,3,3,3,4,5,4,12
174 Jy DATA 11,4,4,5,5,12,14,14,3,3,3,3,3,3,3,5,4,4,12
175 vr DATA 11,4,4,5,4,12,14,3,3,3,3,3,3,3,3,4,5,4,12
176 uu DATA 11,4,4,5,4,14,3,3,3,3,3,3,3,3,4,5,4,12
177 5L DATA 14,14,14,14,14,3,3,3,3,3,3,3,3,14,14,14,14

```



```

178 S2 FOR y=20 TO 39:FOR x=40 TO 59
179 m9 READ d:PSET(x,y),d:NEXT x:NEXT y
180 40 DATA 13,11,11,11,11,11,11,11,11,11,13,11,11,11,11,11,11,11,
11,11
181 RU DATA 13,4,4,4,4,4,4,4,4,4,5,4,4,4,4,4,4,4,12
182 SV DATA 13,4,4,4,4,4,4,4,4,4,5,4,4,4,4,4,4,4,12
183 TW DATA 13,4,4,4,4,4,4,4,4,4,5,4,4,4,4,4,4,4,12
184 e1 DATA 13,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,14
185 GW DATA 11,4,4,4,4,5,4,4,5,0,5,4,5,4,5,4,4,4,12
186 J4 DATA 11,4,4,4,5,5,5,5,0,14,0,14,0,14,0,5,4,5,4,4,12
187 wm DATA 11,4,4,4,5,4,14,0,14,0,14,0,14,0,5,5,5,4,12
188 oh DATA 11,4,4,5,4,4,14,0,14,0,14,0,14,0,4,4,5,4,12
189 MW DATA 13,5,5,5,4,0,0,0,0,0,0,0,0,0,4,5,5,14
190 na DATA 13,4,4,5,5,0,14,0,14,0,14,0,14,0,3,5,5,4,12
191 gW DATA 13,4,4,5,4,0,14,0,14,0,14,0,14,0,3,4,5,4,12
192 D8 DATA 13,4,4,5,4,0,0,0,14,0,14,0,14,0,3,0,3,4,5,4,12
193 PH DATA 13,4,4,5,5,0,0,0,14,0,14,0,3,0,3,5,4,4,12
194 Hv DATA 13,5,5,5,4,0,0,0,14,0,3,0,3,0,3,5,5,5,14
195 xb DATA 11,4,4,5,4,0,14,0,14,0,3,0,3,0,3,4,5,4,12
196 YQ DATA 11,4,4,5,5,0,14,0,3,0,3,0,3,0,3,5,5,4,12
197 79 DATA 11,4,4,5,4,0,0,0,0,0,0,0,0,0,0,4,5,4,12
198 68 DATA 11,4,4,5,4,0,3,0,3,0,3,0,3,0,3,4,5,4,12
199 TJ DATA 14,14,14,14,14,0,3,0,3,0,3,0,3,0,3,14,14,14,14
200 8v FOR y=20 TO 39:FOR x=60 TO 79
201 8V READ d:PSET(x,y),d:NEXT x:NEXT y
202 IY DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
203 eu DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3
204 29 DATA 1,2,2,2,2,2,12,12,12,12,12,12,12,12,2,2,2,2,3
205 eu DATA 1,2,2,2,2,12,12,12,12,12,12,12,12,12,2,2,2,2,3
206 ps DATA 1,2,2,2,2,2,1,1,1,1,1,1,1,1,1,2,2,2,2,3
207 JE DATA 1,2,2,2,2,1,13,13,13,13,13,13,13,12,2,2,2,2,3
208 Tb DATA 1,2,2,2,2,2,12,13,13,13,13,13,13,12,2,2,2,2,3
209 wN DATA 1,2,2,2,2,2,12,1,13,13,13,13,1,12,2,2,2,2,3
210 V3 DATA 1,2,2,2,2,2,12,2,1,13,13,1,2,12,2,2,2,2,3
211 O1 DATA 1,2,2,2,2,2,12,2,1,1,2,2,2,12,2,2,2,2,3
212 mU DATA 1,2,2,2,2,2,12,2,2,13,1,2,2,12,2,2,2,2,3
213 He DATA 1,2,2,2,2,2,12,2,1,1,1,1,2,12,2,2,2,2,3
214 Ir DATA 1,2,2,2,2,2,12,1,1,1,13,1,1,12,2,2,2,2,3
215 FY DATA 1,2,2,2,2,2,12,1,1,1,1,1,1,1,12,2,2,2,2,3
216 cN DATA 1,2,2,2,2,2,1,1,1,13,1,1,1,12,2,2,2,2,3
217 7I DATA 1,2,2,2,2,2,1,1,1,13,13,1,1,12,2,2,2,2,3
218 r7 DATA 1,2,2,2,2,12,12,12,12,12,12,12,12,12,12,2,2,2,3
219 HO DATA 1,2,2,2,2,2,12,12,12,12,12,12,12,12,2,2,2,2,3
220 vB DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3
221 BR DATA 3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3
222 cK FOR y=20 TO 39:FOR x=80 TO 99
223 Ur READ d:PSET(x,y),d:NEXT x:NEXT y
224 eu DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
225 OG DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3
226 IH DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3
227 s7 DATA 1,2,2,2,2,2,2,13,13,13,2,2,2,2,2,2,2,3
228 hn DATA 1,2,2,2,2,2,2,2,13,5,13,14,2,2,2,2,2,3
229 Hf DATA 1,2,2,2,2,2,2,2,5,13,14,5,14,14,2,2,2,3
230 JX DATA 1,2,2,2,2,2,2,2,2,12,12,12,2,2,2,2,2,3
231 LS DATA 1,2,2,2,2,2,2,13,13,13,13,13,5,5,2,2,2,2,3
232 7a DATA 1,2,2,2,2,13,13,13,13,13,13,5,13,5,14,2,2,2,3
233 r7 DATA 1,2,2,2,13,13,14,14,5,0,5,14,5,14,14,2,2,3
234 3K DATA 1,2,2,13,13,13,13,5,0,0,0,0,14,5,14,2,2,3
235 QO DATA 1,2,2,13,13,5,13,5,0,0,0,5,5,14,14,2,2,3
236 wq DATA 1,2,2,13,5,14,5,5,0,0,0,5,14,14,14,2,3
237 9R DATA 1,2,2,13,13,5,14,5,13,5,0,5,0,5,14,14,2,3
238 vx DATA 1,2,2,2,13,14,5,13,0,0,0,5,5,14,14,2,2,3
239 sK DATA 1,2,2,2,14,14,14,14,14,0,5,14,14,14,14,2,2,3
240 b6 DATA 1,2,2,2,2,2,2,2,14,14,14,14,14,14,2,2,2,3
241 GW DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3
242 Hx DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3
243 XN DATA 3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3
244 WL FOR y=20 TO 39:FOR x=100 TO 119
245 qD READ d:PSET(x,y),d:NEXT x:NEXT y
246 xw DATA 15,15,1,15,15,15,15,15,15,15,15,15,15,15,15,15,1
,15
247 3I DATA 15,1,15,15,1,15,15,15,15,15,15,15,15,15,15,15,1
,15
248 mo DATA 1,15,15,1,15,15,15,15,1,15,15,15,15,15,15,15,1
5
249 Jc DATA 15,15,1,15,15,15,15,1,15,15,15,15,15,15,15,15,
1
250 T2 DATA 15,1,15,15,15,15,1,15,15,15,15,15,15,15,15,1,
15
251 we DATA 15,15,15,15,15,1,15,15,15,15,15,15,15,15,15,1
5

```

```

252 E4 DATA 15,1,15,15,1,1,15,15,15,15,1,15,15,15,15,1,15,15,  

253 90 DATA 15,15,15,1,15,15,15,15,1,15,15,15,15,15,1,15,15,15,  

254 41 DATA 15,15,1,15,15,15,15,15,1,15,15,15,15,15,1,15,15,15,15,  

255 fd DATA 15,1,15,15,15,15,15,1,1,15,15,15,15,15,1,15,15,15,15,15,  

256 zY DATA 1,15,15,15,15,15,1,1,15,15,15,15,15,1,15,15,15,15,15,15,  

257 us DATA 15,15,15,15,15,1,15,15,15,15,15,1,15,15,15,15,15,15,15,  

258 qY DATA 15,15,15,15,15,15,15,15,15,1,15,15,15,15,15,15,15,1,15,  

259 xG DATA 15,15,15,15,15,15,15,15,1,15,15,15,15,15,15,15,1,15,15,  

260 4y DATA 15,15,15,15,15,15,15,1,1,15,15,15,15,15,15,15,1,15,15,  

261 V5 DATA 15,15,15,15,15,15,15,1,15,15,15,15,15,15,1,15,15,15,15,  

262 DC DATA 15,15,15,15,15,15,1,15,15,15,15,15,15,15,1,15,15,15,15,  

263 1K DATA 15,1,15,15,1,1,15,15,15,15,15,15,15,15,15,15,15,15,15,  

264 SR DATA 1,15,15,15,1,15,15,15,15,15,15,15,15,15,15,15,15,15,15,  

265 Z9 DATA 15,15,15,1,15,15,15,15,15,15,15,15,15,15,15,15,15,15,1  

266 Cv FOR y=20 TO 39:FOR x=120 TO 139  

267 CZ READ d:PSET(x,y),d:NEXT x:NEXT y  

268 Me DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,  

269 iy DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3  

270 jz DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3  

271 2I DATA 1,2,2,2,2,2,2,6,6,6,6,6,6,6,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3  

272 1D DATA 1,2,2,2,2,6,6,6,6,6,6,6,6,6,6,6,2,2,2,2,2,2,2,2,2,2,2,2,3  

273 k4 DATA 1,2,2,2,6,6,6,6,0,0,0,0,0,0,0,6,6,6,6,2,2,2,2,2,2,2,2,3  

274 3Z DATA 1,2,2,2,2,0,0,0,0,2,2,2,2,2,6,6,6,6,2,2,2,2,2,2,2,2,2,2,3  

275 Wa DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,6,6,6,6,0,0,2,2,2,2,2,2,3  

276 jn DATA 1,2,2,2,2,2,6,6,6,6,6,6,6,6,6,6,0,0,2,2,2,2,2,2,2,2,2,2,3  

277 im DATA 1,2,2,2,2,6,6,6,6,0,0,0,0,0,0,0,0,0,0,2,2,2,2,2,2,2,2,3  

278 Hp DATA 1,2,2,2,6,6,6,6,0,0,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3  

279 YY DATA 1,2,2,2,2,6,6,6,6,2,2,2,2,2,6,6,6,6,6,2,2,2,2,2,2,2,2,3  

280 nr DATA 1,2,2,2,2,2,6,6,6,6,6,6,6,6,6,6,0,0,2,2,2,2,2,2,2,2,2,2,3  

281 As DATA 1,2,2,2,2,2,2,0,6,6,6,6,6,6,0,0,0,0,2,2,2,2,2,2,2,2,3  

282 Nf DATA 1,2,2,2,2,2,2,2,2,2,2,0,0,0,0,0,0,2,2,2,2,2,2,2,2,3  

283 Om DATA 1,2,2,2,2,2,2,2,2,2,6,6,6,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3  

284 Hd DATA 1,2,2,2,2,2,2,2,2,2,6,6,6,0,2,2,2,2,2,2,2,2,2,2,2,2,2,3  

285 my DATA 1,2,2,2,2,2,2,2,2,2,0,0,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3  

286 zF DATA 1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3  

287 FV DATA 3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3  

288 sV FOR y=20 TO 39:FOR x=140 TO 159  

289 Yv READ d:PSET(x,y),d:NEXT x:NEXT y  

290 iy DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,  

291 8g DATA 1,2,2,2,2,2,2,2,2,2,8,8,8,8,2,2,2,2,2,2,2,2,2,2,2,2,2,3  

292 tr DATA 1,2,2,2,2,2,2,2,8,8,8,8,8,8,8,8,2,2,2,2,2,2,2,2,2,2,2,3  

293 cW DATA 1,2,2,2,2,2,2,2,8,8,8,8,8,8,8,8,2,2,2,2,2,2,2,2,2,2,2,3  

294 nx DATA 1,2,2,2,2,2,2,2,8,8,8,8,8,8,8,8,8,8,2,2,2,2,2,2,2,2,3  

295 bd DATA 1,2,2,2,2,2,2,2,8,8,8,8,8,8,8,8,8,8,2,2,2,2,2,2,2,2,3  

296 zx DATA 1,2,2,12,12,2,2,8,8,0,0,0,0,0,0,0,8,8,2,2,2,2,2,2,2,2,3  

297 m4 DATA 1,2,2,12,2,2,2,2,8,8,0,0,0,0,8,8,2,2,2,2,2,2,2,2,2,2,2,3  

298 p5 DATA 1,2,2,12,2,2,2,2,8,8,8,8,8,8,8,8,2,2,2,2,2,2,2,2,2,2,3  

299 8x DATA 1,2,12,2,2,2,2,2,3,12,8,8,8,12,14,2,2,2,2,2,2,2,2,2,2,2,3  

300 m5 DATA 1,2,12,12,2,2,13,13,12,14,14,14,14,14,14,2,2,2,2,2,2,2,2,2,2,3  

301 15 DATA 1,2,12,12,13,13,12,14,14,5,0,14,14,12,14,14,2,2,2,2,2,2,2,2,3  

302 se DATA 1,2,2,13,13,13,12,14,5,0,0,0,0,12,14,14,2,2,2,2,2,2,2,2,2,2,3  

303 vW DATA 1,2,2,13,13,5,12,14,0,5,0,5,5,12,14,14,14,2,2,2,2,2,2,2,2,3  

304 e5 DATA 1,2,2,13,5,14,12,14,5,0,0,0,5,12,14,14,14,2,2,2,2,2,2,2,2,3  

305 hd DATA 1,2,2,13,13,12,12,12,13,5,0,5,5,12,12,12,14,14,2,2,2,2,2,2,2,3  

306 br DATA 1,2,2,2,2,12,12,12,12,12,0,0,0,0,12,12,12,12,12,12,2,2,2,2,2,2,3  

307 yQ DATA 1,2,2,2,2,14,14,14,14,14,14,0,5,14,14,14,14,2,2,2,2,2,2,2,2,3  

308 hC DATA 1,2,2,2,2,2,2,2,2,14,14,14,14,14,14,14,14,2,2,2,2,2,2,2,2,3  

309 br DATA 3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3  

310 ql OPEN"d:\fo\Objects" FOR OUTPUT AS #1  

311 W8 FOR i=0 TO 47:READ d:PRINT #1,CHR$(d);:NEXT i  

312 Q5 DIM g#(19)  

313 pX FOR i=0 TO 39  

314 7b GET(0,i)-(299,i),g#  

315 2Q FOR o=0 TO 19:PRINT #1,MKD$(g#(o));:NEXT o  

316 Xn NEXT i  

317 L8 CLOSE #1

```

Listing 2. (Fortsetzung)


```

318 C1 ERASE g#
319 io WINDOW CLOSE 2:SCREEN CLOSE 2
320 OT PRINT"Jau,jetz is das Objektbild aufa Diskette.Forever."
321 Yw PRINT"Good Byte !"
322 wr END
323 4L DATA 0,0,0,12,12,15,10,10,13,8,8,11,11,4,0,12
324 Mp DATA 10,10,15,0,0,12,11,0,4,10,0,2,4,11,9,8
325 R2 DATA 0,13,6,4,8,0,0,14,12,12,10,8,8,13,13,14
326 UK MoveTo:
327 Cm DIM g#(50)
328 GV GetX=GetX*20:GetY=GetY*20
329 xE PutX=PutX*20:PutY=PutY*20
330 bK GET(GetX,GetY)-(GetX+19,GetY+19),g#
331 Em PUT(PutX,PutY),g#,PSET
332 Qz ERASE g#
333 jL RETURN
334 no Switch:
335 Ne GetX=GetX*20:GetY=GetY*20
336 xE FOR y=GetY TO GetY+19
337 i6 FOR x=GetX TO GetX+19
338 i9 IF POINT(x,y)=Which THEN PSET(x,y),ToWhich
339 Ot NEXT x
340 Rx NEXT y
341 rT RETURN
(C) 1990 M&T

```

Listing 2. (Schluß)

```

43 ow DATA 13,0c,01,0b,13,13,08,0e,0a,10
44 ew DATA 01,13,01,01,01,01
(C) 1990 M&T

```

Listing 3. Die Daten für Level 1

Programmname:	Level 2
Computer:	A500, A1000, A2000 mit Kickstart 1.2 oder 1.3
Sprache:	Amiga-Basic
Bemerkung:	erzeugt die Daten für Level 2

```

Programmautor: Jrg Isselman
-----
1 eRO REM Generiert Level 2
2 ag CLS
3 oP OPEN " 2" FOR OUTPUT AS 1
4 BS READ anz
5 oa FOR i=1 TO anz
6 3n1 READ h$
7 yB2 wert1=ASC(LEFT$(h$,1))
8 bP IF wert1>64 THEN wert1=wert1-87 ELSE wert1=wert1-48
9 FI wert1=wert1*16
10 7c wert2=ASC(RIGHT$(h$,1))
11 wp IF wert2>64 THEN wert2=wert2-87 ELSE wert2=wert2-48
12 Pi wert=wert1+wert2
13 9G PRINT #1,CHR$(wert);
14 JOO NEXT
15 3n CLOSE 1
16 Ov END
17 yc Werte:
18 5Y DATA 256
19 JZ DATA 00,00,00,00,00,00,00,00,00,00,00
20 q9 DATA 00,00,00,01,12,00,01,10,01,01
21 Sp DATA 01,01,01,01,01,01,01,06,01,12
22 DW DATA 00,01,03,01,12,00,01,12,01,00
23 sB DATA 12,01,13,09,12,10,01,0d,01,00
24 Ah DATA 13,10,13,10,13,00,01,08,01,12
25 6O DATA 03,01,10,01,01,10,01,01,01,10
26 GJ DATA 01,01,0b,01,01,07,13,13,10,13
27 t9 DATA 00,13,00,01,13,12,01,10,00,15
28 La DATA 10,12,12,01,01,10,01,01,01,10
29 rD DATA 01,01,02,01,15,12,12,12,01,00
30 Fl DATA 13,10,13,10,13,00,01,04,01,15
31 ou DATA 12,12,12,01,12,00,01,12,00,00
32 Wm DATA 00,00,00,00,00,00,00,00,00,00
33 CS DATA 00,01,12,00,01,10,01,01,01,01
34 Ws DATA 01,01,01,01,01,06,01,12,00,01
35 Xo DATA 03,01,12,00,01,12,01,00,12,01
36 rN DATA 13,09,12,10,01,0d,01,00,13,10
37 lc DATA 13,10,13,00,01,08,01,12,03,01
38 IT DATA 10,01,01,10,01,01,01,10,01,01
39 Yx DATA 0b,01,01,07,13,13,10,13,00,13
40 xI DATA 00,01,13,12,01,10,00,15,10,12
41 Re DATA 12,01,01,10,01,01,01,10,01,01
42 7R DATA 02,01,15,12,12,12,01,00,13,10
43 Kr DATA 13,10,13,00,01,04,01,15,12,12
44 ag DATA 12,01,12,00,01,12
(C) 1990 M&T

```

Listing 4. Die Daten für Level 2

Programmname:	Level 3
Computer:	A500, A1000, A2000 mit Kickstart 1.2 oder 1.3
Sprache:	Amiga-Basic
Bemerkung:	erzeugt die Daten für Level 3

```

Programmautor: Jrg Isselman
-----
1 gUO REM Generiert Level 3
2 ag CLS
3 sU OPEN " 3" FOR OUTPUT AS 1
4 BS READ anz

```

Programmname:	Level 1
Computer:	A500, A1000, A2000 mit Kickstart 1.2 oder 1.3
Sprache:	Amiga-Basic
Bemerkung:	erzeugt die Daten für Level 1

```

Programmautor: Jrg Isselman
-----
1 oO REM Generiert Level 1
2 ag CLS
3 kK OPEN " 1" FOR OUTPUT AS 1
4 BS READ anz
5 oa FOR i=1 TO anz
6 3n1 READ h$
7 yB2 wert1=ASC(LEFT$(h$,1))
8 bP IF wert1>64 THEN wert1=wert1-87 ELSE wert1=wert1-48
9 FI wert1=wert1*16
10 7c wert2=ASC(RIGHT$(h$,1))
11 wp IF wert2>64 THEN wert2=wert2-87 ELSE wert2=wert2-48
12 Pi wert=wert1+wert2
13 9G PRINT #1,CHR$(wert);
14 JOO NEXT
15 3n CLOSE 1
16 Ov END
17 yc Werte:
18 5Y DATA 256
19 qo DATA 08,0c,03,0b,01,01,01,01,01,01
20 D1 DATA 01,0b,04,12,09,01,01,01,0b,0b
21 nP DATA 03,0b,03,0b,03,0b,0b,01,01,10
22 9l DATA 0b,13,13,0b,01,01,0b,01,01,01
23 ha DATA 01,01,01,0b,0b,04,01,01,01,01
24 9g DATA 12,13,0b,02,03,0b,13,13,0b,01
25 eX DATA 10,0b,01,12,01,01,0b,01,01,01
26 TO DATA 13,01,01,11,12,01,0b,13,13,0b
27 4o DATA 01,0b,03,0b,03,0b,01,01,01,01
28 Yr DATA 01,01,01,13,13,01,01,01,03,01
29 Os DATA 13,01,01,01,01,01,15,13,13,0b
30 k9 DATA 01,07,13,0c,01,0b,13,13,08,0e
31 ss DATA 0a,10,01,13,01,01,01,01,08,0c
32 xs DATA 03,0b,01,01,01,01,01,01,01,0b
33 wm DATA 04,12,09,01,01,01,0b,0b,03,0b
34 U4 DATA 03,0b,03,0b,0b,01,01,10,0b,13
35 2y DATA 13,0b,01,01,0b,01,01,01,01,01
36 aW DATA 01,0b,0b,04,01,01,01,01,12,13
37 Xz DATA 0b,02,03,0b,13,13,0b,01,10,0b
38 YW DATA 01,12,01,01,0b,01,01,01,13,01
39 wY DATA 01,11,12,01,0b,13,13,0b,01,0b
40 qn DATA 03,0b,03,0b,01,01,01,01,01,01
41 79 DATA 01,13,13,01,01,01,03,01,13,01
42 lv DATA 01,01,01,01,15,13,13,0b,01,07

```



```

5 oa FOR i=1 TO anz
6 3n1 READ h$
7 yB2 wert1=ASC(LEFT$(h$,1))
8 bP IF wert1>64 THEN wert1=wert1-87 ELSE wert1=wert1-48
9 FI wert1=wert1*16
10 7c wert2=ASC(RIGHT$(h$,1))
11 wp IF wert2>64 THEN wert2=wert2-87 ELSE wert2=wert2-48
12 P1 wert=wert1+wert2
13 9G PRINT #1,CHR$(wert);
14 JOO NEXT
15 3n CLOSE 1
16 Ov END
17 yc Werte:
18 zQ DATA 155
19 9I DATA 10,13,10,01,10,13,13,13,13,13
20 8U DATA 13,13,13,13,10,13,10,13,13,10
21 zA DATA 01,01,10,13,13,13,13,13,10,10
22 in DATA 13,13,13,01,01,01,01,13,01,0b
23 t6 DATA 10,10,10,10,10,13,13,13,10,13
24 F1 DATA 13,10,13,01,10,10,10,10,01,10
25 ZU DATA 13,13,13,13,01,0a,13,13,01,01
26 Lh DATA 01,01,10,01,10,10,10,13,13,01
27 FT DATA 10,13,10,01,10,10,10,10,01,10
28 Oe DATA 01,01,13,13,01,01,00,01,10,10
29 v8 DATA 01,01,01,01,10,10,13,13,13,13
30 7N DATA 10,01,01,10,01,10,10,10,01,10
31 vI DATA 13,01,13,10,13,13,10,01,10,01
32 XT DATA 10,01,10,01,10,10,13,0b,01,01
33 DO DATA 10,10,01,10,10,10,01,10,10,10
34 IF DATA 06,06,02,00,0e8
(C) 1990 M&T

```

Listing 5. Die Daten für Level 3

Programmname:	Level 4
Computer:	A500, A1000, A2000 mit Kickstart 1.2 oder 1.3
Sprache:	Amiga-Basic
Bemerkung:	erzeugt die Daten für Level 4

```

Programmautor: Jrg Isselmann
-----
1 iX0 REM Generiert Level 4
2 ag CLS
3 wZ OPEN " 4" FOR OUTPUT AS 1
4 BS READ anz
5 oa FOR i=1 TO anz
6 3n1 READ h$
7 yB2 wert1=ASC(LEFT$(h$,1))
8 bP IF wert1>64 THEN wert1=wert1-87 ELSE wert1=wert1-48
9 FI wert1=wert1*16
10 7c wert2=ASC(RIGHT$(h$,1))
11 wp IF wert2>64 THEN wert2=wert2-87 ELSE wert2=wert2-48
12 P1 wert=wert1+wert2
13 9G PRINT #1,CHR$(wert);
14 JOO NEXT
15 3n CLOSE 1
16 Ov END
17 yc Werte:
18 zQ DATA 155
19 b1 DATA 01,01,00,00,00,01,01,00,00,00
20 Fm DATA 01,10,10,07,06,00,00,00,01,00
21 i0 DATA 00,01,00,01,00,00,00,01,03,01
22 o6 DATA 00,01,02,01,01,00,00,00,01,01
23 xe DATA 01,06,10,0c,01,00,01,00,00,01
24 uw DATA 01,01,01,01,12,12,12,01,0a,00
25 5Q DATA 10,01,01,08,00,12,01,00,00,00
26 Of DATA 01,15,01,01,00,0a,03,01,00,00
27 wA DATA 12,01,10,10,01,01,00,00,00,00
28 en DATA 05,00,00,01,01,01,01,00,0b,01
29 F9 DATA 09,00,01,01,04,00,01,00,00,03
30 DG DATA 00,01,01,00,01,01,01,01,00,0e
31 q6 DATA 00,01,01,01,01,00,00,01,00,00
32 7W DATA 00,00,01,00,01,00,00,15,00,01
33 n7 DATA 01,00,00,00,01,01,00,00,00,01
34 5A DATA 07,04,01,00,0e8
(C) 1990 M&T

```

Listing 6. Die Daten für Level 4

Programmname:	Level 5
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	Amiga-Basic
Bemerkung:	erzeugt die Daten für Level 5

```

Programmautor: Jrg Isselmann
-----
1 ka0 REM Generiert Level 5
2 ag CLS
3 Oe OPEN " 5" FOR OUTPUT AS 1
4 BS READ anz
5 oa FOR i=1 TO anz
6 3n1 READ h$
7 yB2 wert1=ASC(LEFT$(h$,1))
8 bP IF wert1>64 THEN wert1=wert1-87 ELSE wert1=wert1-48
9 FI wert1=wert1*16
10 7c wert2=ASC(RIGHT$(h$,1))
11 wp IF wert2>64 THEN wert2=wert2-87 ELSE wert2=wert2-48
12 P1 wert=wert1+wert2
13 9G PRINT #1,CHR$(wert);
14 JOO NEXT
15 3n CLOSE 1
16 Ov END
17 yc Werte:
18 zQ DATA 155
19 60 DATA 11,10,10,10,11,01,10,10,10,10
20 KB DATA 0b,01,10,10,10,11,10,10,01,0b
21 2D DATA 01,10,10,10,01,10,01,10,10,10
22 20 DATA 11,10,10,01,10,01,10,01,10,01
23 xI DATA 10,10,10,10,10,10,10,10,01,10
24 ai DATA 10,10,10,0b,01,10,10,01,10,10
25 2I DATA 10,01,01,01,01,01,01,01,01,01
26 FI DATA 10,10,10,10,10,10,0b,10,10
27 QT DATA 0b,10,10,10,10,10,01,10,01,10
28 30 DATA 10,10,01,01,01,01,01,01,01,10
29 80 DATA 10,10,01,10,10,10,10,10,10,10
30 HP DATA 10,10,0b,10,10,01,10,01,01,10
31 80 DATA 10,10,01,10,10,10,01,01,01,10
32 B9 DATA 01,10,01,0b,10,00,10,10,0a,01
33 CS DATA 10,10,10,10,10,10,10,10,10,01
34 mo DATA 00,09,07,00,78
(C) 1990 M&T

```

Listing 7. Die Daten für Level 5

Programmname:	Level 6
Computer:	A500, A1000, A2000 mit Kickstart 1.2 oder 1.3
Sprache:	Amiga-Basic
Bemerkung:	erzeugt die Daten für Level 6

```

Programmautor: Jrg Isselmann
-----
1 md0 REM Generiert Level 6
2 ag CLS
3 4j OPEN " 6" FOR OUTPUT AS 1
4 BS READ anz
5 oa FOR i=1 TO anz
6 3n1 READ h$
7 yB2 wert1=ASC(LEFT$(h$,1))
8 bP IF wert1>64 THEN wert1=wert1-87 ELSE wert1=wert1-48
9 FI wert1=wert1*16
10 7c wert2=ASC(RIGHT$(h$,1))
11 wp IF wert2>64 THEN wert2=wert2-87 ELSE wert2=wert2-48
12 P1 wert=wert1+wert2
13 9G PRINT #1,CHR$(wert);
14 JOO NEXT
15 3n CLOSE 1
16 Ov END
17 yc Werte:
18 zQ DATA 155
19 yE DATA 10,10,10,10,10,01,10,10,10,10
20 H7 DATA 0b,01,10,10,10,10,10,10,01,0b
21 2D DATA 01,10,10,10,01,10,01,10,10,10

```

Listing 8. Die Daten für Level 6


```

22 zK DATA 10,10,10,01,10,01,10,01,10,01
23 xI DATA 10,10,10,10,10,10,10,10,01,10
24 aI DATA 10,10,10,0b,01,10,10,01,10,10
25 2I DATA 10,01,01,01,01,01,01,01,01,01
26 FI DATA 10,10,10,10,10,10,10,0b,10,10
27 QT DATA 0b,10,10,10,10,10,01,10,01,10
28 3O DATA 10,10,01,01,01,01,01,01,01,10
29 8O DATA 10,10,01,10,10,10,10,10,10,10
30 HP DATA 10,10,0b,10,10,01,10,01,01,10
31 8O DATA 10,10,01,10,10,10,01,01,01,10
32 B9 DATA 01,10,01,0b,10,00,10,10,0a,01
33 CS DATA 10,10,10,10,10,10,10,10,10,01
34 mo DATA 00,09,07,00,78
(C) 1990 M&T

```

Listing 8. (Schluß)

```

5 oa FOR i=1 TO anz
6 3n1 READ h$
7 yB2 wert1=ASC(LEFT$(h$,1))
8 bP IF wert1>64 THEN wert1=wert1-87 ELSE wert1=wert1-48
9 FI wert1=wert1*16
10 7c wert2=ASC(RIGHT$(h$,1))
11 wp IF wert2>64 THEN wert2=wert2-87 ELSE wert2=wert2-48
12 Pi wert=wert1+wert2
13 9G PRINT #1,CHR$(wert);
14 JOO NEXT
15 3n CLOSE 1
16 Ov END
17 yc Werte:
18 zQ DATA 155
19 yE DATA 10,10,10,10,10,01,10,10,10,10
20 H7 DATA 0b,01,10,10,10,10,10,10,01,0b
21 2D DATA 01,10,10,10,01,10,01,10,10,10
22 zK DATA 10,10,10,01,10,01,10,01,10,01
23 xI DATA 10,10,10,10,10,10,10,10,01,10
24 aI DATA 10,10,10,0b,01,10,10,01,10,10
25 2I DATA 10,01,01,01,01,01,01,01,01,01
26 FI DATA 10,10,10,10,10,10,10,0b,10,10
27 QT DATA 0b,10,10,10,10,10,01,10,01,10
28 3O DATA 10,10,01,01,01,01,01,01,01,10
29 8O DATA 10,10,01,10,10,10,10,10,10,10
30 HP DATA 10,10,0b,10,10,01,10,01,01,10
31 8O DATA 10,10,01,10,10,10,01,01,01,10
32 B9 DATA 01,10,01,0b,10,00,10,10,0a,01
33 CS DATA 10,10,10,10,10,10,10,10,10,01
34 ze DATA 00,09,07,00,3c
(C) 1990 M&T

```

Listing 10. Die Daten für Level 8

Programmname:	Level 7
Computer:	A500, A1000, A2000 mit Kickstart 1.2 oder 1.3
Sprache:	Amiga-Basic
Bemerkung:	erzeugt die Daten für Level 1

Programmautor: Jrg Isselmann

```

1 og0 REM Generiert Level 7
2 ag CLS
3 8o OPEN " 7" FOR OUTPUT AS 1
4 BS READ anz
5 oa FOR i=1 TO anz
6 3n1 READ h$
7 yB2 wert1=ASC(LEFT$(h$,1))
8 bP IF wert1>64 THEN wert1=wert1-87 ELSE wert1=wert1-48
9 FI wert1=wert1*16
10 7c wert2=ASC(RIGHT$(h$,1))
11 wp IF wert2>64 THEN wert2=wert2-87 ELSE wert2=wert2-48
12 Pi wert=wert1+wert2
13 9G PRINT #1,CHR$(wert);
14 JOO NEXT
15 3n CLOSE 1
16 Ov END
17 yc Werte:
18 zQ DATA 155
19 6N DATA 00,01,01,01,01,01,01,01,00,13
20 Sf DATA 13,13,03,01,01,00,01,01,01,01
21 o8 DATA 01,00,10,13,10,00,00,00,01,01
22 v9 DATA 00,01,01,01,01,01,00,01,10,01
23 u8 DATA 01,01,00,01,01,00,01,01,01,01
24 zN DATA 01,00,10,07,10,00,01,00,01,01
25 4E DATA 00,01,01,01,01,01,01,01,10,01
26 dr DATA 00,01,00,01,01,00,00,00,00,00
27 NQ DATA 00,00,01,0a,01,00,01,00,01,01
28 5Q DATA 00,01,03,01,01,01,00,01,01,01
29 mw DATA 00,01,00,01,01,00,01,00,00,00
30 kz DATA 01,00,00,00,00,00,01,00,01,01
31 1F DATA 00,01,01,01,00,01,01,01,01,01
32 KC DATA 01,01,00,01,01,00,03,0b,01,00
33 ew DATA 00,00,00,00,00,00,00,00,01,01
34 Yj DATA 00,04,01,00,46
(C) 1990 M&T

```

Listing 9. Die Daten für Level 7

Programmname:	Level 9
Computer:	A500, A1000, A2000 mit Kickstart 1.2 oder 1.3
Sprache:	Amiga-Basic
Bemerkung:	erzeugt die Daten für Level 9

Programmautor: Jrg Isselmann

```

1 sm0 REM Generiert Level 9
2 ag CLS
3 Gy OPEN " 9" FOR OUTPUT AS 1
4 BS READ anz
5 oa FOR i=1 TO anz
6 3n1 READ h$
7 yB2 wert1=ASC(LEFT$(h$,1))
8 bP IF wert1>64 THEN wert1=wert1-87 ELSE wert1=wert1-48
9 FI wert1=wert1*16
10 7c wert2=ASC(RIGHT$(h$,1))
11 wp IF wert2>64 THEN wert2=wert2-87 ELSE wert2=wert2-48
12 Pi wert=wert1+wert2
13 9G PRINT #1,CHR$(wert);
14 JOO NEXT
15 3n CLOSE 1
16 Ov END
17 yc Werte:
18 zQ DATA 155
19 tc DATA 08,07,10,0e,12,12,15,15,12,13
20 jf DATA 12,13,12,15,0a,10,01,01,10,12
21 Qd DATA 15,15,15,12,13,12,13,12,15,15
22 dq DATA 0d,10,10,10,12,15,12,15,12,12
23 sC DATA 12,12,12,15,15,11,13,13,13,13
24 fx DATA 13,13,13,04,13,13,13,13,10,10
25 oz DATA 00,01,13,01,0b,13,12,13,02,03
26 IL DATA 01,02,01,03,0b,00,01,13,01,10
27 MW DATA 13,13,13,03,05,01,04,01,02,10
28 mI DATA 00,13,13,13,13,0b,13,0b,05,0b
29 ay DATA 0b,0b,0b,0b,15,0e,00,00,00,01
30 yv DATA 13,0b,13,01,01,01,0b,01,10,10
31 MJ DATA 10,01,01,11,01,13,01,10,10,0b
32 pF DATA 0b,0b,01,11,11,06,09,10,13,13
33 ST DATA 0b,01,00,00,01,01,01,01,01,01
34 la DATA 07,09,0f,00,64
(C) 1990 M&T

```

Listing 11. Die Daten für Level 9

Programmname:	Level 8
Computer:	A500, A1000, A2000 mit Kickstart 1.2 oder 1.3
Sprache:	Amiga-Basic
Bemerkung:	erzeugt die Daten für Level 8

Programmautor: Jrg Isselmann

```

1 qj0 REM Generiert Level 8
2 ag CLS
3 Ct OPEN " 8" FOR OUTPUT AS 1
4 BS READ anz

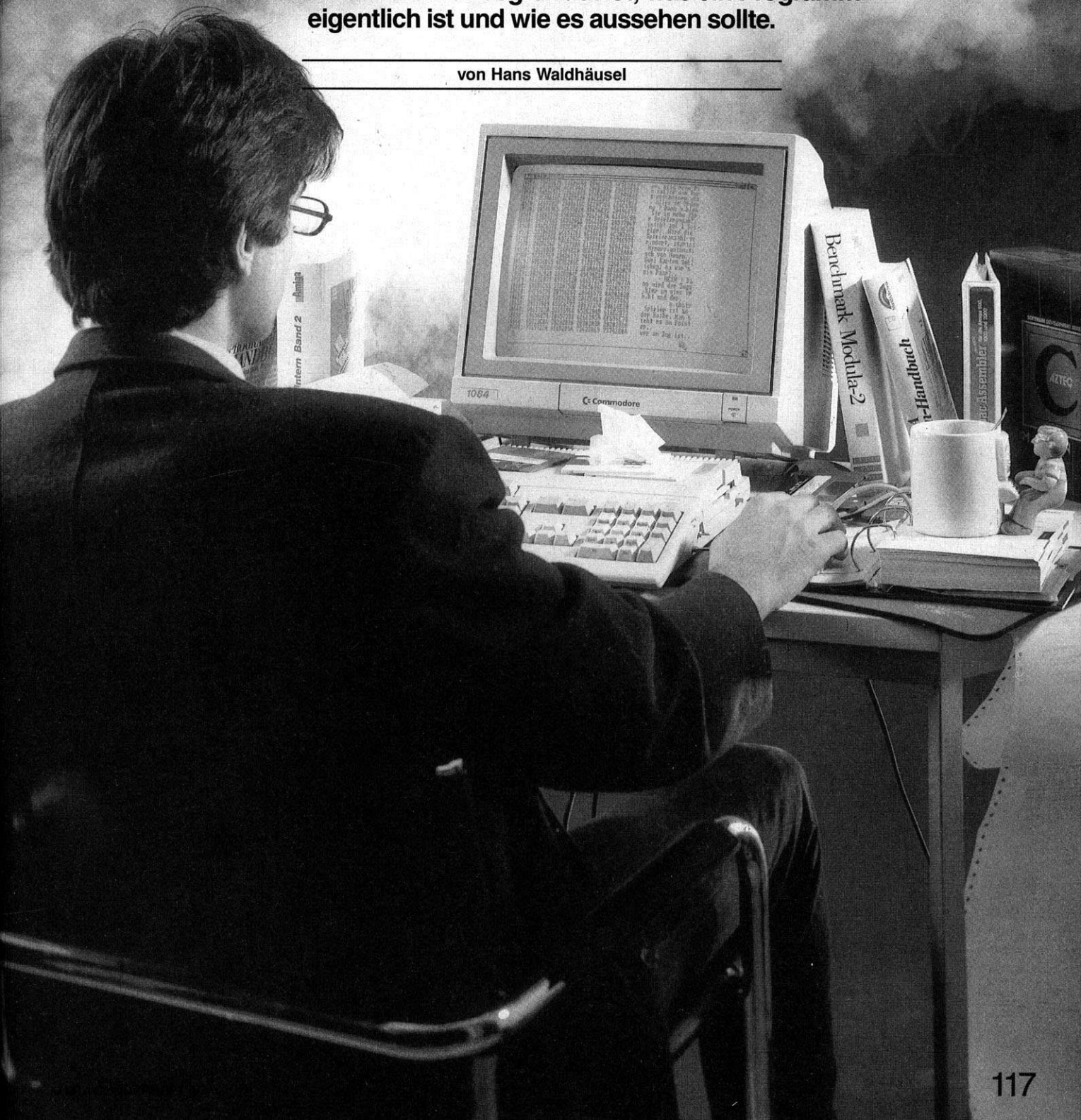
```


Programmieren - was ist das?

Bits, Bytes und Spaghetti

Es soll Leute geben, die sind auch noch stolz darauf, daß sie ihre Programme »stehend-freihändig von der Bettkante« aus dem Handgelenk in die Tastatur klopfen. Wir sagen Ihnen, warum dieser Stolz unbegründet ist, was ein Programm eigentlich ist und wie es aussehen sollte.

von Hans Waldhäusel



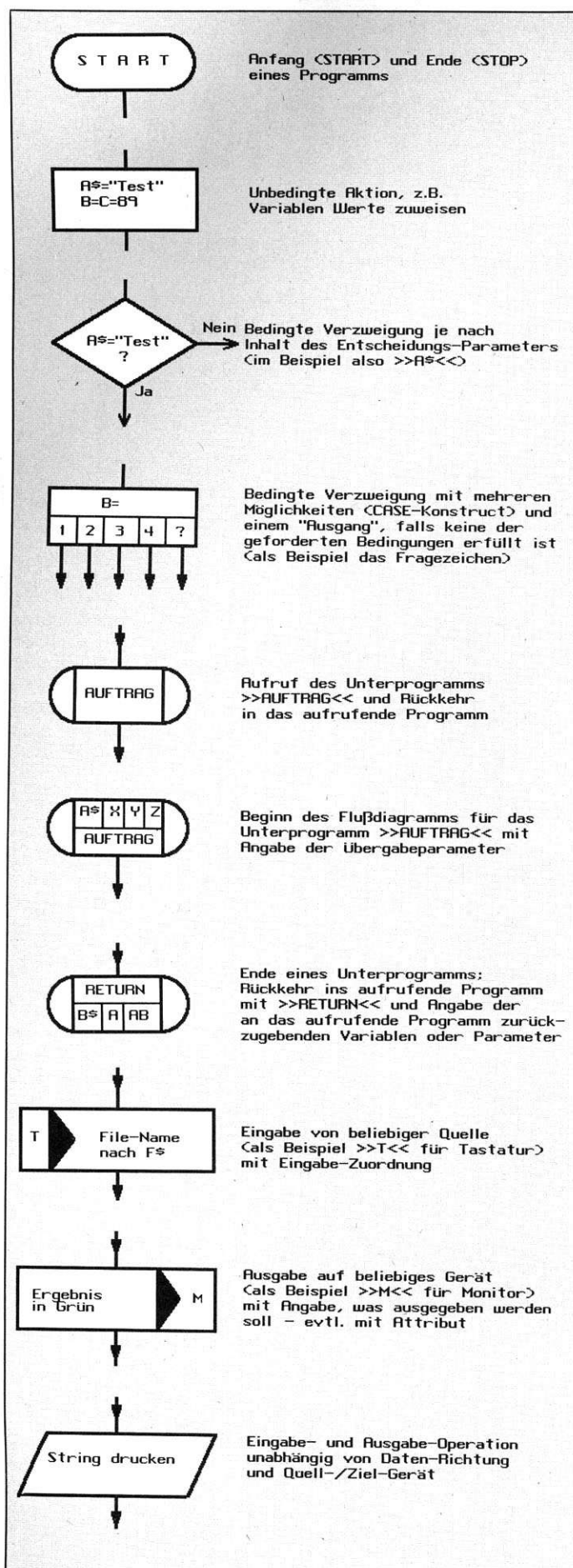


Bild 1. Die Grundsymbole eines Flußdiagramms

Die deutsche Sprache wurde seit der Erfindung des Computers um viele Worte bereichert. Anglizismen wie »Computer«, »Assembler«, »Harddisk« und »Compiler« wurden in den deutschen Sprachschatz ebenso aufgenommen wie Übersetzungen aus dem Englischen, als da wären: »Flußdiagramm« (vom englischen »Flowchart«), »Diskette« (abgeleitet aus »Disk«, also »Scheibe«), »Diskettenlaufwerk« (von »Floppy-Disk-Drive« - aber wer möchte schon »Schwabbelscheibenantrieb« sagen?) und viele andere mehr. Ein Wort haben sich die Computerleute angeeignet, das es schon vor der Erfindung des Computers gab, allerdings in ganz anderer Bedeutung: »Programm«.

In der präcomputerischen Zeit (auch dieses Wort finden Sie in keinem Lexikon) war ein Pro-

gramm die Beschreibung einer zeitlichen Reihenfolge geplanter Ereignisse; noch heute können Sie im Zirkus ein »Programm« bekommen, in dem der vorgesehene chronologische Ablauf der Veranstaltung festgehalten ist. In den Kinos gab und gibt es ebenfalls »Programme«, aber die haben schon nichts mehr mit einer Ablaufbeschreibung zu tun - das sind mehr Beschreibungen des Films selbst, seiner Darsteller usw.

Fast wie im Zirkus

Seit sich Computer auf der Welt verbreiten, hat »Programm« eine zusätzliche, ganz spezifische Bedeutung bekommen, die am ehesten noch mit dem Beispiel des Zirkusprogramms zu vergleichen ist: eine Aneinanderreihung von Befehlen, die der Computer abzu-

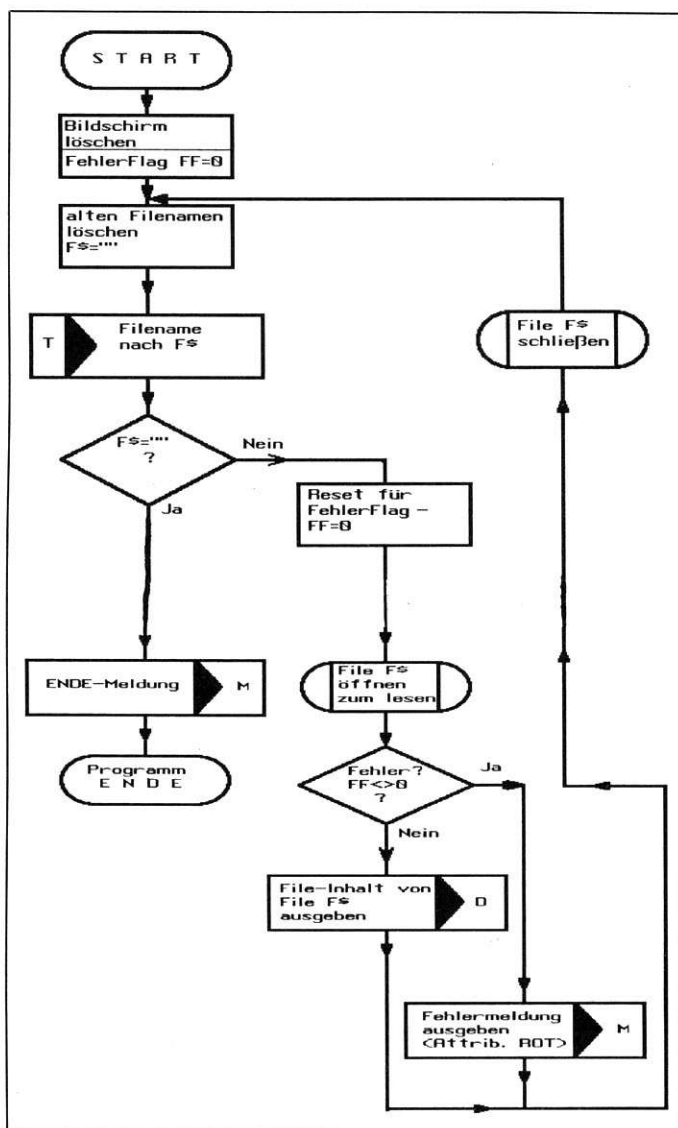


Bild 2. So ein Flußdiagramm macht doch alles gleich viel klarer - oder?

beiten hat, nach dem Willen des Programmierers (auch den Beruf des »Programmierers« gibt es erst, seit es Computer gibt). Der »Große Brockhaus« (zumindest der, in dem wir nachgeschlagen haben) hat nicht weniger als sieben verschiedene Definitionen für das Wort »Programm«, nicht mitgezählt die ganzen Ableitungen dieses Begriffs.

In der Computerwelt versteht man unter einem Programm die Folge von Anweisungen, die festlegt, was dieser graue Kasten gerade macht. Programme können dabei sehr unterschiedlich aussehen. Sie können unbemerkt im Hintergrund laufen, ganzseitige Fenster öffnen und den Benutzer aufdringlich zu einer Eingabe auffordern oder sie können den Eindruck erwecken, gar nicht vorhanden zu sein und dabei auf die Eingabe einer bestimmten Tastenkombination lauern. Selbstverständlich können auch mehrere Programme gleichzeitig ablaufen oder selbstständig andere Programme aufrufen.

Ein solches Programm ist beispielsweise dafür verantwortlich, daß Sie nach dem Einschalten eine Meldung auf dem Bildschirm erhalten, die nach der Workbench verlangt. Dieses Programm ist im sog. »ROM« (Read Only Memory – Speicherbereich, der nur gelesen werden kann) enthalten und wird beim Einschalten automatisch gestartet. Ein weiteres Programm ist dafür verantwortlich, daß die Workbench geladen wird und vom Benutzer über die Workbench eingegebene Befehle ausgeführt werden.

Bei jedem Aufruf eines Befehls in der Shell wird ebenfalls ein Programm desselben Namens gestartet. Sie sehen also, Programme sind für die Arbeit mit Computern einfach unumgänglich.

Sisyphus läßt grüßen

Wie kommen Sie zu einem Programm? Da gibt's einmal die Möglichkeit des Kaufens – manchmal preiswert (Public Domain ist eine wahre Fundgrube), manchmal weniger preiswert. Als nächstes wäre das Kopieren zu erwähnen, wobei ausnahmsweise einmal kein Unterschied zwischen legalem und illegalem Kopieren gemacht wird; das soll aber wirklich eine Ausnahme blei-

ben – was wir von Raubkopierern halten, überlassen wir Ihrer eigenen Phantasie (denken Sie sich am besten eine Sprechblase: »Grrrrrrr! Abscheu! Vabrecha! ...«). Eine weitere Variante ist das Abtippen von Listings – Sisyphus läßt grüßen. Die vierte und letzte Alternative ist die kreativste von allen: »Do it yourself«, auf deutsch »selber denken macht schlau«.

Sie haben sich entschlossen, zu den Kreativen gehören zu wollen? Dann lassen Sie sich nicht dazu verleiten, einfach drauflos zu »hacken«. Ein ungeplantes Programm ist wie eine alte Flinte – man weiß nie, wann sie nach hinten losgeht. Erlauben Sie uns, Ihnen einige Tips und Ratschläge zu geben, die Sie beim Programmieren beachten sollten.

Zuerst einmal: es spielt überhaupt keine Rolle, ob Sie mittels Basic, Assembler, C, Modula oder einer anderen Sprache mit Ihrem Computer in Verbindung treten wollen. Die Grundregeln sind überall die gleichen, nur ist die Konsequenz der Nicht-Einhaltung dieser Regeln nicht bei allen Sprachen gleich durchschlagend.

Zum Beispiel ist Basic in den schlechten Ruf gekommen, den Programmierer zum »Spaghetti-Code« zu animieren und zu wilden »GOTO«-Sprüngen zu verleiten. Lassen Sie sich gesagt sein: in Assembler geht das genauso. Bloß wird bei einem Assembler-Programm die Lesbarkeit durch solche Aktionen derart schlecht, daß es schon purer Masochismus wäre, solche Dinge zu verwenden. Lassen Sie sich nicht durch Berichte oder Gerüchte über ach so begnadete Programmierer irritieren, die sich einfach an die Tastatur setzen und denen die tollsten Programme aus den Handwurzelknochen rinne – die Ergebnisse sind in keinem Fall als »optimal« zu bezeichnen.

Okay, wir wissen schon, daß sich das alles nach »Ein Rat fürs Leben« und nach erhobem Zeigefinger anhört. Aber bitte glauben Sie uns: Wir wissen, wovon wir sprechen. Wenn Sie wollen, klappen Sie das Sonderheft zu, setzen Sie sich an Ihren Computer und seien Sie genial intuitiv.

Lassen Sie möglichst alle Kommentare weg und springen Sie lustig von einem Label zum anderen. Flickten Sie hier was ein und ändern Sie dort eine Routine. Wenn Sie 200 Programmzeilen geschrieben ha-

12 wichtige Regeln

- | | |
|-------------------|--|
| Regel # 1 | Finger weg von der Tastatur. Hinsetzen und einhacken ist nur auf den ersten Blick die schnellste Variante. Jede Minute, die Sie für die Planung »verschwenden«, macht sich spätestens beim Fehlerbeseitigen bezahlt. |
| Regel # 2 | Die erste Lösung ist selten die beste. Suchen Sie ständig nach Alternativen. |
| Regel # 3 | Dagobert ist geizig. Sie sollten es auch sein – geizen Sie mit Speicherplatz, mit Variablen, mit allem. Ferdinand Raimund war ein wunderbarer Dichter – aber »Der Verschwenderr« gehört auf die Bühne und nicht vor einen Computer. |
| Regel # 4 | Schreiben Sie ein »Lastenheft« für Ihr Programm vorhaben. Die Betonung liegt auf SCHREIBEN . Alles, was Sie sich von Ihrem Programm erwarten. Eingaben, Ausgaben, mögliche Fehleingaben abfangen usw. |
| Regel # 5 | Verwenden Sie möglichst aussagekräftige Bezeichnungen für Variablen und Programmteile. Ein Label mit der Bezeichnung »xyz_558« erinnert Sie zwar daran, daß hier das 558ste Unterprogramm kommt, für das Ihnen wieder kein besserer Name einfiel. Aber wissen Sie in zwei Wochen noch, was hier geschieht? |
| Regel # 6 | Kommentieren Sie Ihr Programm. Schreiben Sie lieber einen ganzen Roman in REM-Zeilen, als daß Sie nach drei Monaten raten müssen, was zum Kuckuck Sie wohl damals gemeint oder beabsichtigt haben könnten. |
| Regel # 7 | Bauen Sie Ihr Programm modular auf. Ein gut strukturierter Programmteil ist wesentlich wartungsfreundlicher als über das ganze Programm verstreute, immerwiederkehrende Funktionsblöcke. Und diese Module können Sie auch in anderen Programmen unproblematisch verwenden. |
| Regel # 8 | Bevor Sie noch die erste Zeile tippen: Zeichnen Sie ein Flußdiagramm Ihres Programms. Es kann auch ein Struktogramm oder eine beliebige von Ihnen bevorzugte Methode sein – aber führen Sie sich wenigstens einmal bevor Sie loslegen plastisch vor Augen, was Sie da eigentlich veranstalten wollen. Zusammenhänge und Querverbindungen in Ihrem Programm, von denen Sie nicht einmal bei Vollmond und Ebbe geträumt hätten, werden plötzlich erkennbar sein. Sie glauben's nicht? Ging uns ebenso. |
| Regel # 9 | Halten Sie sich immer die zwei letzten Entwicklungsstufen Ihres Programms in Reserve. Und dokumentieren Sie möglichst ausführlich die Unterschiede zwischen den Versionen. Oft ruft eine Änderung links unten die unglaublichsten Ergebnisse rechts oben hervor. |
| Regel # 10 | Deklariieren Sie zu Beginn eines Programmes oder eines eigenständigen Programm-Moduls die verwendeten Variablen, lokale ebenso wie globale. Manche Hochsprachen wie »C« oder »MODULA« lassen Ihnen ohnehin kaum eine andere Wahl; tun Sie's auch in einem Basic-Programm. |
| Regel # 11 | Wenn Sie Regel # 8 beherzigen (was wir in Ihrem Interesse hoffen), dann sollten Sie Ihre Flußdiagramme auch ständig überarbeiten. Es gibt nichts Tödlicheres für ein Programm-Update als unvollständige oder veraltete Unterlagen. |
| Regel # 12 | Jedes Programm braucht einen »Kopf«. Dieser Kopf sollte folgendes enthalten: <ul style="list-style-type: none"> – Programm-Name – Versions-Nummer – Erstellungsdatum der ersten Version – Datum der letzten Version – Autor – Stichworte über Sinn und Zweck des Programms – Unterschiede zur vorhergehenden Version – andere für das Programm (oder für Sie !) wichtige Daten |

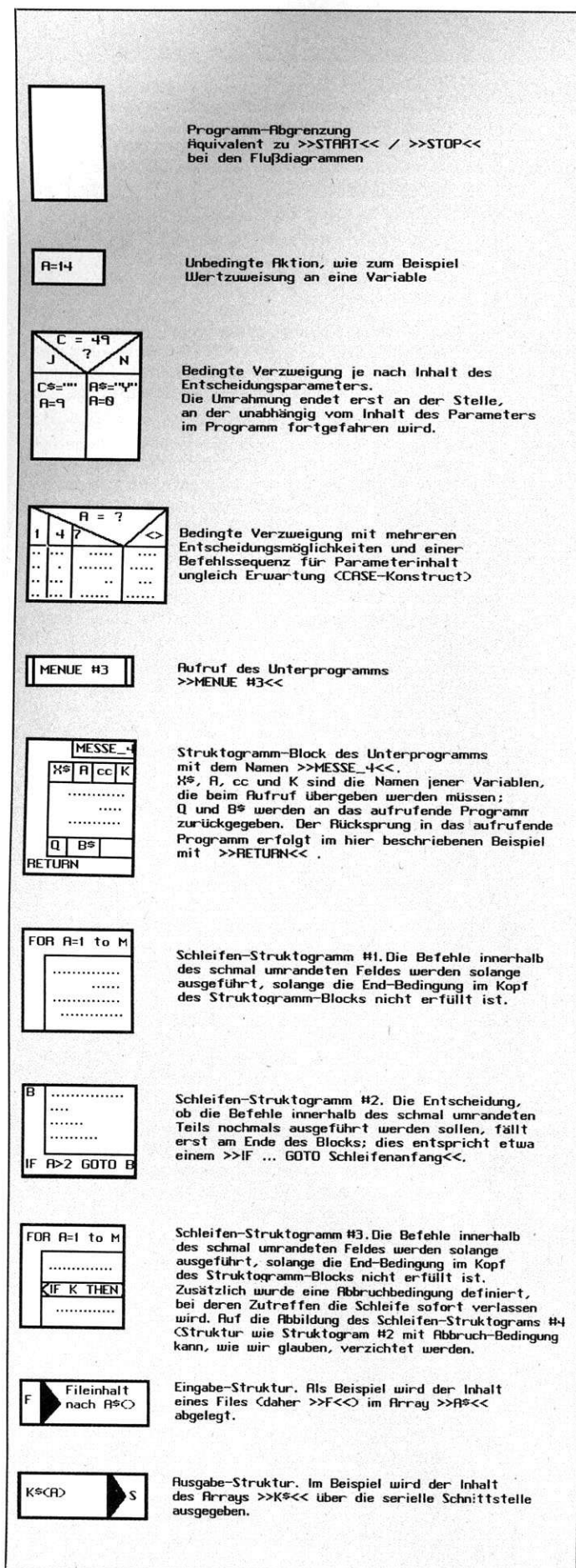


Bild 3. Symbole für Struktogramm-Verwender

ben, und Ihr Programm – was immer es auch tun soll – funktioniert (oder auch nicht), legen Sie das Listing in einen Schrank und nehmen es nach drei Monaten wieder heraus. Wenn Sie dann versuchen, den Programmfluß nachzuvollziehen, werden Sie bestimmt wieder an diesen Artikel denken und reuig das AMIGA-Sonderheft 8 hervorkramen.

Zugegeben: Wir waren auch nicht besser. Aber wenn der Tolle vor Ihnen auf die Bananenschale steigt und danach glücklich seine Versicherung anruft – steigen Sie dann auch drauf?

Genug gepredigt. Hier also, wie angekündigt, einige Regeln für den Einsteiger. Mag sein, daß das alles etwas nach »zwei Tafeln« und »Berg Sinai« klingt – aber vielleicht denken Sie bei Ihren ersten Schritten trotzdem daran. Wenn Sie die angeführten Regeln zur Basis Ihrer Programmierer-Stunden machen, werden Ihre Programme allen Ansprüchen der professionellen Programm-Dokumentation gerecht werden.

Regel ohne Ende

Wie Programme in den verschiedenen Programmiersprachen aussehen, können Sie an anderer Stelle in diesem Heft sehen (Seite 121). Was Sie noch brauchen, um die Regel #8 anwenden zu können, sind die Symbole, aus denen sich Flußdiagramm und Struktogramm zusammensetzen. Symbole für Flußdiagramme finden Sie in Bild 1, ein Beispiel dafür in Bild 2. Falls Sie es lieber mit Struktogrammen versuchen wollen –

auch gut. Bild 3 zeigt Ihnen die Symbolik und Bild 4 ein kleines Struktogramm. Bild 2 und 4 schematisieren dasselbe Programm, das folgendes tun soll: Der Benutzer wählt durch Eingabe des Namens ein ASCII-File, dessen Inhalt auf dem Drucker ausgegeben wird. Wird kein Name eingegeben, ist das Programm zu beenden. Bei Auftreten eines Fehlers ist eine Warn-Meldung auf dem Monitor auszugeben und ein neuerlicher Eingabe-Versuch zulässig. Welche Art der Programm-Schematisierung gefällt Ihnen besser?

An die Symbole brauchen Sie sich **nicht** sklavisch gebunden zu fühlen. Es gibt zwar Empfehlungen von kompetenter Stelle (gemeint ist ausnahmsweise nicht die AMIGA-Sonderheft-Redaktion), doch variiert unserer Erfahrung nach jeder Programmierer die Symbole ein wenig. Was wir Ihnen hier anbieten, sind lediglich Vorschläge und Anregungen. Die Symbol-Tabellen sind auch bei weitem nicht vollständig. Es ist eine Auswahl aus jenen Symbolen, die wir auch verwenden, soweit uns die Arbeit an den AMIGA-Sonderheften noch Zeit zum Programmieren läßt. Wie gesagt: Wichtig ist nur, daß Sie es tun. Viel Erfolg wünscht die AMIGA-Redaktion.

so

Hans Waldhäusel ist Leiter einer Fertigungsabteilung bei Philips. Mit einem C 128 als »Einstiegsdroge« stieg er bis in die Abgründe der Assemblerprogrammierung hinab. Seit einiger Zeit arbeitet er als Autor für die AMIGA-Sonderhefte. Sie erreichen ihn über unsere Verlagsadresse (Markt & Technik Verlag AG, Redaktion Sonderhefte, z. Hd. Herrn Waldhäusel, Hans-Pinsel-Str. 2, 8013 Haar bei München).

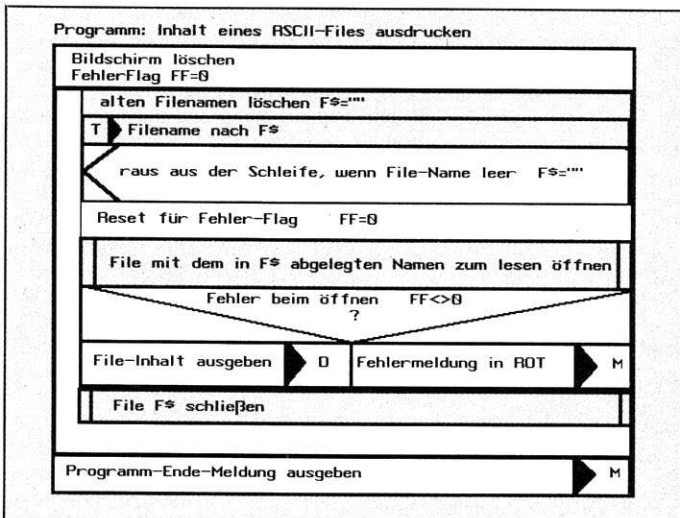


Bild 4. Auch ein Struktogramm sorgt für den nötigen Durchblick

Programmiersprachen für den Amiga

Turmbau zu Babel

von Hans Waldhäusel

Aufsteiger von Computern wie dem C64 kennen oft nur den Unterschied zwischen dem »langsamen« Basic und dem »schnellen« Assembler. Daß dies nicht die ganze Vielfalt der Computerwelt darstellt, wissen zumindest Umsteiger (etwa aus Richtung »ATA-ri«, dem ersten Scheuermittel mit Mikroprozessor). Den »echten« Einsteiger aber, voll der Neugier und bar jeden Wissens, befällt leicht Verwirrung bei all den Interpretern, Compilern, Linkern, Hochsprachen und was es so alles an mit merkwürdigen Ausdrücken gespickten Dingen auf dem Amiga-Programmiersprachen-Markt gibt. Genau damit wollen wir uns hier befassen.

Nein – dies wird nicht die zweihundertneunundfünfzigste »Programmiersprachenmarkt-übersicht« (ein schönes Wortmonster). Wir haben etwas anderes vor. Sie lesen gerade ein AMIGA-Sonderheft, das speziell für Einsteiger konzipiert wurde. Wir nehmen an, daß auch Sie, lieber Leser, der Sie jetzt diesen Artikel lesen, zu den Einsteigern zählen. Vielleicht sind Sie sogar ein frischgebackener Amiga-Besitzer, soeben mit dem Paket Ihrer Träume unter dem Arm nach Hause gekommen. Sie brauchen Information und holen sich diese aus diesem Heft. Recht so, Ihre Wahl war doppelt richtig – für den Amiga und für dieses Sonderheft. Sie haben sich auch schon etwas umgehört, was es so alles gibt für Ihren Traum-Computer: Spiele, Textverarbeitungsprogramme, Tabellenkalkulationen, Programmiersprachen, etc. Erfahrungsgemäß ist die verfügbare Informationsmenge bei Spielen am größten und geht beim Thema »Programmieren« gegen Null. Manchmal kratzen Sie sich am Hinterkopf: »Was, um alles in der Welt, ist ein Compiler?« oder »Welche Programmiersprachen gibt es eigentlich für den Amiga?« Fragen über Fragen, und der Durst nach Information wird immer schlimmer.

Wie gesagt, dieser Artikel soll keine Aufzählung von Programmen oder Markennamen

Der Weg zur »richtigen« Programmiersprache kann ein dorniger sein. Mit diesem Grundlagenartikel sehen Sie Licht am Ende des Dickichts.

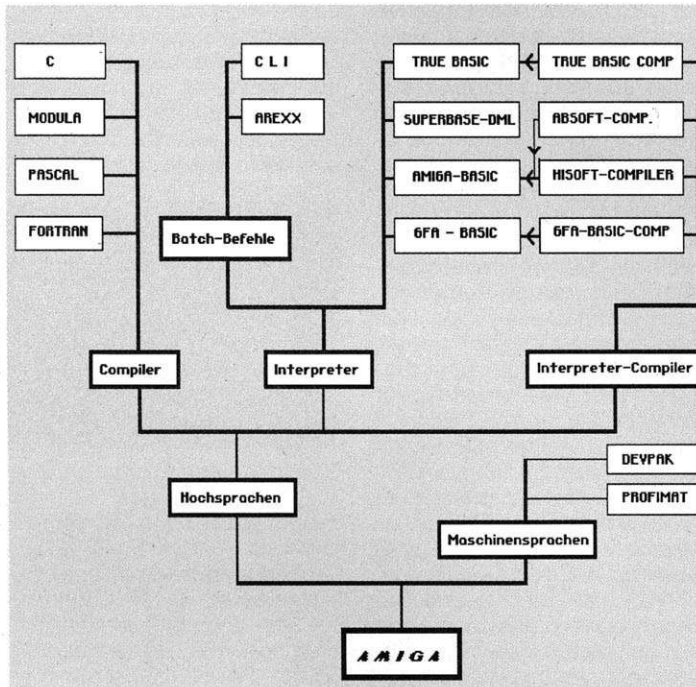


Bild 1. Der Sprachbaum gibt einen Eindruck von den vielfältigen Programmiersprachen für den Amiga

sein. Auch keine Marktübersicht mit Bezugsquellen und Richtpreisen. Wir wollen Ihnen die Hauptgruppen der für den Amiga verfügbaren Programmiersprachen und -möglichkeiten näherbringen und hoffen, Ihnen damit den ersten Kontakt mit der Amiga-Programmierung etwas zu erleichtern.

Zur Einstimmung, und um Ihnen einen kleinen Eindruck der Sprachenvielfalt auf Ihrem Amiga zu vermitteln, haben wir für Sie einen kleinen »Stammbaum« der Sprachenverwirrung entworfen (Bild 1). Die auf-

Stammbaum der Sprachen

geführten Programm-Namen sind nur als Beispiel gedacht, bedeuten keine Wertigkeit und sind auch (Gott sei Dank) bei weitem nicht alles, was es für den Amiga an Programmiersprachen gibt.

Also ran an den Baum der Erkenntnis. Wir geben zu, es ist ein etwas mißratener Baum –

keine Wurzeln, nur Stamm und Äste. Der Stamm, die »tragende Säule« des Ganzen, ist der Amiga. Das Kästchen »Amiga« symbolisiert die Hardware (Denise, Paula, 68000er, Agnus,...) und die Sprache, die der 68000er »wirklich« versteht – eine Folge von Nullen und Einsen, von gesetzten und gelöschten Bits. Wer die Anfangszeiten der Mikroprozessoren aktiv miterlebt hat (Sie haben ?? – Und da lesen Sie ein Einsteiger-Heft?), der wird sich vielleicht noch an den KIM-1 erinnern. Das war eines der ersten Mikroprozessor-Systeme, die in den »normalen« Handel kamen, aufgebaut mit ein paar Peripherie-Bausteinen auf einer nackten Platine ohne Netzteil und Massenspeicher.

Programmiert wurde der Prozessor des KIM-1 (ein 6502) in hexadezimaler Form über eine HEX-Tastatur ähnlich der eines Taschenrechners. Wenn Sie das Akkumulatorregister des Prozessor-Chips mit dem Wert 55 laden wollten, mußten Sie

die HEX-Zahlen »A9« für »LDA« und »37« für »dezimal 55« eingeben. In Anlehnung an Nestroys »Die Welt ist die wahre Schul« könnte man jetzt sagen: »Nur wer einmal in dieser echten Maschinensprache programmiert hat, weiß den Komfort der heutigen Programmiermöglichkeiten wirklich zu schätzen.«

Genug der Steinzeit – Was bietet sich auf dem Amiga? Sie sehen, wir unterteilen zuerst einmal in »Hochsprachen« und »Maschinensprachen«. Als Hochsprache bezeichnet man eine Programmiersprache, die

Ende der Steinzeit

in ihrer Syntax weitgehend der (meist englischen) Sprache entlehnt ist (somit der Lesbarkeit durch den Menschen entgegenkommt), die die Konstruktion von komplexeren Programmstrukturen wie Schleifen oder Bedingungsabfragen durch spezielle Befehle unterstützt und die einen möglichst großen Vorrat an Schlüsselbefehlen für gewisse Standardprozeduren, wie z.B. Dateneingabe, zur Verfügung stellt. »Basic« ist eine solche Hochsprache, aber dazu später mehr. Wenn Sie als Einsteiger schon einmal ein Programm in »C« gesehen und sich über die kryptischen Zeilen entsetzt haben, werden Sie uns vielleicht nicht glauben, daß auch »C« eine Hochsprache ist (von wegen »Lesbarkeit«) – aber jede Sprache muß man lernen, bevor man sie verstehen und sich durch sie und mit ihr verständigen kann.

Im Vergleich mit der Hochsprache ist die Maschinensprache schon wesentlich schwerer zu beherrschen. In ihrem ursprünglichen Sinn besteht die Maschinensprache eigentlich nur aus einer Aneinanderreihung von Binärzeichen (also Null und Eins, Ja und Nein, High und Low, usw.), die der-einst die Schaltzustände von Relais definierten. Im aktuellen Sprachgebrauch versteht man unter »Maschinensprache« die Verwendung von sog. Mnemonics, also von der natürlichen Sprache entfernt ähnelnden Abkürzungen. So ein Mnemonic für den 68000er wäre z.B.

CMPM.W (A1)+, (A5)+

»CMPM.W« bedeutet »Com-Pare Memory« – also vergleiche einen Bereich im Speicher – und das »W« steht für »Word«, also 16 Bit Daten-Breite; den Rest sollten Sie bei Bedarf in einem Assembler-Lehrbuch nachlesen.

Die zugehörige Bitfolge, die eigentlich die »richtige« Maschinensprache darstellt, sähe dann so aus:

1011101101001001

Die Übersetzung von Mne-monic in jene Zeichenfolge, die

Nahe der Baumwurzel

Ihr Prozessor wirklich verstehen kann, erledigt der »Assembler«. Wenn Ihr Freund von sich behauptet, er programmiere in Maschinensprache, so ist er entweder ein hoffnungsloser Nostalgiker, der seinen Trommelspeicher mit der Handkurbel auf Touren bringt und mit dem linken Eckzahn den Lochstreifen stanzt, oder er meint eigentlich, er programmiere in und mit einem Assembler.

Eine Erweiterung, und eigentlich der heutige Standard bei den Assemblern, sind sog. »Makroassembler«. Sie stellen die Vorstufe zur Hochsprache dar. Als »Makro« bezeichnet man einen Programmblock für eine bestimmte Aufgabe (etwa die Ausgabe einer Zeichenkette), der irgendwann geschrieben und gemeinsam mit anderen solchen »Programmmchen« (beispielsweise auf Diskette) gespeichert wurde. Eine Sammlung mehrerer solcher Programme nennt man auch »Bibliothek« oder – weil es sich um »Makros« handelt – »Makro-Bibliothek«. Dadurch, daß nahezu beliebig komplexe Programmblöcke mit einem einprägsamen Namen versehen und als Makro deklariert werden können, bietet sich dem Programmierer die Chance, sich seine eigene »Halbhochsprache« selbst zusammenzustellen. So, wie der Basic-Interpreter (nur Geduld, zum Thema »Interpreter« kommen wir noch) beim Erkennen eines Befehls in seiner »Bibliothek« nachsieht, was jetzt zu tun ist, so fügt ein »Makroassembler« beim Assemblieren eines Programms anstelle des Makro-Namens die komplette Funktion ein. Einen Grundstock solcher Makros liefern die Anbieter von Assembler-Paketen von Haus aus mit. Listing 1 ist ein Beispiel für ein

Assembler-Programm (bitte NICHT ABTIPPEN – es handelt sich nur um ein Fragment).

»Assembler« wird oft als »geschwätzige Sprache« abqualifiziert. In gewisser Weise ist das auch durchaus zutreffend; einer einzigen Basic-Zeile steht in einem Assembler-Quellcode ein ganzes Rudel von Anweisungen gegenüber. Auch die Verwaltung von Speicherbereichen und Variablen muß der Assembler-Programmierer selbst erledigen, was dem Basic-Anwender vom Interpreter abgenommen wird. Und es gibt doch weiß Gott genug Basic-Compiler und Compiler-Hochsprachen wie »C«, »Modula« usw., die für ansprechende Laufzeiten sorgen (was ein Compiler ist, lesen Sie weiter unten). Warum also »Assembler«? Einfach deshalb, weil kein Compiler den »optimalen« Code erzeugt. Nur der Assembler-Spezialist kann ein Maximum an Funktion mit einem Minimum an Speicherplatz- und Rechenzeit-Bedarf herausholen. Als Beispiel möge das »Amiga-DOS Replacement Project«, kurz »ARP« genannt, dienen, mit dem Charlie Heath und Partner erfolgreich versuchen, die größtenteils mit »C« und »BCPL« geschriebenen Befehle und Betriebssystem-Routinen Ihres Amiga durch Assembler-Programme zu ersetzen.

Die Erfolge sind bereits erkennbar: deutlich kürzere Programm-Äquivalente zu den CLI-Kommandos, schnellerer Programm-Ablauf, mehr Komfort und volle Kompatibilität. Also: **deshalb** noch immer Assembler – wenn es auch einigen Spezialisten vorbehalten bleibt, mit dem Computer in dessen Muttersprache zu parlieren.

Wenden wir uns nach diesem Ausflug in das Reich der Bits und Opcodes nun etwas mehr den Hochsprachen und deren Implementationen auf dem Amiga zu. Wie bereits erwähnt, erleichtern uns die Hochsprachen durch ihre Sprachnähe und ihre »eingebaute Intelligenz« die Verwirklichung unserer Programmiervorhaben. Der Oberbegriff »Hochsprache« umschließt auf dem Amiga drei verschiedene Untergruppen von Hochsprachen: die »Interpretersprachen«, die »Batch-Kommandos« und die »Compilersprachen«. Widmen wir uns zunächst dem Zweig »Interpreter«. Die Bezeichnung enthält bereits das charakteristische Merkmal – hier wird interpretiert. »Interpre-

tieren« bedeutet in diesem Zusammenhang so etwa: Simultan-Dolmetschen – und das trifft genau den Kern der Sache. Ein Programm, das unter der Kontrolle eines Interpreters abläuft, wird während des Ablaufs Befehl für Befehl in die Sprache des Computers übersetzt. Der Interpreter ist der Dolmetscher zwischen dem Programm-Verfasser und dem Computer. Wie geht das vor sich?

Bei jedem Programmdurchlauf wird jeder im Programm angeführte Befehl aus der Reihe der vorhandenen Befehle herausgesucht, auf mögliche Variationen untersucht, auf seine Zulässigkeit überprüft und erst dann abgearbeitet. Endet

lich gewordenen (Maschinen-) Befehle zu verzichten – daher auch der Name: RISC steht für **Reduced Instruction Set Computer** (Prozessor mit reduziertem Befehlssatz). Damit steigt zwar der Aufwand für die Programmierung, weil nur für die nötigsten Vorgänge fertige Befehle vorhanden sind, aber dafür wird die Ausführungs-geschwindigkeit drastisch gesteigert; der Mehraufwand beim Programmieren wird meist in Form von Pseudo-Befehlen aufgefangen, die genaue-nommen Makros darstellen.

Der Trick der RISC-Rechner hätte – auf einen Basic-Interpreter angewandt – nicht viel Sinn. Je kleiner der Befehlssatz, de-

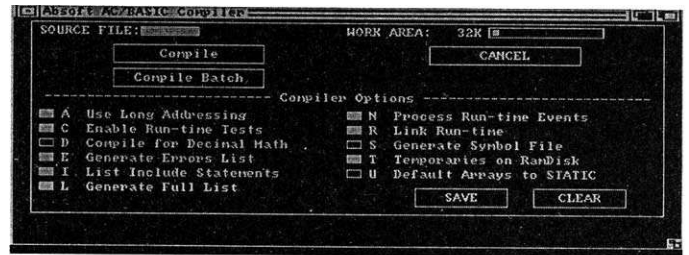


Bild 2. Vollgas für Basic ohne Interpreter

das Programm und Sie starten es erneut, so beginnt wieder die Sucherei. All diese Prüfungen kosten enorm viel Zeit. Stellen Sie sich doch bitte einmal die Sisyphusarbeit des bedauernswerten Basic-Interpreters vor, wenn er jeden auftretenden Befehl aus einer Liste von rund 240 möglichen heraussuchen muß. Hat er den Befehl endlich hervorgekramt, kommt die Überprüfung, ob der Befehl im aktuellen Programmstadium überhaupt zulässig ist. Steht der Ausführung des Kommandos noch immer nichts Grundsätzliches im Wege, ist auf eventuelle Zusätze zu achten (z.B., ob einem »PRINT« ein »#« oder ein »PTAB« folgt). Auch diese Zusätze müssen wieder auf Zulässigkeit getestet werden. Und so geht es weiter – je mächtiger der Befehlssatz einer Interpreter-Sprache ist, desto langsamer wird sie.

Ein Ast und tausend Blätter

Unter solchem »Übergewicht« leiden übrigens auch Prozessoren. Sicher haben Sie schon von dem Modewort »RISC« Prozessor gelesen. Dies hat nichts mit Risiko zu tun (meistens), sondern mit der Eigenart dieser Prozessoren, auf einen Großteil der mittlerweile üb-

sto umständlicher die Programmgestaltung. Das Basic des C64 brauchte zwei Befehle, um das Directory einer Diskette ausgeben zu können (erinnern Sie sich noch? Das Programm im Speicher war dann auch beim Teufel); das Basic des C128 hatte dagegen einen eigenen Befehl parat (paradoxi-erweise sogar zwei mit identi-scher Wirkung, nämlich »DI-RECTORY« und »CATALOG«), war also »mächtiger« – aber auch langsamer. Sie sehen, wir treffen hier wieder einmal auf die »Schwanzbeißerkatze«. Nun geht es bekanntlich gerade den Computer-Benutzern nie schnell genug und auch nie bequem genug. Kein Anwender einer Interpreter-Sprache verzichtet gern auf deren Vorteile, gleich-zeitig will er aber die Nachteile nicht hinnehmen. Was hilft aus diesem Dilemma? Ein Compi-ler muß her.

Damit wären wir bei einem Nebenzweig des Astes mit der Bezeichnung »Hochsprachen« – bei den Interpreter-Compilern. Bitte nehmen Sie den Ursprung dieses Astes in unser-m Schaubild nicht so genau – wir waren uns lange nicht im klaren, wo wir diese Programm-art ansiedeln sollen; schließlich haben wir uns dafür entschie-den, möglichst nahe an den Stamm zu gehen, ohne die Ver-bindung zu den Interpretern zu

verlieren. Darüber ließe sich sicherlich lange streiten... aber weiter im Text – der Interpreter-Compiler. Sie erinnern sich: Bequemlichkeit beim Programmieren und Austesten stehen in Widerspruch zu hoher Ablaufgeschwindigkeit des Programms. Was kann ein Interpreter-Compiler für Sie tun? Er übersetzt (ähnlich dem Interpreter) Ihr Programm – aber nur einmal. Das Ergebnis einer solchen Übersetzung ist ein direkt vom Computer ausführbares Maschinenprogramm mit entsprechender Schnelligkeit. Der Grundgedanke ist naheliegend: Wenn die Hauptbremse im Programmablauf (beispielsweise eines Basic-Programms) eigentlich nichts mit dem Zweck dieses Programms zu tun hat, sondern im Suchen und Testen von Befehlen und im Verwalten von Variablen begründet ist, was liegt da näher, als diesen Such-, Test- und Verwaltungsaufwand nur solange zu betreiben, bis das Programm ausgetestet ist und funktioniert, und danach all die zeitintensiven und nunmehr überflüssig gewordenen Aktivitäten zu eliminieren.

Best of both Worlds

Ebenso kann auch auf all jene Interpreter-Fähigkeiten verzichtet werden, die vom Programm gar nicht angefordert werden – Zeit- und Speichersparnis sind die Folge.

Genau das (und natürlich noch einiges mehr) macht ein Interpreter-Compiler. Er übersetzt die »lesbare« Klartext-Struktur des Hochsprachen-Programms in Maschinensprache und macht daraus ein vom Prozessor direkt les- und ausführbares Programm, sozusagen ohne bürokratische Umwege. Wir betonen hier absichtlich die Zusammengehörigkeit von Interpreter-Compiler und Interpreter, um keine falsche Verbindung zu den Non-Interpreter-Compilern aufkommen zu lassen; dort liegen die Dinge ein klein wenig anders. Die gängigsten Interpreter-Sprachen, die auf dem Amiga implementiert sind, haben wir in unserem »Stammbaum« gemeinsam mit ihren Compilern (soweit vorhanden) angeführt. Ein Beispiel für ein Basic-Programm und eine Bildschirm-Hardcopy des ersten brauchbaren Basic-Compilers, der für den Amiga verfügbar war, haben wir für Sie in Listing 2 und Bild 2 abge-

druckt. Der abgebildete Compiler stammt aus der Byte-Küche der Absoft-Corporation und ist zur Zeit in der Version 1.3 im Fachhandel erhältlich. In Bild 2 sehen Sie auch, daß so ein Compiler dem Benutzer weitreichende Möglichkeiten zur Beeinflussung des Endprodukts, also des compilierten Programms, zur Verfügung stellt.

Schnell noch eine Warnung: **Vorsicht** – Es führt kein Weg zurück. **Löschen Sie NIEMALS Ihr Original-Programm!** Es ist uns zur Zeit kein Programm bekannt, das in der Lage wäre, ein compiliertes Basic-Programm in seinen Urzustand, also das lesbare und vom Basic-Interpreter bearbeitbare File, zurückzuführen. Prinzipiell wäre ein »Re-Compiler« für jeden Compiler zwar denkbar, aber für einen Amiga-Basic-Compiler gibt es eben noch keinen. Und noch ein Nachteil (wenn es einer ist) sei an dieser Stelle erwähnt. Es kann Ihnen passieren, daß aus einem Basic-Programm mit 20 Zeilen plötzlich ein compiliertes File mit 30 K Länge wird. Dies liegt daran, daß der Compiler (wenn Sie es von ihm verlangen) sämtliche benötigten Routinen für Bildschirmausgabe, etc. an Ihr Programm anhängt. Dies können Sie umgehen, wenn Sie mehrere compilierte Programme auf einer Diskette haben; dann reicht es nämlich, das sog. »Runtime«-Modul auf dieser Diskette abzulegen. In diesem Modul sind dann die Routinen enthalten, die nicht in Ihrem Programm stehen, die Ihr Programm in seiner compilierten Form aber braucht. Der Vorteil eines getrennten »Runtime«-Files besteht darin, daß diese Routinen von mehreren Programmen verwendet werden können, was bei vielen compilierten Files schon eine ganze nette Platzersparnis auf der Diskette bedeuten kann. Natürlich ist bereits bei der Compilierung anzugeben, ob mit oder ohne Runtime-Library gearbeitet werden soll. In Bild 2 sehen Sie in der rechten Hälfte einen »Schalter« mit der Bezeichnung »Link Run-time« – das »isser«.

Bevor wir die Interpreter-Sprachen verlassen, möchten wir noch eine kleine Ergänzung anbringen: Basic ist nicht die einzige Interpreter-Sprache auf dem Amiga. Sie ist lediglich die bekannteste.

Stellvertretend für alle Non-Basic-Interpreter steht in diesem Artikel und in unserem Sprachen-Baum die Programm-

Sprache »DML«, die von Precision Software zur Steuerung der Datenbank »Superbase Professional« entwickelt wurde (ein Beispiel sehen Sie in Listing 3). Allerdings gibt es für DML noch keinen Compiler. Der Erfolg von Superbase sollte Precision Software durchaus zur Entwicklung eines solchen Beschleunigers ermutigen. Und noch ein kleiner Hinweis erscheint uns geboten: Im »Sprachbaum« sehen Sie einen Compiler zum GFA-Basic-Interpreter. Dieser Compiler wird zwar bereits seit längerer Zeit beworben, doch hatten wir noch keinen in der Hand: etwas Geduld, bitte...

Steuern per Script

Auf zum nächsten Ast, den »Batch-Kommandos« und »Batch-Sprachen«. Die sind eine Klasse für sich. Ein Programm, das diese Kommandos verwendet, nennt man eine »Batch-Datei«. »Programm« ist eigentlich zu hoch gegriffen. Eine Batch-Datei ist eine ASCII-Datei, in der Anweisungen in der gewünschten Reihenfolge angeführt sind. Auch Kommentare, ähnlich den »REMS« in Basic, sind erlaubt. Wenn Sie jetzt eine Analogie zu Basic sehen, haben Sie beinahe recht. Diese Batch-Kommandos sind so eine Art »zerstückelter Interpreter«, bloß brauchen Sie zur Abarbeitung einer Batch-Datei keinen speicherfressenden Interpreter zu laden. Also eine Analogie zu einem durch einen Compiler gejagten Basic-Programm? Auch nicht ganz. Die in einer Batch-Datei angeführten Befehle sind eigenständig ablauffähige Programme, die in der Reihenfolge ihres Erscheinens in der Batch-Datei aufgerufen werden. Das Tolle an diesen Dateien ist, daß sie programmübergreifend sind. Damit können Sie verschiedene Programme starten und während oder auch nach Beendigung des einen ein zweites Programm aktivieren. Selbst der Aufruf einer Batch-Datei aus einer Batch-Datei ist möglich. Und seit »PIPE-Handler« und »ARexx« können auch Daten von einem Programm zum anderen übergeben werden (sofern diese dafür ausgelegt sind, wie z.B. »CygusEd« und »Cape 68K« für »ARexx«).

Wie beim Basic-Interpreter unterscheiden wir auch bei den Batch-Kommandos zwischen dem »Direkt-Modus«, also der Eingabe von Befehlen nach

dem CLI-Prompt, und dem »Programm-Modus«, der in diesem Fall durch Aufruf und Abarbeitung einer Stapel- oder Batch-Datei initiiert wird. Die Befehle, die Ihnen zur Verfügung stehen, finden Sie im Verzeichnis »c« Ihrer System-Diskette und nähere Erklärungen im Handbuch, das Sie zu Ihrem Amiga beigegepackt bekamen.

Viele Befehle, die Sie dort finden, sind erst in einer Batch-Datei, also im »Programm-Modus« sinnvoll anzuwenden; was sollten Sie auch mit einem »ASK«-Befehl oder einem »WAIT«-Kommando im Direktmodus anfangen? In einer Stapel-Datei wie z.B. der Startup-Sequence sieht das schon ganz anders aus. Wie wär's mit einer Abfrage, ob die Workbench geladen werden soll oder der CLI aktiviert bleibt? Das geht ganz einfach. Laden Sie einen Texteditor (beispielsweise den »MicroEmacs«) und fügen Sie in das File »Startup-Sequence« – es liegt im Verzeichnis »s« Ihrer Boot-Diskette – vor der Stelle, wo »LOADWB« steht, die folgenden Zeilen ein:

```
ASK "Soll ich die Workbench laden ? [Y/N]
<RETURN> N"
IF WARN LOADWB ENDIF

»IF«, »LOADWB« und »ENDIF« sind Programmfiles im »c«-Directory. Eine Datei namens »WARN« werden Sie allerdings vergeblich suchen – »WARN« ist eine Bedingung, die vom Betriebssystem erkannt wird. Speichern Sie die geänderte Startup-Sequence. Beim nächsten System-Start wartet Ihr Amiga auf Ihre Entscheidung.
```

Start nach Belieben

Drücken Sie <N> <RETURN> (nur <RETURN> reicht auch), so teilen Sie dem Amiga mit, daß Sie an der Workbench nicht interessiert sind und sich Ihren Speicher für andere Dinge aufsparen wollen. Entscheiden Sie sich für die Workbench und tippen <Y> <RETURN>, so wird die Workbench in den Speicher geholt und aktiviert – könnte hilfreich sein, oder? Arbeiten Sie stets mit einer Kopie Ihrer System-Diskette; eine durcheinandergebrachte Startup-Sequence hat schon so manchem Einsteiger den Frust des Jahres beschert. Übrigens noch ein kleiner Tip, bevor wir uns zum nächsten Ast des Sprach-Bau-

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Redaktionsdirektor: Richard Kerler

Chefredakteur: Hans-Günther Beer

Stellv. Chefredakteur: Gottfried Knechtel (kn) - verantwortlich für den redaktionellen Teil

Chef vom Dienst: Susanne Kirmaier

Redaktion: Klaus Sonnenleiter (so), Albert Petryszyn (pe)

Mitarbeiter dieser Ausgabe: Brigitte Bobenstetter, Dittmar Inäbni, Martin Jobst, Andreas Regul, Michael Thomas, Hans Waldhäusel

Redaktionsassistent: Brigitte Bobenstetter, Sylvia Derenthal, Helga Weber (202)

Telefax: 089/4613-778, Hotline: Donnerstag 14 bis 17 Uhr

Alle Artikel sind mit dem Kurzzeichen des Redakteurs und/oder mit dem Namen des Autors/Mitarbeiters gekennzeichnet

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt & Technik Verlag AG verlegten Publikationen. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Art-director: Friedemann Porscha

Titelgestaltung: Rolf Boyke

Layout: Alexander Kowarzyk (Chefflyouter), Andrea Miller

Bildredaktion: Janos Feitser (Litg.), Sabine Tennstaedt; Roland Müller (Fotografie); Ewald Standke, Norbert Raab (Spritzgrafik); Werner Nienstedt (Computergrafik)

Bildnachweis: Südde. Zeitung (Titelmotiv)

Anzeigenleitung: Alicia Clees (313) - verantwortlich für die Anzeigen

Telefax: 089/4613-775

Anzeigenverwaltung und Disposition: Monika Burseg (147)

Auslandsrepräsentation:

Anzeigenverkaufsleitung Ausland: Thomas Schlüter (299)

Auslandsniederlassungen:

Schweiz: Markt & Technik Vertriebs AG, Kollerstr. 37, CH-6300 Zug, Tel. 042-440550/660, Fax 042-415770, Telex: 862329 mut ch

USA: M&T Publishing Inc., 501 Galveston Drive Redwood City, CA 94063, Telefon: (415) 366-3600, Telex 752-351

Österreich: Markt & Technik Ges. mbH
Große Neugasse 28, A 1040-Wien
Telefon: 0222/5871393, Telex: 047-132532

Anzeigen-Auslandsvertretung:

England: F. A. Smyth & Associates Limited, 23a, Aylmer Parade, London, N2 0PQ.
Telefon: 0044/13405058, Telefax: 0044/13419602

Israel: Baruch Schaefer, Haeskel-Str. 12, 58348 Holon, Israel, Tel. 00972-3-5562256
Taiwan: Aim International Inc., 4F-1, No. 200, Sec. 2, Hsin-I Rd.; Taipei, Taiwan, R.O.C., Tel. 00886-2-7548631, -7548633, Fax 00886-2-7548710

Vertriebsleitung: Helmut Grünfeldt (189)

Vertrieb Handelsaufgabe: Inland (Groß-, Einzel- und Buchhandelsbuchhandel) sowie Österreich und Schweiz: ip Internationale Presse, Hauptstätter Straße 96, 7000 Stuttgart 1, Tel. 0711/6483-110

Bezugsmöglichkeiten: Leser-Service: Telefon (089) 4613-366. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen.

Preis: Das Einzelheft kostet DM 16,-

Produktion: Technik: Klaus Buck (Litg./180), Wolfgang Meyer (Stellv./887); Herstellung: Otto Albrecht (Litg./917)

Druck: SOV Graphische Betriebe, Laubanger 23, 8600 Bamberg

Urheberrecht: Alle in diesem Heft erschienenen Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen, vorbehalten. Reproduktionen, gleich welcher Art, ob Fotokopie, Mikrofilm oder Erfassung in Datenverarbeitungsanlagen, nur mit schriftlicher Genehmigung des Verlages. Aus der Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen oder verwendeten Bezeichnungen frei von gewerblichen Schutzrechten sind.

Haftung: Für den Fall, daß in diesem Heft unzutreffende Informationen oder in veröffentlichten Programmen oder Schaltungen Fehler enthalten sein sollten, kommt eine Haftung nur bei grober Fahrlässigkeit des Verlages oder seiner Mitarbeiter in Betracht.

Sonderdruck-Dienst: Alle in dieser Ausgabe erschienenen Beiträge sind in Form von Sonderdrucken zu erhalten. Anfragen an Reinhard Jarczok, Tel. 089/4613-185, Fax 4613-776.

© 1989 Markt & Technik Verlag Aktiengesellschaft

Vorstand: Otmar Weber (Vors.), Bernd Balzer, Richard Kerler

Leiter Produktmarketing: Hans-Günther Beer

Direktor Zeitschriften: Michael M. Pauly

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen: Markt & Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon 089/4613-0, Telex 522052, Telefax 089/4613-100

Telefon-Durchwahl im Verlag: Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089/4613 und dann die Nummer, die in den Klammern hinter dem jeweiligen Namen angegeben ist.

mes schwingen: Wenn Sie den Artikel »Von Fenstern, Mäusen und Konsolen« auf Seite 28 gelesen haben, wissen Sie, daß Sie mit

EXECUTE Batch-Datei

einen Stapel abarbeiten lassen können, der sich im aktuellen Directory befindet und »Batch-Datei« genannt wurde. Aber wußten Sie auch, daß Sie sich sogar das »EXECUTE« ersparen können, wenn Sie mit der Workbench 1.3 das Script-Flag für Ihre Stapel-Datei setzen:

```
protect s/Batch-Datei s  
add
```

Jetzt genügt der Aufruf:

Batch-Datei

um den Inhalt der Stapel-Datei ausführen zu lassen.

Um die Batch-Sprachen nicht zu benachteiligen (sie sind wirklich nicht als »unwichtig« einzustufen), haben wir in Listing 4 ein Stückchen »Startup-Sequence« abgedruckt. Dieses »Stückchen« fragt ab, ob externe Laufwerke angeschlossen sind und weist ihnen bei Vorhandensein einen größeren als den vorgesehenen Pufferspeicher zu - vielleicht können Sie diesen Tip gebrauchen.

»Da war doch noch ein Ast.« Folgen Sie uns zu dem Hochsprachen-Ast, der sich mit den »echten« Compiler-Sprachen befaßt. Dies sind Programmiersprachen, bei denen ohne Compiler nichts läuft. »C«, »Modula«, »Pascal«, und wie sie alle heißen, sind Vertreter dieser Gattung. Bei ihnen gibt es weder Direktmodus noch Ablauf unter der Kontrolle eines »Steuerprogramms« (sprich Interpreter). Der Unterschied zu einem Interpreter läßt sich am deutlichsten so beschreiben: Wenn Sie ein Basic-Programm schreiben, brauchen Sie zu dessen Ablauf immer den Basic-Interpreter (einzige Ausnahme: kompilierte Basic-Programme); die Befehle Ihres Programms werden bei jedem Durchlauf vom Interpreter Schritt für Schritt in die Sprache des Prozessors übersetzt. Schreiben Sie hingegen ein Programm für »Modula« und compilieren dieses, so können Sie nach dem Übersetzen Ihren Modula-Compiler im Schrank lassen. Beim Compilieren wird - ähnlich den Basic-Compilern - das Programm nur einmal übersetzt und liegt dann in maschinenlesbarer Form vor. Sie bekommen also als Ergebnis ein eigenständiges Programm. Allerdings kann dieses eben nur

das, was Sie ihm in Ihrem Quellcode beigebracht haben. Wenn Sie dagegen einen Interpreter laden, so haben Sie (fast) alle Fähigkeiten dieses Interpreters im Speicher und können durch eine einfache Änderung im Quelltext all diese Fähigkeiten unmittelbar nutzen. Ein Compiler-Programm in »C« (»C« nur deshalb, damit nicht der Eindruck entsteht, wir hätten etwas gegen »Modula«) können Sie nur im Quelltext ändern und müssen es erneut compilieren, ehe Sie diese Programmänderung ausführen können.

Was ist nun das Tolle an diesen »Hochsprachen-Compilern«, daß sie in aller Munde sind und meist über den grünen Klee gelobt werden? Sie kennen doch diese Bemerkung: »Ach - in Basic programmiert Du?! Kannst Du denn keine richtige Programmiersprache?« Was hebt diese Programmiersprachen »über« einen Basic-Interpreter?

Es sind dies neben vielen anderen Vorteilen die erzielbare Ausführungsgeschwindigkeit der Programme (da verblaßt meist auch der Basic-Compiler); die Möglichkeit, alle Betriebssystem-Routinen des Amiga relativ einfach zu nutzen; die Portabilität (Übertragbarkeit) der Programme auf Quelltext-Ebene (Sie finden viele Unix-Programme auf Public Domain, die auf den Amiga übertragen wurden). Und es gilt als DIE professionelle Art, den Amiga zu programmieren, ohne in die Abgründe der Assembler-Programmierung hinabzusteigen.

Es sei allerdings an dieser Stelle nicht verschwiegen, daß die Übersetzungen der Hochsprachen-Compiler nicht immer den optimalen Code liefern. Das Entwickeln eines noch kompakteren Code bleibt wohl den ausgefuchtesten Spezialisten vorbehalten, die mit Taktzyklen aufstehen und mit Mnemonics schlafen gehen.

Für Spezialisten gedacht ist auch die Fähigkeit mancher Compiler-Hochsprachen, auf Wunsch ein Assembler-Quelltext-File zu generieren, das von den »wahrhaft Wissenden« nachgebessert werden kann, bevor das Programm zusammengebraut wird.

Schließen Sie nicht von der Kürze des Absatzes über Compiler-Sprachen auf deren Bedeutung - das Betriebssystem Ihres Amiga wurde hauptsächlich in »C« und »BCPL« (einem Vorläufer von »C«) geschrieben und keineswegs in »Basic«. Als

Abschluß haben wir in Listing 5 ein Stück aus einem C-Programm abgedruckt.

Bleibe noch ein häufiges Mißverständnis zu klären: Es fällt schon schwer genug, ein Urteil zu fällen, welches die leistungsfähigste Programmiersprache ist. Eine Empfehlung, welche Programmiersprache am Anfang am besten geeignet ist, wollen wir jedoch beim besten Willen nicht geben.

- Dennoch möchten wir Ihnen einige kleine Tips nicht ersparen: Sehen Sie sich alles in Ruhe an, bevor Sie sich für »Ihre« Sprache entscheiden.

- Die preiswerteste Variante wird wohl auch weiterhin Basic bleiben. Diese Programmiersprache ist auf Ihrer Extras-Diskette enthalten.

- Halten Sie sich anfangs von Assembler fern - diese Sprache ist eng an die Fähigkeiten

der Maschine angepaßt, aber der menschlichen Denkmöglichkeit kommt sie nicht entgegen.

- In verschiedenen Public-Domain-Serien finden Sie leistungsfähige Sprachen, die zum (fast) kostenlosen Ausprobieren geradezu einladen.

- Je besser eine Sprache strukturiert ist und je mehr Sie den Programmierer zwingt, Konventionen einzuhalten, desto besser ist sie für den Anfänger geeignet.

Wir hoffen, diese Übersicht hilft Ihnen bei Ihren ersten Programmier-Schritten oder Kaufentscheidungen. Uns bleibt nur, Ihnen viel Erfolg beim Programmieren zu wünschen, ganz gleich, für welche Sprache Sie sich entscheiden. Und wenn Sie der Meinung sind, daß Ihnen etwas Tolles gelungen ist - wir freuen uns über jede Zusendung. so

```
OPEN FILE "AMIGASoft"
```

```
REQUEST "REPORT TO PRINTER?", "", 1, a%
IF a% THEN PRINT;
REPORT
```

```
HEADING
```

```
?
? UL OFF ;@18; ATTR OFF ;"AMIGA-SOFTWARE"
?
END HEADING
```

```
SELECT UL OFF ;@2; ATTR OFF ;Programm.AMIGASoft;@33;&25 Programm_
Typ.AMIGASoft;@65;Programm_Code.AMIGASoft NEWLINE
@2;Diskette.AMIGASoft;@33;Kurzbeschr.AMIGASoft;@65;Comp_Ass.AMIGA-
Soft NEWLINE
```

```
CLOSE FILE "AMIGASoft"
```

Listing 3. Auch »DML« ist eine Interpreter-Sprache

```
Failat 11
assign >NIL: DF1: exists
IF NOT WARN
    Addbuffers df1: 25
    echo "Buffer added to Drive #1"
ENDIF
;
assign >NIL: DF2: exists
IF NOT WARN
    Addbuffers df2: 25
    echo "Buffer added to Drive #2"
ENDIF
Failat 10
```

Listing 4. Batch as batch can

```
/* ZeigeLibs.c
#include <stdio.h> */
#define VERSION 0
char *libname[15]={"exec","elist",
"graphics","layers","intuition",
"mathffp","mathieedoubbas",
"mathtrans","dos","translator","icon",
"diskfont","ramlib","info","version"};
main()
{
    char string[25];
    int loop;
    long Base;
    printf("\fZeigeLibs / Listing aus AMIGA-Magazin 2/88\n");
    printf(" zeigt verfügbare Libraries:\n\n");
    printf("1.) ROM : je nach KICKSTART\n");
    printf("2.) LIBS: falls nicht im ROM,");
    printf(" sucht das System in LIBS:\n");
    printf(" Aenderung mit ASSIGN");
    printf(" ist moeglich.\n\n");
    printf(" Library-Name");
    printf(" Startadresse\n");
    for (loop=0; loop < 15; loop++)
    {
        *string='0';
        strcpy(string,libname[loop]);
        strcat(string,".library");
        Base=OpenLibrary(string,VERSION);
        if (Base !=0)
        {
            printf("%24s :%7ld=%6lx",string,Base);
            printf(" (hex)\n");
            CloseLibrary(Base);
        }
        else printf("%24s : nicht da!\n",string);
    }
}
```

Listing 5. Mit »C« beherrschen Sie Ihren Amiga

```
start:
    jsr    opendos        ;DOS.Library öffnen
    jsr    loadhiscore    ;Highscore einladen

    jsr    allocmem       ;Speicher holen

    jsr    setgfx         ;Grafiken vorbereiten

    move.l $4,a6
    jsr    -132(a6)       ;Multitasking aus

    jsr    makecop        ;Copperliste erstellen
    move.l copadr,$dff080 ;Copperzeiger auf neue Liste
    clr.w  $dff088        ;und starten

    move.w #0,level       ;Levelnummer = 0
    move.w #1,ausflag

    move.l #7,mt_anfangpart
    move.l #29,mt_endepart ;erste Musik setzen
    jsr    mt_init
    move.w #0,mt_sub
    move.w #0,musiciconoff
```

Listing 1. Wenn Sie so etwas sehen: Alles deutet auf Assembler hin

```
FOR X=1 TO 6
    READ Hilf$
    FOR T=1 TO 2
        OPEN "R",1,"Highscore/"+Hilf$,27
        FIELD #1,20 AS Nam$,2 AS Versch$,5 AS Zeit$

        FOR I=1 TO 5
            LSET Nam$="The Famous Guy"
            LSET Versch$=MKI$(3000)
            LSET Zeit$="59:59"
            PUT #1,I
        NEXT I

        Hilf$=Hilf$+".Zeit"
        CLOSE
    NEXT T
NEXT X
END

DATA Standard,Standard II,Megabild,Elefant,Auto,Klassik
```

Listing 2. Basic haben Sie - worauf warten Sie noch?

von Hans Waldhäusel

Big Brother is

Es ist 3 Uhr 20 nachts. Eine gequälte Kreatur richtet sich stöhnend auf. Schmerzhafte Fingerkuppen, rechteckige Augen (Diagonale 14 Zoll), im Blut mehr Koffein als rote Blutkörperchen. Acht Seiten Listing abgetippt – hoffentlich fehlerlos. Jetzt noch schnell das tolle Programm anwerfen, sich freuen und ab ins Bett. Laden ... starten ... »UNDEFINED SUBPROGRAM« – da werden Aggressionen wach! Also doch ein Fehler! Aber wo? Und ist es der einzige? Nagen-de Ungewißheit, grenzenloser Frust! – Tja, der Gute hatte keinen Checksummer.

Ähnlichkeiten mit lebenden Personen sind natürlich rein zufällig, aber damit es Ihnen nicht so ergeht wie unserem fiktiven Helden, haben wir für Sie eine Eingabehilfe entwickelt. Den Checksummer »CHECKIE42_V1.1«. »Checkie42« kann Ihnen zwar nicht die Knochenarbeit des Eintippens abnehmen – so weit sind wir noch nicht (kann ja noch kommen); aber »Checkie42« kann Sie auf Fehleingaben aufmerksam machen. Na, wenn das nicht beruhigend ist.

Damit Sie den Checksummer identifizieren können, haben wir eine Hardcopy der Version 1.1 in Bild 1 abgebildet.

Abtipp-Assistent

V1.1 ist der Letztstand unserer Eingabehilfe. Um Mißverständnissen vorzubeugen, haben wir für Sie auch eine Abbildung der Version 1.0 abgedruckt (Bild 2). Nach Möglichkeit sollten Sie die neueste Version verwenden; der Komfort ist ungleich größer.

Wie funktioniert dieser Checksummer? Und wie kann er Sie vor Nervenschäden und Wutausbrüchen bewahren? Darauf gibt es mehrere Antworten. Zuerst einmal brauchen Sie das Programm »Checkie42_V1.1« in Ihrem Amiga. Entweder Sie kopieren es von einem Freund oder Sie kaufen sich doch einmal eine Leserservice-Diskette (da ist er immer mit drauf!). Oder Sie tippen Listing 1 ganz vorsichtig und gewissenhaft ab, eventuell unter Verwendung der Version 1.0 unseres Checksummers. Wie Sie unschwer erkennen, ist »Checkie42« ein Basic-Programm, doch ist seine

Anwendung nicht auf Basic-Programme beschränkt. Sie können jedes (!) in Zeitschriften von Markt & Technik veröffentlichte Amiga-Programm mit diesem Checksummer eingeben, wenn das Programm mit Prüfsummen versehen ist. Also auch Programme in »C«, »Modula«, »Assembler«, usw.

Woran erkennen Sie, ob eine Checksumme angegeben ist? Eine Checksummer-Programmzeile hat folgenden typischen Aufbau:

```
379 MJ2 IF Checkfile THEN
Zeilennummer Checksumme
Programmzeile
```

Zeilennummer und Checksumme sind meist grau unterlegt, damit Sie sie leichter von der eigentlichen Programmzeile unterscheiden können. Die Checksumme kann auch nur zwei Stellen haben; die Zeilennummer ist auf maximal vier Stellen begrenzt. Sie sehen also: die kryptischen Zeichen zwischen Zeilennummer und Programmzeile, über die Sie sich vielleicht schon gewundert haben, sind das Kennzeichen dafür, daß ein Programm mit dem Checksummer eingegeben und/oder überprüft werden kann. Wundern Sie sich bitte nicht über das »und/oder«. Es gibt nämlich mehrere Möglichkeiten, »Checkie42« anzuwenden. Da wäre zunächst einmal die Möglichkeit, sozusagen »unter Aufsicht« ein Programm einzutippen. Dazu laden und starten Sie Amiga-Basic und danach den Checksummer. Das erfolgt entweder durch Anklicken des Checksummer-Icons, wenn Sie ein Workbench-Fan sind, oder aus dem CLI. Klicken Sie das Icon an, so prüft das Betriebssystem zunächst, ob Amiga-Basic erreichbar ist. Dazu wird die Pfad- und Dateiangabe aus dem »Default-Tool«-Eintrag des »info«-Files von »Checkie42« herangezogen. Von all diesen Dingen merken Sie – außer den Diskettenzugriffen – nichts. Für den Fall, daß Sie das CLI der Workbench vorziehen, gehen Sie folgendermaßen vor: Sie wechseln zuerst einmal in das Verzeichnis, in dem Sie Amiga-Basic abgelegt haben. Dann starten Sie Ami-

ga-Basic und, wenn Sie wollen, den Checksummer gleich hinterher. Dazu geben Sie nach dem CLI-Prompt folgende Zeilen ein:

```
cd BASIC-DISK:
AMIGA-BASIC ram:CHECKER/
Checkie42_V1.1
```

Das gilt natürlich nur unter der Voraussetzung, daß Sie Amiga-Basic unter dem Namen »AMIGA-BASIC« auf Ihrer Diskette namens »BASIC-DISK« im aktuellen Laufwerk abgelegt und den Checksummer in die RAM-Disk ins Verzeichnis »CHECKER« kopiert haben. Das wär's dann auch schon gewesen; »Checkie42_V1.1« steht zu Ihrer Verfügung.

Aufruf

Nachdem Sie das Programm gestartet haben, werden Sie nach dem Namen der zu bearbeitenden Datei gefragt, wobei übrigens auch Device- und Pfadangaben nach den Konventionen von Amiga-Basic zulässig sind. Ihre Beantwortung dieser Frage stellt die Weichen für den weiteren Programmablauf. Geben Sie nämlich einen Dateinamen ein, der schon existiert, dann nimmt der Checksummer an, daß Sie diese Datei editieren wollen. Andernfalls wird ein File mit dem gewünschten Namen neu angelegt. Als nächstes gibt Ihnen »Checkie42« eine Zeilennummer vor, logischerweise zu Programmbeginn die Nummer »Eins«, und erwartet von Ihnen die Eingabe einer Checksumme.

Sind Sie mit der vorgeschlagenen Zeilennummer nicht einverstanden oder verlangt das abzutippende Listing eine andere, so nehmen Sie bitte jetzt durch einen Druck auf <F7>

**Nobody is perfect
macht Fehler. Das Abtipp-
Listings kann schon**

und Eingabe der gewünschten Zeilennummer eine Korrektur vor. Überlegen Sie sich das aber genau, denn die Zeilennummer ist selbstverständlich Bestandteil der Prüfsumme. Ganz gleich, ob Sie jetzt die Zeilennummer verändert haben oder nicht, als nächstes ist die Checksumme dran. Die geben Sie bitte exakt so ein, wie sie im Listing steht. Großbuchstaben müssen Großbuchstaben bleiben, Kleinbuchstaben Kleinbuchstaben! Sollte Ihnen bei der Eingabe der Prüfsumme ein Tippfehler unterlaufen und Sie bemerken ihn erst, nachdem Sie die Checksummer-Eingabe mit <RETURN> verlassen haben, brauchen Sie bloß auf <F1> zu drücken – damit wird die alte Prüfsumme gelöscht und der Cursor in das Prüfsummen-Eingabefeld gesetzt. Nach Eingabe der Prüfsumme springt der Cursor immer in jenes Eingabefeld, in dem Sie die eigentliche Programmzeile eintragen – aber **ohne Zeilennummer und Prüfsumme!** Verlassen Sie dieses Eingabefeld durch einen Druck auf die <RETURN>-Taste, so überprüft der Checksummer, ob die angegebene Checksumme sich mit der Kontrollrechnung deckt, wobei diese Rechnungen sowohl die Zeilennummer als auch die Programmzeile erfassen. Ist alles okay, übernimmt »Checkie42« die Zeile in die geöffnete Datei, erhöht die vorgeschlagene Zeilennummer um Eins und wartet erneut auf Ihre Eingabe von Checksumme oder Zeilennum-

Der Aufbau einer Prüfsumme

Eine Prüfsumme (auch Checksumme genannt) besteht aus zwei oder drei alphanumerischen Zeichen. Zugelassen und verwendet werden die Großbuchstaben »A« bis »Z«, die Kleinbuchstaben »a« bis »z« und die Ziffern »0« bis »9«. Sonderzeichen wie »Ä«, »ß« usw. sind nicht vorgesehen.

Die ersten beiden Zeichen der Prüfsumme enthalten den Prüfcode für Zeilennummer und Programmzeile. Das dritte Zeichen enthält den Code für die Spaltenposition der ersten »Nicht-Leer-Stelle« der aktuellen und der folgenden Zeilen.

der Checksummer

checking You

und wer arbeitet,
tippen seitenlanger
in Arbeit ausarten.

mer. Wird hingegen eine Abweichung entdeckt, so heißt das, daß entweder eingegebene Checksumme oder Zeilennummer oder Programmzeile nicht mit den Vorgaben übereinstimmen. In diesem Fall springt der Cursor an den Anfang der Programmzeile und zeigt Ihnen dadurch an, daß »irgendetwas«

stimmt; denkbar wäre eine Kommentarzeile, die Sie bereits bei der Dateneingabe etwas aussagekräftiger oder vielleicht auch kürzer machten. Auch dafür ist unser Checksummer gerüstet.

Mit einem Druck auf <F6> wird die Zeile in jedem Fall gespeichert, ganz gleich, ob Prüfsumme und Kontrollrechnung zusammenpassen oder nicht. Da solche Manipulationen aber nicht unbeträchtliche Risiken in sich bergen, werden Sie nach Beendigung der Daten-Eingabe

vor der Tastatur sitzen, müssen wir uns mit dem zu Beginn dieses Artikels kurz angesprochenen Fall »Editieren eines bereits vorhandenen Files« befassen. Auch diese Funktion unseres Checksummers ist auf den Benutzer zugeschnitten, um ihm möglichst viel an Arbeitserleichterung zu bieten.

Gehen Sie bitte folgendermaßen vor: Laden und starten von Amiga-Basic und »Checkie 42_V1.1« wie zu Beginn beschrieben. Auf die erste Frage »Dateiname ?« antworten Sie mit der Eingabe des Namens jenes Files, das Sie bearbeiten wollen. Nach einer Kontrolle, ob das angeforderte File auch tatsächlich vorhanden ist, folgt die nächste Frage nach »Nur Checksummen ausgeben (j/n) ?«. Antworten Sie auf diese Frage bitte mit der Eingabe von <n>. Auch die nächste Frage, nämlich »Eingabe aus Datei (j/n) ?«, beantworten Sie mit <n>. Das Programm erkennt daran, daß Sie die Eingabe eines bestehenden Programms fortsetzen wollen, ermittelt die Anzahl der im Programm bereits vorhandenen Zeilen und

Neustart als nächste Zeilennummer »96« vorgeschlagen wird, obwohl Sie bei Zeile »101« weitermachen sollten – an dieser Stelle sollten Sie höllisch aufpassen.

Sie wollen jetzt sicher wissen, was passiert, wenn Sie die oben angesprochenen Fragen mit <j> statt mit <n> beantworten. Nun, in diesem Fall benutzen Sie Eigenschaften des Checksummers, die eigentlich für jene Benutzer gedacht sind, denen »Checkie42« bei der Daten-Eingabe zu langsam oder zu unkomfortabel ist und die ihre Programm-Dateien lieber mit einem echten Text-Editor abfassen. Der Checksummer kann jede ASCII-Programm-Datei auch nachträglich bearbeiten. Erlauben Sie uns bitte, kurz zu erklären, was eine ASCII-Datei ist. Darunter versteht man eine »reine« Text-Datei, in der also keinerlei Steuerzeichen oder Formatierungsanweisungen enthalten sind.

Überlistet

Eine solche ASCII-Datei kann auch von Amiga-Basic angelegt werden, wenn ein Programm mit der Endung ».a« abgespeichert wird (als Beispiel: SAVE "Checksummer_V0", a). Auch jedes bessere Textverarbeitungsprogramm bietet die Möglichkeit, »Text-Dateien« abzuspeichern. Antworten Sie auf die Frage »Nur Checksummen ausgeben (j/n) ?« mit <j>, so wird die angegebene Datei durchgerechnet und die zu jeder Programmzeile sich ergebende Prüfsumme mit Zeilennummer und Programmzeile ausgegeben. Damit Sie aber nicht an den Bildschirm angegelt sind (die Rechnerei kostet doch ganz schön viel Zeit), fragt das Programm nach, ob das mit den Prüfsummen versehene Programm nur auf dem Bildschirm ausgegeben werden soll, oder ob Sie einen Ausdruck vorziehen. Ihre Wahl geben Sie durch die Beantwortung der Frage »Bildschirm oder Drucker ? (b/d)« dem Programm bekannt. Wählen Sie die Bildschirm-Ausgabe, so wird parallel zur Bildschirmausgabe im aktuellen Directory ein File mit dem Namen »Dateiname.chk« angelegt. Dies ist dann recht praktisch, falls Sie noch keinen Drucker haben, denn dann können Sie sich das Programm inklusive Prüfsummen

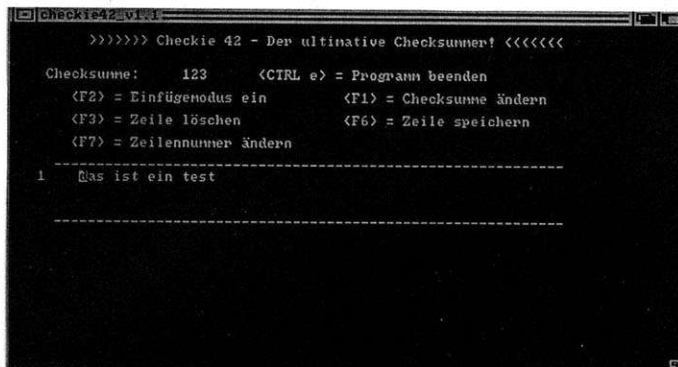


Bild 1. Wenn Sie dieses Bild auf Ihrem Monitor sehen, dann haben Sie Checkie42 in der Version 1.1 gestartet

nicht stimmt. Leider ist der Checksummer noch nicht schlaue genug, Ihnen den exakten Ort der Abweichung anzugeben.

Um Ihnen zu ersparen, wegen eines fehlenden Zeichens die ganze Zeile nochmals eingeben zu müssen (»Checkie42« unterscheidet sich hier angenehm vom CLI), hat der Checksummer-Programmierer noch eine weitere nützliche Fähigkeit in das Programm gepackt: Durch einen Druck auf <F2> aktivieren Sie den Einfügemodus. Allerdings verlangsamt dieser Insert-Mode den Programm-Ablauf beträchtlich, weshalb Sie durch einen erneuten Druck auf <F2> möglichst bald wieder in den normalen Eingabe-Modus zurückkehren sollten. War aber die ganze Zeile Schrott, löschen Sie sie am besten komplett mit <F3>.

Natürlich kann durchaus einmal der Fall eintreten, daß Sie ganz bewußt eine Programmzeile übernehmen möchten, die nicht mit dem Original überein-

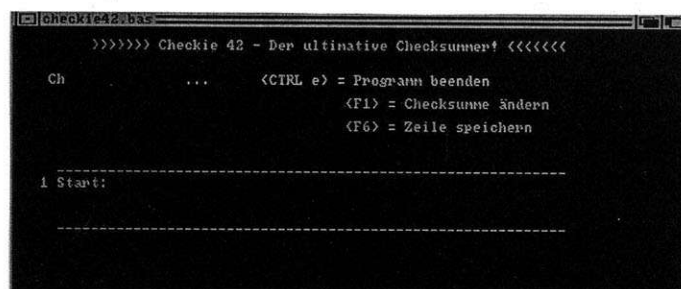


Bild 2. Sollte Ihr Checksummer dieses Aussehen haben, dann arbeiten Sie mit einem überholten Programm

mit dem Hinweis konfrontiert, daß und wie viele Zeilen mit inkorrekt Checksumme übernommen wurden.

Sind Sie mit dem Abtippen fertig oder steht Ihnen der Sinn nach einer Pause, so beenden Sie bitte die Arbeit mit »Checkie 42« durch Drücken von <CTRL e>. Das angelegte File wird danach ordnungsgemäß geschlossen und der Checksummer beendet seinen Ablauf.

Falls Sie die Abtipperie nur unterbrechen wollten und jetzt mit neuer Kraft voll Tatendrang

schlägt Ihnen als Zeilennummer für die weitere Eingabe die nächsthöhere vor. Sie geben dann das Programm weiter ein, als hätten Sie nie unterbrochen. Der vorletzte Satz ist **wörtlich** gemeint! Die **Zeilen** in der angeforderten Datei werden **gezählt** und **nicht numeriert**! Wenn Sie ein Programm mit der Zeilennummer »5« begannen und insgesamt 100 Zeilen eingaben, so hat dieses Programm in der Checksummer-Zählung nur 95 Zeilen, was dazu führt, daß Ihnen bei einem

jederzeit z.B. mit dem CLI-Befehl »type« ansehen, ohne auf die Berechnung der Prüfsummen warten zu müssen. Ist die Rechnerlei beendet, erwartet das Programm einen Tastendruck. Ist dieser erfolgt, wird das Programm beendet.

Eine weitere Eigenschaft, die Ihnen »Checkie42« zur Verfügung stellt, aktivieren Sie, wenn Sie die Frage »Nur Checksummen ausgeben?« verneinen und auf die Frage »Eingabe aus Datei? (j/n)« mit der Eingabe von <j> antworten. Dann nämlich wird die angegebene Datei geöffnet, auf dem Bildschirm die Eingabemaske aufgebaut und auf die Eingabe einer Checksumme zur ersten Programmzeile gewartet; Sie können natürlich wie gewohnt die Zeilennummer ändern. Nach Eingabe der Checksumme wird die korrespondierende Programmzeile aus der Datei geholt, in den Maskenbereich »Programmzeile« geschrieben und auf Übereinstimmung zwi-

schen Zeileninhalt und eingetragener Prüfsumme getestet. Ist alles in Ordnung, wird auf die nächste Programmzeile weitergeschaltet und die Eingabe der nächsten Checksumme erwartet. Wird eine Abweichung gefunden, setzt das Programm den Cursor auf das erste Zeichen der Programmzeile. Sie verfahren dann wie bei der Neueingabe eines Programms

Evolution

(siehe oben). Damit im Extremfall aber nicht Ihre Originale den Jordan hinunterschwimmen, legt der Checksummer zur Sicherheit eine Kopie des Original-Files unter dem alten Dateinamen, aber mit der Endung ».bak« im aktuellen Directory an; sollte Ihnen ein größerer Schnitzer unterlaufen, können Sie so noch immer auf diese Backup-Datei zurückgreifen und ein eventueller Schaden hält sich in Grenzen.

Konnten wir Sie von der Nützlichkeit unseres Checksum-

mers überzeugen? Dabei ist dessen Entwicklung sicher nicht abgeschlossen. Zwei Weiterentwicklungen sind in Planung, aber in dieser Version noch nicht eingebaut. Die erste bezieht sich auf die Reaktion des Checksummers bei Entdeckung eines Fehlers. Momentan wird der Cursor lediglich auf den Beginn der Programmzeile gesetzt und eine Korrektur erwartet. Wenn Ihnen dies zu dezent ist, sollten Sie den Programmblock mit dem Label »Wiederholung:«, der am Anfang des Programms steht, folgendermaßen ändern:

Wiederholung:

```
GOSUB EingabeZeile
IF FEnde=wahr THEN Ende
GOSUB CalcSumme
IF FZok = falsch THEN
BEEP:GOTO Wiederholung
GOSUB Uebernahme
GOTO NeueZeile
```

Wenn Sie diese Änderung einbauen, brauchen Sie nicht ständig auf den Bildschirm zu

schauen, um Fehler zu erkennen – es wird ein akustisches Warnsignal ausgegeben. Die zweite Idee, die Arbeit mit »Checkie42« etwas komfortabler zu machen, sucht noch nach einem Programmierer. Wir geben zu, die Berechnungen der Prüfsumme erfolgen nicht gerade mit allzu großem Tempo. Sollte sich daher jemand finden, der diese Rechenroutinen, die unter dem Label »CalcSumme« zu finden sind, beschleunigen kann (es muß nicht immer Basic sein) – die AMIGA-Sonderheft-Redaktion würde sich über eine Zusage sehr freuen und baldmöglichst ein Update abdrucken.

Die AMIGA-Redaktion hofft, Ihnen mit dem Checksummer »Checkie42_V1.1« ein wirkungsvolles Instrument zur Vermeidung von Tippfehlern beim Abtippen von Listings zur Verfügung gestellt zu haben. Möge Ihnen der Spaß am Abtippen nie vergehen!

so

Komprimiertes für eilige Leser

Checkie42 in Stichworten:

Tastenbelegung:

<F1> Checksumme ändern
<F2> Einfüge-Modus ein/aus
<F3> Zeile löschen
<F6> Zeile speichern (Fehler ignorieren!)
<F7> Zeilennummer ändern
<CTRL e> Programm beenden

Die Funktionen <F2>, <F3> und <F7> stehen erst ab Version 1.1 zur Verfügung.

Mit Checkie42 können Sie

- Listings mit Checksumme korrekt und kontrolliert abtippen
- Listings »häppchenweise« eingeben
- Listings, die Sie mit einem Texteditor eingegeben haben, nachträglich mit Checksummen versehen und so durch Vergleich mit dem abgedruckten Listing kontrollieren
- Listings mit Checksumme auf dem Bildschirm, dem Drucker und auf Diskette ausgeben lassen
- Listings unabhängig von der jeweiligen Programmiersprache kontrolliert eingeben
- Listings, in die sich Fehler eingeschlichen haben, einfach korrigieren
- sicher sein, in Ihrem Programm keinen Tippfehler eingebaut zu haben

Was Sie bei der Arbeit mit »Checkie42« beachten sollten:

Geben Sie die Programmzeilen, die Prüfsumme und die Zeilennummer **genau wie abgedruckt** ein. Dies erhöht die Sicherheit. Auch Groß- und Kleinschreibung, sowie Position des Befehls sind Bestandteil der Prüfsumme!

Verarbeiten Sie mit dem Checksummer **nur ASCII-Files**.

Steuerzeichen im Text können zu überraschenden Ergebnissen führen. Auch mit binär (Save-Option »b«) oder codiert (SAVE-Option »p«) gespeicherten Basic-Files weiß »Checkie42« nichts anzufangen.

Eine dreistellige Prüfsumme weist darauf hin, daß »ab jetzt« der Beginn der Programmzeile in einer anderen als der bisherigen Spalte erfolgt (wichtig bei Einrückungen!). Wenn Sie an solchen Einrückungen nicht interessiert sind und auf deren Überprüfung verzichten wollen, genügt die Eingabe der ersten beiden Zeichen der Prüfsumme.

Arbeiten Sie bitte immer mit deaktiviertem Schreibschutz, damit »Checkie42« bei Bedarf die ».bak«- und ».chk«-Dateien anlegen kann. Ein Programmabbruch wäre sonst unvermeidlich.

```
1 Cq0 REM *****
2 KF REM *** Checksummer: Checkie 42 ***
3 4c REM *** Version 1.1 ***
4 Ft REM *****
5 Z7 Start:
6 W42 GOSUB Init
7 Pn GOSUB OpenDatei
8 mb IF dn$="" THEN Ende
9 b2 GOSUB Bild
10 Ow0 NeueZeile:
11 K42 GOSUB loeschen
12 zT GOSUB EingabeSumme
13 pt IF FEnde=wahr THEN Ende
14 MVO Wiederholung:
15 sQ2 GOSUB EingabeZeile
16 sw IF FEnde=wahr THEN Ende
17 m0 GOSUB CalcSumme
18 fB IF FZok = falsch THEN GOTO Wiederholung
19 oC GOSUB Uebernahme
20 iv GOTO NeueZeile
21 mA0 Ende:
22 7S2 GOSUB fertig
23 72 END
24 UGO Init:
25 fD2 wahr=-1
26 fy falsch=0
27 Kh LZeile=240 : REM Anzahl Zeichen/Zeile
28 tz LBZeile=60 : REM Anzahl Zeichen/Bildschirmzeile
29 OG AnzBZeilen=LZeile/LBZeile
30 N1 zx=6 : zy=14 : REM Position Zeile
31 PC sx=20 : sy=5 : REM Position Checksumme
32 mF AnzCsZ=3 : REM Anzahl Ziffern/Checksumme
33 QF DIM z(LZeile)
34 O5 DIM cs(AnzCsZ*2)
35 iz cs(AnzCsZ)=0 : REM Zeilenstart
36 gv a=0 : b=0 : c=0 : REM Hilfsvariablen
37 AI i=0 : j=0 : k=0 : REM Zählvariablen
38 XL FZok = wahr : FCz3=100
39 Wn Checkfile=0 : Zeile=1
40 Bb READ Faktor(i)
41 X1 WHILE Faktor(i)<>0
42 ga4 i=i+1
43 cm READ Faktor(i) : REM Faktorenreihe
44 9x2 WEND
45 Gm AnzFak=i
46 RB DATA 2,3,4,5,6,0
47 7J RETURN
48 PWO OpenDatei:
49 BR2 ON ERROR GOTO Dateifehler
50 MS CLS
51 Vu LOCATE 3,5:dn$="":INPUT "Dateiname: ";dn$
```



```

52 Fp OPEN dn$ FOR INPUT AS #1
53 KU IF FFehler=53 THEN NeueDatei
54 4s LOCATE 5,5:PRINT "Nur Checksummen ausgeben? (j/n) ":e$=INP
UT$(1)
55 gM IF e$="j" THEN
56 Kq4 LOCATE 7,5:e$="":PRINT "Bildschirm oder Drucker? (b/d) "
:e$=INPUT$(1)
57 Ku IF e$="d" THEN
58 KM7 OPEN "prt:" FOR OUTPUT AS #4
59 YH4 ELSE
60 7Y6 OPEN dn$+".chk" FOR OUTPUT AS #4
61 vo4 END IF
62 RR WHILE NOT EOF(1)
63 yj6 LINE INPUT #1,e$
64 fg apos=LEN(e$)
65 to z(apos)=32
66 4L FOR i=1 TO apos
67 sq8 z(i-1)=ASC(MID$(e$,i,1))
68 Xn6 NEXT i
69 cE GOSUB CalcSumme
70 vp i=0
71 Rh WHILE z(1)=32 : i=i+1 : WEND
72 4v IF FCz3=1 THEN
73 f98 cs(6)=-16 : REM Leerstelle
74 nW6 ELSE
75 XA8 cs(6)=1 : FCz3=1
76 A36 END IF
77 2c PRINT USING "####";Zeile; : PRINT " ";
78 Wf PRINT #4,USING "####";Zeile; : PRINT #4," ";
79 mC FOR i=4 TO 6
80 7d8 IF cs(i)>35 THEN cs(i)=cs(i)+6
81 nS IF cs(i)>9 THEN cs(i)=cs(i)+7
82 Jy PRINT CHR$(cs(i)+48);
83 8L PRINT #4,CHR$(cs(i)+48);
84 n36 NEXT i
85 19 PRINT " ";e$
86 t1 PRINT #4," ";e$
87 Qm Zeile=Zeile+1
88 rf4 WEND
89 IF CLOSE 1 : CLOSE 4
90 H4 dn$=""
91 Fv PRINT:PRINT "Taste drücken"
92 5r e$=INPUT$(1)
93 rT RETURN
94 SL2 END IF
95 70 LOCATE 7,5:PRINT"Eingabe aus Datei? (j/n) ":e$=INPUT$(1)
96 L1 IF e$="j" THEN
97 u24 Checkfile=1
98 08 CLOSE 1
99 eX GOSUB backup
100 WZ OPEN dn$+".bak" FOR INPUT AS #1
101 Sk OPEN dn$ FOR OUTPUT AS #2
102 0c RETURN
103 bu2 END IF
104 77 WHILE NOT EOF(1)
105 eP4 LINE INPUT #1,e$
106 j5 Zeile=Zeile+1
107 Ay2 WEND
108 Y1 CLOSE 1
109 8A0 NeueDatei:
110 aM2 OPEN dn$ FOR APPEND AS #2
111 91 RETURN
112 7Q0 Dateifehler:
113 Og2 FFehler=ERR
114 PB IF FFehler=53 THEN RESUME NEXT
115 vy ON ERROR GOTO 0
116 aD0 backup:
117 yY2 OPEN dn$ FOR INPUT AS #1
118 SJ OPEN dn$+".bak" FOR OUTPUT AS #2
119 8F flen=LOF(1)
120 ih WHILE flen>=32000
121 i14 e$=INPUT$(32000,1)
122 NK PRINT #2,e$;:flen=flen-32000
123 QE2 WEND
124 e7 IF flen THEN
125 SB4 e$=INPUT$(flen,1)
126 xP PRINT #2,e$;
127 zs2 END IF
128 zg CLOSE 2 : CLOSE 1
129 R3 RETURN
130 d00 Bild:
131 f12 CLS
132 Gi LOCATE 2,10
133 yY PRINT ">>>>>>> Checkie 42 - Der ultimative Checksum
mer! <<<<<<<<<"
134 xR LOCATE 13,zx
135 Le FOR i=1 TO 2

```

```

136 kx4 FOR j=1 TO LBZeile
137 HR6 PRINT "-";
138 hy4 NEXT j
139 H1 LOCATE 13+AnzBZeilen+1,zx
140 hx2 NEXT i
141 H1 LOCATE 5,30:PRINT "<CTRL e> = Programm beenden"
142 vd LOCATE 7,8:PRINT "<F2> = Einfügemodus ein"
143 uQ LOCATE 7,40:PRINT "<F1> = Checksumme ändern"
144 1H LOCATE 9,40:PRINT "<F6> = Zeile speichern"
145 ME LOCATE 9,8:PRINT "<F3> = Zeile löschen"
146 v5 LOCATE 11,8:PRINT "<F7> = Zeilennummer ändern"
147 jL RETURN
148 5f0 loeschen:
149 w82 LOCATE zy,1
150 8S FOR i=1 TO AnzBZeilen
151 jf4 PRINT TAB(zx);SPACE$(LBZeile)
152 t92 NEXT i
153 00 FOR i=0 TO LZeile
154 V84 z(i)=32
155 wC2 NEXT i
156 3F LOCATE zy,1
157 om PRINT USING "####";Zeile
158 dy apos=0
159 vX RETURN
160 v50 EingabeSumme:
161 va2 Farbe=0 : i=1
162 AB LOCATE sy,sx+1
163 Xm PRINT "...
164 040 blinken:
165 aC2 Farbe=Farbe XOR 1
166 QH COLOR Farbe,0
167 P6 LOCATE sy,sx-15
168 1n PRINT "Checksumme:"
169 ni LOCATE sy,sx+1
170 ya e$=INKEY$
171 1T IF e$="" THEN blinken
172 Xp COLOR 1,0
173 d1 e=ASC(e$)
174 fM IF e=5 THEN
175 KU4 FEnde=wahr : REM <CTRL e>
176 hf i=AnzCsZ+1 : REM fertig
177 Cc2 ELSEIF e=135 THEN
178 dc4 GOSUB NeuZeile
179 Bz e=135
180 LB2 ELSEIF e=8 THEN
181 4h4 IF i>1 THEN
182 sZ6 i=i-1 : REM <BS>
183 XY LOCATE sy,sx+1 : PRINT "."
184 un4 END IF
185 Tn2 ELSEIF e=13 THEN
186 KZ4 IF i=AnzCsZ THEN i=AnzCsZ+1 : REM <CR>
187 cL2 ELSE
188 ra4 IF e>47 AND e<58 THEN
189 vi6 e=e-48 : REM 0-9
190 AI4 ELSEIF e>64 AND e<91 THEN
191 Rv6 e=e-55 : REM A-Z
192 jX4 ELSEIF e>96 AND e<123 THEN
193 Zx6 e=e-61 : REM a-z
194 JS4 ELSE
195 yz6 GOTO blinken : REM weder noch
196 6z4 END IF
197 JY PRINT e$;
198 ve cs(i)=e
199 D7 i=i+1
200 A32 END IF
201 wn IF i<=AnzCsZ THEN blinken
202 f50 ESEnde:
203 2K2 COLOR 1,0
204 Oh LOCATE sy,sx-15
205 c0 PRINT "Checksumme:"
206 gI RETURN
207 aE0 NeuZeile:
208 du2 IF FZok = wahr THEN
209 P44 NeuZeile=0
210 nn WHILE e<>13 OR NeuZeile=0
211 Ux6 LOCATE zy,1:PRINT USING "####";NeuZeile;
212 oF e=ASC(INPUT$(1))
213 gP IF e>47 AND e<58 THEN NeuZeile=NeuZeile*10+e-48
214 Jn IF NeuZeile > 9999 THEN e=8
215 in IF e=8 THEN NeuZeile=INT(NeuZeile/10)
216 vj4 WEND
217 kh IF Checkfile THEN
218 tq6 IF NeuZeile < Zeile THEN
219 yy8 WHILE NOT EOF(1)

```

Listing 1. Checkie42_V1.1 – ein Basic-Programm für alle Programmiersprachen


```

220 VGA LINE INPUT #1,e$
221 cd PRINT #2,e$
222 lp8 WEND
223 MH CLOSE 1 : CLOSE 2
224 FY GOSUB backup
225 Xa OPEN dn$+".bak" FOR INPUT AS #1
226 Tl OPEN dn$ FOR OUTPUT AS #2
227 9t Zeile=1
228 cv6 END IF
229 Yp WHILE (NeuZeile > Zeile) AND (NOT EOF(1))
230 fQ8 LINE INPUT #1,e$
231 mm PRINT #2,e$
232 l7 Zeile=Zeile+1
233 CO6 WEND
234 cR IF EOF(1) THEN
235 bL8 CLOSE 1
236 81 NeuZeile=Zeile
237 uN LOCATE zy,1:PRINT USING "###";NeuZeile;
238 9G Checkfile=0
239 ng6 END IF
240 oh4 END IF
241 vV Zeile=NeuZeile
242 qJ2 END IF
243 Ht0 RETURN
244 lH EingabeZeile:
245 n72 x=cs(AnzCsZ)
246 BWO weiter:
247 ga2 cy=zy+INT(x/LBZeile):cx=zx+(x MOD LBZeile)
248 vy LOCATE cy,cx
249 o6 COLOR 0,1
250 mI PRINT CHR$(z(x));
251 y1 LOCATE cy,cx
252 68 IF x>apos THEN apos=x
253 xx IF Checkfile AND FZok THEN
254 wL4 IF EOF(1) THEN
255 OR6 Checkfile=0 : CLOSE 1
256 JS4 ELSE
257 ix6 e$=INPUT$(1,1)
258 6z4 END IF
259 mV2 ELSE
260 Q24 e$=INKEY$
261 922 END IF
262 gp4 IF e$="" THEN weiter
263 OI COLOR 1,0
264 OW PRINT CHR$(z(x));
265 CF LOCATE cy,cx
266 8G e=ASC(e$)
267 UR2 IF ((e AND 127)<32) OR e=127 THEN Controlcode
268 op IF imode THEN GOSUB insert
269 I6 PRINT e$
270 K1 z(x)=e : e=30
271 3D0 Controlcode:
272 gF2 IF e=13 OR e=10 THEN
273 lN4 RETURN
274 092 ELSEIF e=30 THEN
275 2p4 a=1
276 Ea2 ELSEIF e=29 THEN
277 034 a=LBZeile
278 6G2 ELSEIF e=31 THEN
279 DD4 a=-1
280 0b2 ELSEIF e=28 THEN
281 Rg4 a=-LBZeile
282 9s2 ELSE
283 UX4 GOTO noCrs
284 WP2 END IF
285 hz x=x+a
286 5Z IF x>=0 AND x<LZeile THEN weiter
287 tD x=x-a
288 HB GOTO weiter
289 kEO noCrs:
290 dN2 IF e=8 THEN
291 ls4 IF x>0 THEN
292 Kk6 x=x-1
293 AM LOCATE zy+INT(x/LBZeile),zx+(x MOD LBZeile)
294 Bb FOR i=x TO apos
295 788 z(i)=z(i+1)
296 Ja PRINT CHR$(z(i));
297 Z0 IF i MOD LBZeile=59 THEN PRINT:PRINT TAB(zx);
298 FV6 NEXT i
299 pr z(apos)=32 : PRINT " "
300 yR apos=apos-1
301 ng4 END IF
302 H12 ELSEIF e=127 THEN
303 Kk4 FOR i=x TO apos
304 GH6 z(i)=z(i+1)
305 SJ PRINT CHR$(z(i));
306 I9 IF i MOD LBZeile=59 THEN PRINT:PRINT TAB(zx);
307 Oe4 NEXT i
308 y0 z(apos)=32 : PRINT " "

```

```

309 7a apos=apos-1
310 Vy2 ELSEIF e=129 THEN
311 oI4 GOSUB EingabeSumme
312 sC x=cs(AnzCsZ)
313 9U2 ELSEIF e=130 THEN
314 513 imode=imode XOR 1
315 tU5 LOCATE 7,28
316 yZ3 IF imode THEN
317 AT5 PRINT "aus"
318 JS3 ELSE
319 YV5 PRINT "ein"
320 6z3 END IF
321 Kg2 ELSEIF e=131 THEN
322 L54 GOSUB loeschen
323 3N x=cs(AnzCsZ)
324 7Y2 ELSEIF e=134 THEN
325 bD4 RETURN
326 PR2 ELSEIF e=5 THEN
327 ZT4 FEnde=wahr
328 eG RETURN
329 F82 END IF
330 xr GOTO weiter
331 If0 insert:
332 oJ2 IF apos>x THEN
333 3o4 FOR i=apos TO x STEP -1
334 Qt6 z(i+1)=z(i)
335 q64 NEXT i
336 PH z(x)=32
337 Ny apos=apos+1
338 Kp IF apos=LZeile THEN apos=apos-1:z(LZeile)=32
339 uK FOR i=x TO apos
340 lT6 PRINT CHR$(z(i));
341 H1 IF i MOD LBZeile=59 THEN PRINT:PRINT TAB(zx);
342 xD4 NEXT i
343 yA LOCATE zy+INT(x/LBZeile),zx+(x MOD LBZeile)
344 UN2 END IF
345 vX RETURN
346 cX0 CalcSumme:
347 3s2 a=0 : b=0 : c=0
348 z2 IF e=134 THEN
349 pv4 FZok=wahr
350 WX FF6=FF6+1
351 Gz2 ELSE
352 hv4 WHILE z(apos)=32 AND apos>0
353 pl6 apos=apos-1
354 9x4 WEND
355 N1 IF apos>0 THEN
356 O36 WHILE z(c)=32
357 Bt8 c=c+1
358 D16 WEND
359 Jc4 END IF
360 EJ FOR i=c TO apos
361 Y26 j=(i-c) MOD AnzFak
362 Qs k=(i+1-c) MOD AnzFak
363 s1 a=a+((z(i) AND 127)-32)*Faktor(j)
364 2E b=b+((z(i) AND 127)-32)*Faktor(k)
365 Ka4 NEXT i
366 pA cs(4)=a+Zeile-INT((a+Zeile)/62)*62
367 4N cs(5)=b+Zeile-INT((b+Zeile)/62)*62
368 4Q FZok=(cs(1)=cs(4)) AND (cs(2)=cs(5))
369 tm2 END IF
370 Kw RETURN
371 pM0 Uebernahme:
372 xD2 FOR i=0 TO apos
373 lh4 PRINT #2,CHR$(z(i));
374 Tj2 NEXT i
375 VZ4 PRINT #2,""
376 5R2 Zeile=Zeile+1
377 R3 RETURN
378 N90 fertig:
379 MJ2 IF Checkfile THEN
380 ZZ4 WHILE NOT EOF(1)
381 6r6 LINE INPUT #1,e$
382 DE PRINT #2,e$
383 cQ4 WEND
384 Ok CLOSE 1
385 922 END IF
386 5q CLOSE 2
387 nt CLS
388 Rg LOCATE 12,35
389 WL PRINT "F E R T I G !!!"
390 I9 LOCATE 20,1
391 D1 IF FF6<>0 THEN
392 C14 PRINT "ACHTUNG!!! ";
393 9H PRINT FF6;" Zeile(n) wurde(n) ungeprüft gespeichert."
394 IB2 END IF
395 J1 RETURN
396 M&T (C) 1988 M&T

```

Listing 1. (Schluß)

AMIGA

Markt & Technik

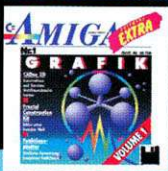
SOFTWARE
EXTRA

Software zum Taschengeldpreis

Jedes Paket nur DM 49,-*

(sFr 45,-
/öS 490,-*)

Grafik



Amiga Extra Nr. 1: Grafik I

Drei Programme, die die außergewöhnlichen Grafikfähigkeiten des Amiga nutzen!
Bestell-Nr. 38708



Amiga Extra Nr. 4: Grafik II

Spielerisch Bobs erzeugen. IFF-Bilder werden zu Bobs und Images. Generieren von animierten Icons.
Booster: par excellence.
Bestell-Nr. 38725

Spiele



Amiga Extra Nr. 12: Spiele

Highway 42: Als Kurier auf dem Planeten Cervezia. Warlords: Ein Brettspiel für taktisch geschickte Spieler.
Zargon: Joystick-Action
Bestell-Nr. 38769



Amiga Extra Nr. 5: Spiele

Breaking out: Actionspiel mit toller Grafik und Sound.
Decoder: Verwandeln Sie Ihren Amiga in eine Morsestation.
Megamind:
Bestell-Nr. 38752



Amiga Extra Nr. 3: Spiele

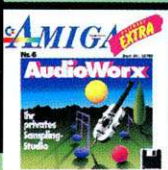
Bliff: Eine ausgeklügelte Variante des Billards.
Quadriga: Ein Spiel für Denker, angelehnt an »Vier gewinnt«.
Wikinger I: ein Strategiespiel.
Bestell-Nr. 38724



Amiga Extra Nr. 13: Regnum

Regnum ist Ihr Königreich in einer imaginären Welt. Die Amiga-Maus dient Ihnen als Zepter. Ihre Aufgabe: Sichern und vergrößern Sie Ihren Besitz.
Bestell-Nr. 38781

Musik



Amiga Extra Nr. 6: Audio Wox

Ihr privates Sampling-Studio.
Bestell-Nr. 38748



Amiga Extra Nr. 9: Sonix-Hitkiste

Tolle, in Sonix editierbare und digitalisierte Geräusche und Effekte für eigene Musikstücke.
Bestell-Nr. 38753



Amiga Extra Nr. 2: Disk Utilities I

Disk-Ed V6, Select Copy, D Copy II, Check, Bootgiri Plus. Mit einem Super-Diskeditor.
Bestell-Nr. 38726



Amiga Extra Nr. 10: Disk Utilities II

DIMO: Diskettenmonitor. Recover II: Datenrettung.
TUC - The Ultimate Cruncher: Dateien extrem komprimieren.
Bestell-Nr. 38766



Amiga Extra Nr. 14: MenuMind 1.0

MenuMind bietet Ihnen eine menügesteuerte, grafische Programmierung, die den CLI völlig ersetzt.
Bestell-Nr. 38771



Amiga Extra Nr. 11: Karteikasten

Finden Sie sich in Ihrem Diskettenbestand nicht mehr zurecht? Oder, oder, oder... Mit Dateiverwaltungen werden Ihre Probleme gelöst.
Bestell-Nr. 38768

Anwendungen und Utilities

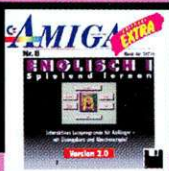
Erscheint
im
November '89

Spielend lernen!



Amiga Extra Nr. 7: Erdkunde I

Die Reihe »Spielend lernen« verknüpft das Begeisterte des Amiga mit dem Nützlichen. Stupide Paukerei wird durch Kurzweil ersetzt
Bestell-Nr. 38774



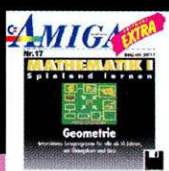
Amiga Extra Nr. 8: Englisch I

»Englisch I« vermittelt Ihnen Grundkenntnisse der englischen Sprache.
Bestell-Nr. 38775



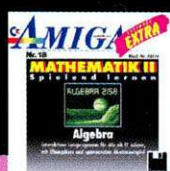
Amiga Extra Nr. 16: Erdkunde II

Spielend lernen! Vereinigte Staaten von Amerika. Interaktives Lernprogramm für alle ab 12 Jahren, mit Übungskurs und Quiz.
Bestell-Nr. 38776



Amiga Extra Nr. 17: Mathematik I (Geometrie)

»Mathematik I« vermittelt Ihnen die Grundlagen der Geometrie bis hin zur sphärischen Trigonometrie.
Bestell-Nr. 38777



Amiga Extra Nr. 18: Mathematik II (Algebra)

vermittelt Ihnen die Grundlagen der Algebra, die die Voraussetzung für die gesamte weiterführende Mathematik sind.
Bestell-Nr. 38778



Amiga Extra Nr. 19: Physik I

Grundlagen der Mechanik, der Wärmelehre und der Optik. Animationen und Soundeffekte erhöhen die Verständlichkeit und die Lerneffizienz.
Bestell-Nr. 38779

INFO-COUPON

Bitte senden Sie mir Ihr Gesamtverzeichnis mit über 500 aktuellen Computerbüchern und Software

Name

Straße

PLZ/Ort

Bitte ausschneiden und schicken an: Markt & Technik Verlag AG, Buch- und Software-Verlag, Hans-Pinsel-Straße 2, 8013 Haar bei München

Buch- und
Software-Verlag
Markt & Technik
Hans-Pinsel-Straße 2
8013 Haar bei München
AMT SH 8

*Unverbindliche Preisempfehlung

Markt & Technik-Bücher und -Software erhalten Sie bei Ihrem Buchhändler, in Computer-Fachgeschäften und in den Fachabteilungen der Warenhäuser

clevere Amiga-User verwenden gleich *ProMigos*

denn wir verwenden nur ausgesuchte Qualitätsware für unsere Produkte, die auch höchsten Ansprüchen gerecht werden. Und das bei einem sehr guten Preis-Leistungs-Verhältnis!!

ProMigos Zubehör rund um den Amiga:

elektronischer Bootselektor

wahlweise booten von DF0: – DF2; (optional auch DF3:)

nur 58,00 DM

ROM/ROM Kickstart-Umschaltplatine

Zwischen zwei original ROMs (40polig) umschalten

nur 49,00 DM

ProMigos 3,5"-Floppystation

abschaltbar, durchgeführter Bus, extrem leise,
NEC-Drive

nur 279,00 DM

ProMigos Boot-Strap-Platine

beliebige Kickstart wie beim A1000 aus RAM booten
(Lieferung ohne RAM & A1000-BootROMs)

nur 98,00 DM

ProMigos Autoboot-Harddisk- Station für A500 & A1000

Kapazität von 20 bis 110 MByte,
ALF2-Treibersystem inkl. Backup-Software

ab 1168,00 DM

ProMigos 512 KByte RAMCARD

mit eingebauter Uhr, abschaltbar, für A500

nur 319,00 DM

ProMigos Kickstart-Umschaltplatine

zwischen einem ROM & zweimal Eprom umschalten

nur 68,00 DM

ProMigos 5,25"-Floppystation

40/80 Track, abschaltbar, durchgeführter Bus,
TEAC-Drive

nur 349,00 DM

Autoboot-Upgrade-Kit

Autoboot-Upgrade-Kit für vorhandene *ProMigos* &
Amigos Harddiskstationen inkl. ALF2-Treiber und
Backup-Software

nur 248,00 DM

ProMigos Autoboot-FileCard für A2000

Kapazität von 20 bis 50 MByte (intern), ALF2-
Treibersystem inkl. Backup-Software

ab 1120,00 DM

- bitte ausführliche Info anfordern
- *ProMigos*-Produkte erhalten Sie
direkt bei uns
oder im guten Fachhandel
- deutsche Qualitätsfertigung
- Ausländische Distributoren gesucht
- Händleranfragen erwünscht!!

Amiga und Commodore sind eingetragene Warenzeichen der Firma Commodore Inc.

Flesch & Hörnemann Computer OHG
Schlägel & Eisen Straße 46
4352 Herten
Telefon: 049-0 23 66-5 51 76
Telefax: 049-0 23 66-5 34 50