

AMIGA DOS

Erstausgabe

1/90

ISSN 0937-2717
DMV-Verlag

Einfach abtippen:
Profi-Textverarbeitung
CONTEXT AMIGA

AMIGA'89 KÖLN

– Die Knüller auf einen Blick

SENSATIONELL

– Amiga-BASIC auf PC
und C-64 übertragen

WERKSTATT

– dPaint richtig nutzen
– AmigaBASIC für Einsteiger

JEDE MENGE TIPS & TRICKS

– IFF Handling
– Viruswürger
– Soundeditor
– und, und, und...



MIT
GROSSEM
**SPIELE-
TEIL**

GFA für AMIGA

GFA-BASIC 3.0 Interpreter Amiga

Strukturiertes Programmieren, über 300 Befehle und Funktionen, zusätzlich alle Betriebssystemfunktionen, extrem hohe Geschwindigkeit.

DM 198,—

neu

GFA-BASIC 3.0 Compiler Amiga

Mit dem integrativen Compiler werden Ihre GFA-BASIC-Programme noch schneller.

Viele Optionen und Linker (kompatibel zu A-Link und B-Link) für andere Programmiersprachen im Lieferumfang enthalten.

DM 99,—

neu



Der Einstieg in GFA-BASIC 3.0 Amiga

Ein Lehrbuch für Programmieranfänger.

Dietmar Schell vermittelt auch dem unerfahrenen Programmierer Ideen und Anwendungsbeispiele für das Programmieren in GFA-BASIC. 248 Seiten, Hardcover, ISBN 3-89317-009-X

DM 29,—

neu



Training für Fortgeschrittene GFA-BASIC 3.0

Wer schon Erfahrung auf dem Amiga oder in irgendeinem BASIC-Dialekt hat, wird von den beiden Autoren bestens betreut.

Man erfährt und lernt eine Menge über Programmiertricks, nützliche und verwendbare Prozeduren,

Anwendungen und die Besonderheiten des GFA-BASIC für Amiga.

329 Seiten, Hardcover, inkl. Diskette, ISBN 3-89317-010-3

DM 49,—

neu

GFA-ASSEMBLER Amiga

Professioneller Makro-Assembler für 68000-Programmierer:

Leistungsfähiger Editor mit integriertem Assembler und Linker.

Nachladbarer Debugger.

Jetzt auch für die Commodore-Amiga-Computer lieferbar.

DM 149,—

neu

ZOETROPE

Das Computer-Animationssystem für Ihren Amiga mit der Funktionalität und den Eigenschaften, die man nur bei erstklassigen Grafiksystemen findet.

Das professionelle 2D-Animationsprogramm von ANTIC-Software, exklusiv von GFA. Umfangreiches Handbuch und Programm in Deutsch.

DM 198,—

Amiga genügend
0211/5504-0

GFA Systemtechnik GmbH
Heerdter Sandberg 30-32
D-4000 Düsseldorf 11
Tel. 0211/5504-0 · Fax 0211/550444

GFA
SYSTEMTECHNIK

Liebe Leser, Sie halten gerade die Erstausgabe einer neuen Amiga-Zeitschrift in Ihren Händen. Noch ein Magazin mit der Zielrichtung Amiga, werden Sie denken. Aber daß AMIGA DOS mehr ist, werden Sie feststellen, wenn Sie diese Zeitung zu Ende gelesen haben. Wir, das AMIGA-DOS-Team, haben uns zum Ziel gesetzt, eine Zeitung zusammenzustellen, in der Sie alles Wissenswerte zum Thema Amiga finden. Und betrachtet man den Softwaremarkt für diesen Computer, so entdeckt man, daß Unterhaltung ein zentrales Thema im Zusammenhang mit dem Amiga darstellt. Wir haben diesem Umstand Rechnung getragen und die interessantesten Spiele für Sie getestet, um Sie Ihnen vorzustellen. Doch auch diejenigen, die es vorziehen auf dem Amiga zu programmieren, finden genug Programme und Kurse, um die eigenen Kenntnisse auszubauen und zu vertiefen. So manches Programm, das Sie hier finden, wird Ihnen in Zukunft wertvolle Dienste leisten. Besonders möchte ich in diesem Zusammenhang auf unsere Toplistings Doublestrike und **CONTEXT-AMIGA** hinweisen. Ein Shoot'em Up und eine Textverarbeitung – wählen Sie selbst! Der Amiga hat aber noch wesentlich mehr zu bieten. Public Domain, Hard- und Software, Tips & Tricks oder Mes- seberichte aus aller Welt sind weitere zentrale Themen der AMIGA DOS. Denn AMIGA DOS hat sich zum Ziel gesetzt, umfassend zu informieren. Deshalb ist AMIGA DOS das Magazin für Sie.

Uns interessiert natürlich Ihre Meinung. Machen Sie mit bei unserer großen Umfrage, und gewinnen Sie mit etwas Glück einen der attraktiven Preise. Denn Ihre Meinung ist für uns wichtig. Schicken Sie uns Ihre Anregungen und Ihre Kritik, damit AMIGA DOS immer Ihren Wünschen entsprechen kann. In diesem Sinne wünschen wir Ihnen ein frohes Fest, einen guten Rutsch ins neue Jahr und viel Spaß bei der Lektüre der neuen AMIGA DOS...



NEWS & TRENDS

- Amiga News** 6
- Die Amiga 89 in Köln** 10
 - Was gibt es Neues für den aktiven Amiga-Anwender?
- Rund um den Amiga World of Commodore** 14
 - Neues aus den USA

SOFTWARE-TEST

- Ein Grafikgigant** 20
 - Digi Paint 3
- Animation spielend einfach** 21
 - Zoetrope
- Dem BASIC Beine gemacht** 24
 - HisoftBASIC und Compiler
- BASIC komfortabel** 25
 - GFA-BASIC 3.014 und Compilertest

HARDWARE-TEST

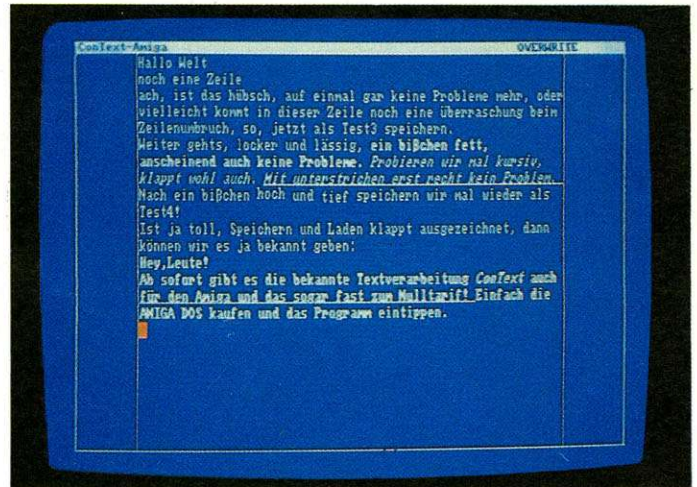
- Das Unsichtbare sichtbar machen** 28
 - Pro Access V 2.17
- Marktübersicht Zweitlaufwerke** 32
 - 3,5-Zoll- und 5,25-Zoll-Zweitlaufwerke im Überblick

TIPS & TRICKS

- Programmierhilfe**
 - Eingabehilfe HexType 36
 - Eingabehilfe Checksummer 37
- Stichwort BASIC und ASSEMBLER** 94
 - IFF-Handling
- Basic in allen Gassen** 100
 - Portieren von AmigaBASIC-Programmen
- Die Jagd geht weiter** 106
 - Viruswürger
- Übersicht beim Programmieren** 108
 - Bibliotheksroutinen im Überblick
- Gewußt wie!** 112
 - Short Tips

KURS

- Programmieren mit Erfolg** 46
 - AmigaBASIC
- Aufsteigerwissen Assembler** 52
 - ASSEMBLER für Profis



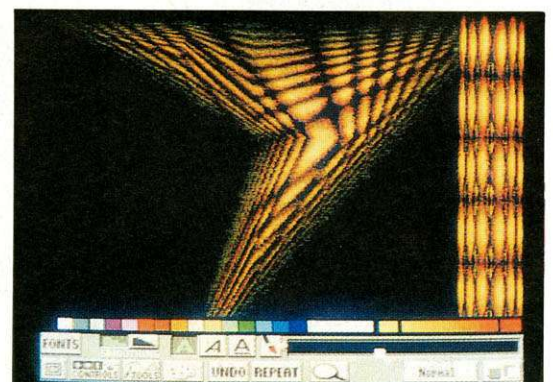
Eine Textverarbeitung zum Abtippen mit WYSIWYG – ConText Amiga. Überzeugen Sie sich selbst von der enormen Leistungsfähigkeit dieses Programms

Seite 84



Weihnachten steht vor der Tür – ein idealer Zeitpunkt, um Peripheriegeräte für den Amiga anzuschaffen. Eine Marktübersicht informiert über Zweitlaufwerke

Seite 32



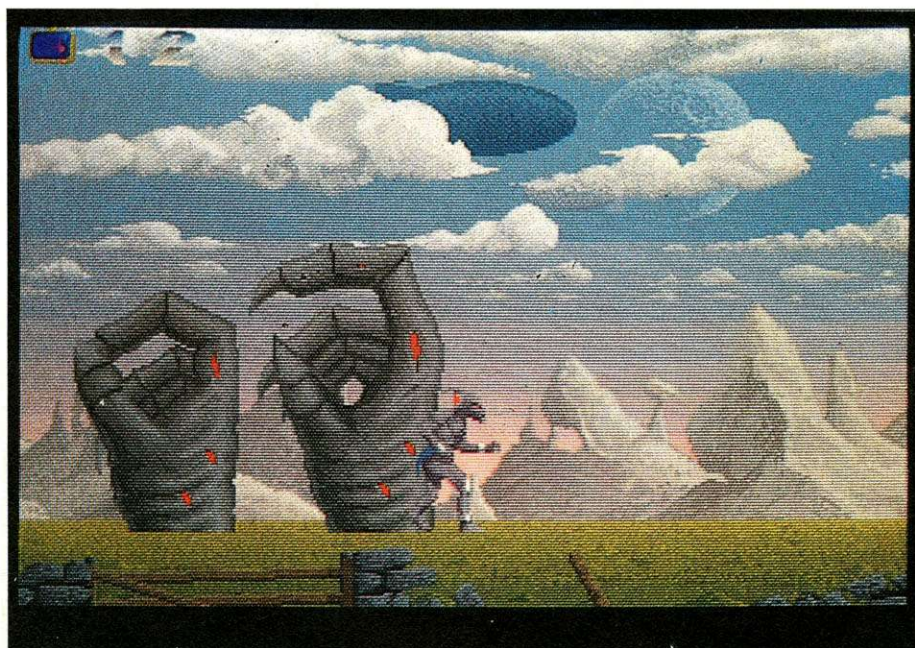
Digi Paint 3 – der Grafikgigant mit 4096 Farben. Was dieses Programm leisten kann, erfahren Sie auf

Seite 20



World of Commodore in New York: Der Amiga liegt voll im Trend. Was es Neues für Ihren Computer gibt, lesen Sie auf

Seite 14



Shadow of the Beast setzt grafisch neue Maßstäbe – einen genauen Testbericht finden Sie auf

Seite 128

LISTING

Bis der Joystick glüht	38
– DoubleStrike	
ASSEMBLER macht's möglich	60
– Soundeditor in Assembler	
Super-Textverarbeitung mit WYSIWYG	84
– ConText Amiga	

WERKSTATT

Modula2	68
– Maschinencode in M2 Amiga	
C	72
– C's Vorarbeiter	
Arbeiten wie die Profis	76
– mit dPaint III	

PUBLIC DOMAIN

Public-Domain-Werkzeugkiste	116
Software zum fairen Preis	120
– Public-Domain-Spiele-Show	

SPIELE

Spielerests	
– Safari Guns	122
– Gilbert	124
– Laser Squad	124
– Rainbow Warrior	125
– Rock'n'Roll	126
– Stadt der Löwen	126
– Shadow of the Beast	128
– Mindbender	130
– Red Lightning	130
– Mr. Heli	135
– Swords of Twilight	136
– Twinworld	137
– Bloodwych	138
– Xenon II, The Megablast	140
– Wayne Gretzky Hockey	140
– Indiana Jones and the last Crusade	141
– Space Quest III	143
Helpline	131
– AMIGA-DOS-Spieletips	
Previews	142
– Demnächst auf Ihrem Computer	

RUBRIKEN

Editorial	3
Wettbewerb	26
Tüftelecke	64
Leserbefragung	65
Bücher	82
Comic	83
Impressum	144
Inserentenverzeichnis	144
Vorschau	146

Die Amiga-Tower-Alternative

Der Amiga 2000 ist ja ziemlich ausbaufähig, nur wie sieht es aus, wenn man eine XT- oder AT-Karte, ein Zweitlaufwerk und eine Filecard besitzt? Da kann es schon zu Platzproblemen kommen. Die Alternative hieße natürlich, sich ein Towergehäuse für knapp 700,- DM zuzulegen, wenn man den Peripheriewust der externen Geräte umgehen will. Doch es gibt noch eine andere Methode: ein Adaptergehäuse, das von der Firma Comp.Z. entwickelt wurde.

Das Adaptergehäuse kann ein 5,25-Zoll- und vier 3,5-Zoll-Laufwerke oder zwei 5,25-Zoll- und zwei 3,5-Zoll-Laufwerke aufnehmen. Ferner können ein Netzteil und ein Lüfter (insofern Festplatten eingebaut sind) betrieben werden. Kontrolleuchten, die an der Vorderfront ange-

bracht sind, werden an die im rückwärtigen Teil liegenden Festplatten angeschlossen, so daß Zugriffe darauf auch erkannt werden. Die Festplatten- und Laufwerkskabel werden durch eine Gehäuseausparung im Adaptergehäuse in den Amiga geführt. Das Adaptergehäuse wird mit dem Amigagehäuse durch seitlich liegende Schrauben verbunden. Der Amiga wird dadurch nur 5,5 cm höher. Der Preis für das Gehäuse liegt bei zirka 265 DM, für das Netzteil, das das interne Amiganetzteil entlasten soll, zirka 145 DM, und der Lüfter soll etwa 35 DM kosten.

Bezugsquelle:
Comp.Z.
Pochgasse 31
7800 Freiburg
Tel.: 0761/554280

Ist Ihr Amiga zu schnell?

Haben Sie das Problem, daß Sie zum x-ten Mal bei einem Ballerspiel von den feindlichen Aliens abgeschossen wurden, weil die Gegner viel schneller waren als Sie? Sicherlich, das kann den schönsten Spielspaß vermiesen.

Doch das muß nicht sein, denn die Firma HK-Computer hat eine Prozessorbremse herausgebracht, die das schnellste Actionspiel zur langsamsten Schneckenrallye macht und so auch dem weniger versierten Ballerkünstler einen Eintrag in den Highscore ermöglicht.

Die Amiga-Bremse wird zwischen dem MC68000-Sockel und dem Hauptprozessor platziert. Sie enthält ein Potentiometer und einen Umschalter, mit dem man den momentan laufenden Prozeß unterbrechen kann, bzw. die Ge-

schwindigkeit mit dem Potentiometer stufenlos regeln kann. Das Umschalten ist jederzeit absturzsicher möglich. Steht der Umschalter in Mittelstellung, so arbeitet der Amiga völlig normal.

Einziger Nachteil: Ein eventuell noch bestehender Garantiespruch auf Ihren Amiga erlischt natürlich, da zum Einbau der Prozessorbremse der Amiga geöffnet und der Hauptprozessor aus seinem Sockel gehoben werden muß.

Die Amiga-Bremse ist für zirka 40 DM unter folgender Adresse erhältlich:

Bezugsquelle:
HK-Computer F.Hansmann
& T. Küpper GBR
Bonner Str. 37
5000 Köln 1
Tel.: 0221/311606

Neue Software aus dem Hause Roßmüller

Auf der Amiga-Messe in Köln wurden die neuen Produkte der Firma Roßmüller Handshake GmbH vorgestellt. Hier eine kleine Übersicht:

Lichte Zeiten für den Amiga

Ein Eingabemedium, das besonders bei Layout-, CAD-, Grafik- und Zeichenprogrammen Anklang findet, macht den Anfang: der Lightpen für den Amiga. Dieser wurde konzipiert, um den Mauszeiger schneller und bequemer als mit der Maus auf dem Bildschirm zu positionieren. Ein Mausersatz ist der Lichtgriffel jedoch nicht, da zum Zeichnen und Aktivieren von Menüpunkten die Maustasten benötigt werden. Angeschlossen wird er am Joystickport 2. Damit auch Amiga-1000-Besitzer den Lightpen betreiben können, benötigt man einen Lightpen/Joystick-Adapter, der bei untenstehender Bezugsquelle für knapp 30,- DM erhältlich ist. Der Preis für den Lightpen inklusive der Treibersoftware liegt bei zirka 80 DM.

Foto-Film-Entwicklung Marke "Homegrown"

Bei diesem Produkt handelt es sich um einen computergesteuerten Roboterarm, der Fotofilme (beispielsweise Ihre Urlaubsfotos) entwickelt. Die

Steuerung des Roboters übernimmt auf der Amiga '89 ein Amiga 500. Fünf elektronisch temperaturgeregelte Bäder gehören dem Paket genauso an wie die Treibersoftware und der Roboterarm. Kosten soll das vollautomatische Fotofilm-Entwicklungslabor knapp 700,- DM.

Tuning für die XT-Karte

Was die Geschwindigkeit der XT-Karte des Amiga angeht, dürfen im Gegensatz zu den PCs einige Abstriche getan werden. Abhilfe soll hier das Produkt Turbo-XT schaffen. Es handelt sich dabei um eine Platine, die auf der XT-Karte angebracht werden muß; es wird also kein zusätzlicher Slot belegt. Für diejenigen, die in ihrem Rechner ungern herumlöten wollen, besteht die Möglichkeit, die Karte bei der Firma Roßmüller einzusenden, auf daß diese Arbeit erfahrenen Hardwarebastlern übergeben wird. Mit diesem Zusatz wird Ihrem "PC im Amiga" eingeheizt: Satte 8 MHz werden in Zukunft Ihre Daten umherwirbeln, wenn Sie sich Turbo-XT zulegen. Wem das Ganze zu schnell ist, der kann natürlich auf die alte Taktfrequenz umschalten (besonders bei schnellen PC-Spielen zu empfehlen). Der



Produktfoto: Lightpen Amiga

Preis für die Platine beträgt zirka 200 DM.

Amiga: Speicher satt

Wer mit "seiner Freundin" professionell arbeiten will, der benötigt eine Speichererweiterung. Hier bietet Roßmüller für alle Amiga-Modelle etwas an:

Den Anfang macht eine 8 MByte RAM-Karte für den Amiga 1000, die mit 2 MByte bestückt ist. Schlappe 798,- DM kostet diese Erweiterung, die übrigens auch in den Atari ST und den Mac eingesetzt werden kann.

Weiter geht's mit Erweiterungen für den 500, als da wären: Eine 512 kByte RAM-Karte mit Akku für 179,- DM, oder ist Ihnen das zu wenig? Eine

interne 2 MByte RAM-Karte, ebenfalls mit Akku, für 649,- DM sollte auch "Speicherfressern" genügen. Wer wissen will, was die Stunde geschlagen hat, sollte sich das Uhren-IC für den Amiga 500 zulegen. Der Preis liegt bei 19,- DM.

Auch der 2000er kommt bei Roßmüller nicht zu kurz. Eine autokonfigurierende und abschaltbare 8 MByte RAM-Karte, die mit 2 MByte bestückt ist, dürfte für den Anfang wohl reichen. Der Preis: 798,- DM.

Bezugsquelle:
Roßmüller Handshake GmbH
Neuer Markt 21
5309 Meckenheim
Tel.: 02225/2061

Festplatten

28 ms, OMTI - Controller, komplett für AMIGA 500/1000, **AutoPark**, durchgeführter Expansionsport, Boil! - Treiber, Einbaumöglichkeit einer 2. Platte, sehr schnell, **1 Jahr Garantie**, formschönes Gehäuse (als Monitoruntersatz verwendbar, LxBxH : ca. 32x32x6 cm)

Bei Drucklegung standen die genauen technischen Daten noch nicht fest :

AutoBoot und ein noch **schnelleres BOIL!** (über 500 KByte/S) in Vorbereitung.

Bitte fordern Sie unverbindlich und kostenlos unsere aktuelle Produkt- und Preisübersicht an.

42 MB 28 ms, 400 KB/S **1248.-**

Für AMIGA 2000, intern, ermäßigen sich unsere Preise um DM 150.-

65 MB 28 ms, 330 KB/S **1398.-** **122 MB** 28 ms, 330 KB/S **2098.-**

42 MB 24 ms, 400 KB/S, 3.5", sehr leise **1298.-**

Testauszug AMIGA (Ausgabe 9/89, Seite 151, Herausgeber Markt & Technik) :

"Als Treibersoftware kommt BOIL (Bootable Omti Interface Loader) zum Einsatz. **BOIL ist einer der schnellsten Festplattentreiber, die für den AMIGA verfügbar sind.** Bei Verwendung einer Seagate ST 251 mit einem OMTI 5520 erreichte die AMIGOS Festplatte eine Schreib/Lese Geschwindigkeit von 380/220 KByte/s. BOIL arbeitet mit allen Festplatten zusammen, die einen OMTI - Controller verwenden. Ab Kickstart 1.3 ist die Festplatte FastFileSystem bootfähig. Auf der mitgelieferten Systemdiskette befinden sich Hilfsprogramme zum Formatieren, ein Festplattentest und ein Programm, daß das Interface auf Funktionstüchtigkeit untersucht."

Testauszug Kickstart (Ausgabe 10/89, Seite 19, Herausgeber Heim Verlag) :

"Die mit dem BOIL - Treiber installierten Platten sind rebootable unter Kickstart 1.3, d.h. daß nur einmal von Diskette gebootet werden muß. Nach einem Reset wird dann automatisch von Festplatte gebootet, wobei auch die Bootpartition unter FFS gewählt werden kann. Die Arbeitsgeräusche der Platten sind gering, auch der leise Lüfter trägt dazu bei. **Insgesamt machen die Festplatten von Frank Strauß Elektronik einen sehr guten Eindruck, was nicht nur an dem hervorragenden BOIL Treiber liegt, sondern am ganzen Konzept;**"

Diskettenlaufwerke

NEC

TEAC

Vollkompatibel, anschlußfertig, abschaltbar, inkl. Kabel, amigafarbenem Metallgehäuse, 2x80 Spuren, alle Laufwerke mit beiger Frontblende. Wir verwenden nur Markenlaufwerke von **TEAC** (FD 235 F oder FD 55 (G)FR) und **NEC** (1037a). Alle 5.25" Stationen werden mit 40/80 Trackumschaltung, durchgeführtem Bus und original Commodore - Treiberplatine geliefert. Auf alle TEAC Diskettenlaufwerke geben wir **1 Jahr Garantie**. Durchgeführter Bus (3.5") : DM +10.-

3.5": 229.-

5.25": 259.-

Festplattentreiber

BOIL! = 400 KB/S
Bootable OMTI Interface Loader

DM 75.-

Speicher

512 KB Erweiterung für A 500, Uhr 279.-

Drucker

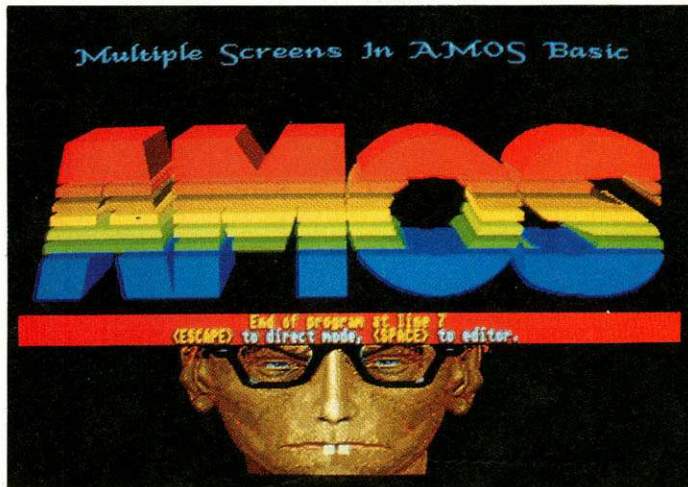
Wir sind autorisiertes STAR Systemhaus
Star LC 24-10 (deutsch) 698.-

Frank Strauß Elektronik
Schmiedstraße 11

6750 Kaiserslautern
Tel.: (0631) 67096 - 98

Fax: 60697

Lieferung erfolgt ab Lager mit
UPS oder Post per Nachnahme.



Bildvorlage: AMOS

AMOS – Das BASIC für Spieleprogrammierung

Der Amiga ist bekannt für seine Grafik- und Soundmöglichkeiten. Leider ist es für den Amiga-Neuling jedoch nicht ganz einfach, diese Möglichkeiten auszunutzen. AMOS (auf dem Atari ST unter dem Namen STOS entwickelt) soll hier Abhilfe schaffen. Es handelt sich hierbei um einen BASIC-Dialekt,

der speziell für die Spieleprogrammierung konzipiert wurde. Das Softwarepaket enthält unter anderem einen Sprite-, Musik- und Copper-List-Editor, sowie drei komplett mit AMOS erstellte Spiele. Laut Hersteller Mandarin sind ein Compiler und eine 3-D-Version von AMOS in Planung.

Amiga 500 New Art – Lifestyle-Gewand für den Amiga

In Zukunft soll Commodore bunt aufgepeppten Amigas auf den Markt bringen. New Art heißt die Sonderkollektion, die Farbe in das Alltagsleben der Amigabesitzer bringen soll. Designed wurde die limitierte Auflage der Modern-Art-Amigas (von jedem Muster nur 10000 Exemplare) übrigens von der Fernsehmoderatorin Stefanie Tücking, bekannt durch verschiedene

Musiksendungen aus dem Fernsehen.

Flippig-verrückt mit Leopardendmustern und bunten Schnipseln zeigt sich die eine Version, kühl, sachlich und anspruchsvoll die andere, mit wenig farbigen Akzenten auf weitmaschigem Rechenkaro.

Der Aufpreis für die poppigen Amigas liegt etwa bei 100,- DM gegenüber den "normalen".



Produktfoto: Amiga 500 New Art



Produktfoto: reis-ware-Maus

Amiga-Maus mit Mikroschaltern

Sind die Tasten Ihrer Amiga-Maus ausgeleiert? Machen Sie nicht mehr das, was Sie von ihnen verlangen? Dann wird es wohl oder übel Zeit, sich nach einem neuen Mäuschen umzusehen. Die Ursache des "Ausleierns" liegt zumeist darin, daß einige Mäuse mit Metallzungen ausgeliefert wurden, auf die ein spitzer Plastikstift gedrückt wird. Man kann sich leicht vorstellen, daß die Lebenserwartung dieser Plastikdorne nicht gerade hoch ist. Die Firma reis-ware hat eine Amiga-Maus

mit Mikroschaltern auf den Markt gebracht, die eine Alternative zu der Commodore-Maus darstellt. Auch die Form präsentiert sich als äußerst ergonomisch. Das eckige Design ist "wohlgeformten Proportionen" gewichen. Der Hersteller gewährt auf das für knapp 90,- DM erhältliche Produkt 12 Monate Garantie.

Bezugsquelle:

reis-ware
Postfach 36
5584 Bullay
Tel.: 06542/2086-87

Vom Amiga 2000 zur DTP-Workstation

Der Amiga 2500 als Basisgerät des Commodore-DTP-Paketes ist schon seit Jahresmitte auf dem Markt. Amiga-2000-Besitzer können ihre Geräte laut Hersteller mit einer entsprechenden Zusatzkarte zum leistungsstärkeren Amiga 2500 aufrüsten. Was diese Karte allerdings kosten soll, darüber schweigen sich die Damen und Herren bei Commodore bislang aus.

Der Amiga 2500 arbeitet mit dem Motorola 68020 mit einer Taktfrequenz von 14,2 MHz und ist softwaremäßig auf den Motorola-Prozessor 68000

umschaltbar. Die drei Prozessoren für DMA, Video, Grafik und Ton, Fat-Agnus, Denise und Paula, sowie der mathematische Coprozessor 68881 entlasten den Hauptprozessor. Für die Speicherverwaltung steht der Motorola 68851 PMMU zur Verfügung. Er unterstützt die Betriebssysteme Amiga-DOS Version 1.3, Kickstart 1.3 (autoboot) und Unix V Version. Standardmäßig stehen dem Amiga 2500 3 MB Arbeitsspeicher (davon 1 MB Chip-Memory) und eine Festplatte von 40 MB zur Verfügung.

Maus und Joystick an einem Port

Es kommt häufig vor, daß bei Spielen von der Workbench mit der Maus gestartet wird, beim eigentlichen Spielen jedoch der Joystick im Mausport benötigt wird. Da heißt es dann Maus raus und Joystick rein! Um dieses Problem

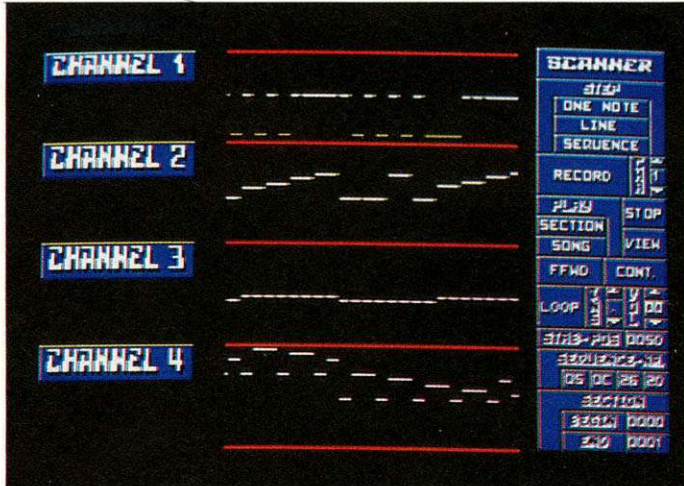
zu umgehen, hat die Firma HK-Computer einen Adapter entworfen, mit dem man die Maus und den Joystick an einem Port benutzen kann. Besonders beim Amiga 2000 entfällt somit das umständliche Wechseln der jeweiligen

Stecker. Mittels Kippschalter kann der Benutzer nun zwischen Maus- und Joystick-Steuerung auswählen.

Bei dem Adapter zeigen zwei LED's das jeweils betriebene Gerät an. Der Preis des Maus-

/Joystick-Adapters liegt bei DM 44,50.

Bezugsquelle:
HK-Computer F.Hansmann
& T. Küpper GBR
Bonner Str. 37
5000 Köln 1
Tel.: 0221/311606



Screenshot: Mark II Sound System

Fetzig Klänge aus dem Amiga – Mark II Sound System

Von der Firma Cachet stammt ein neuer Sound-Editor namens Mark II Sound System. Mit dem nicht kopiergeschützten Sound-Editor ist es möglich, 52 verschiedene Instrumente (IFF- und RAW-Format) gleichzeitig einzuladen. Der Mark II ist auf Festplatten installierbar und unterstützt (man höre und staune!) auch Fast-RAM. 255 voneinander unabhängige programmierbare Takte machen einen wahren Musik-künstler aus dem Amiga. Weitere Features dieses Programmes:

- Taktgröße frei einstellbar
- verändern der Tonhöhe kompletter Takte
- Ein- und Ausblenden be-

liebiger Stimmen

- Portamento zum Verändern der Frequenzwerte und Vibrato
- Tauschen der Noten benachbarter Kanäle
- Recordmodus
- Begleitmöglichkeit über das Keyboard
- intelligentes Filingsystem
- problemloses Einbinden in Programmen

In einer der nächsten Ausgaben werden wir uns mit diesem Softwarepaket etwas näher beschäftigen. Warten wir's ab.

Bezugsquelle:
Cachet
Ostendstr. 32
7253 Oestrigen
Tel.: 07253/22411

Fat Agnus mit 1 MB ChipMem

Der erste Chip der ECS-Reihe ist nun endlich erhältlich. Mit dem neuen Fat Agnus 8372A ist in der Grundversion nun ein MB Chip-RAM und kein Fast-RAM mehr vorhanden.

Der überarbeitete Blitter kann jetzt virtuelle Screens und Windows (bis zu 32000*32000 Pixel groß) verwalten und hat einen weiteren Geschwindigkeitszuwachs erfahren. Weniger "Abstürze" tragen zu ei-

ner höheren Systemsicherheit bei. Der neue Fat Agnus soll laut Hersteller problemlos in den Amiga 500 und 2000B eingebaut werden können. Der Preis liegt bei knapp 300,- DM.

Bezugsquelle:
X-pert Computer Services
Weierwiese 27
6270 Idstein
Tel.: 06126/8809

Imagemaster – Bildverarbeitungssystem für den Amiga

Der Amiga macht auch vor Wissenschaft und Forschung nicht halt. Imagemaster nennt sich eine Bildverarbeitung, die vor allem in wissenschaftlichen Bereichen wie Medizin, Biologie etc. Anklang finden soll. Mit dieser Software lassen sich Videobilder (z.B. Makroaufnahmen) analysieren, wobei bis zu vier verschiedene Videokameras gleichzeitig angeschlossen werden können. Vom Programm aus wird dann die gerade aktuelle Videoquelle be-

stimmt. Die Aufnahmen werden per Knopfdruck digitalisiert und dann in 16 Graustufen ausgewertet. Diese Graustufen kann man durch Farbzuzuweisung (Auswahl aus 4096 Farben) auch bunt gestalten und das digitalisierte Bild auf einem Drucker ausgeben.

Bezugsquelle:
Micro Bio Tec Brand
Leihgesterner Weg 165
Postfach 5824
6300 Gießen
Tel.: 0641-2776 + 2777

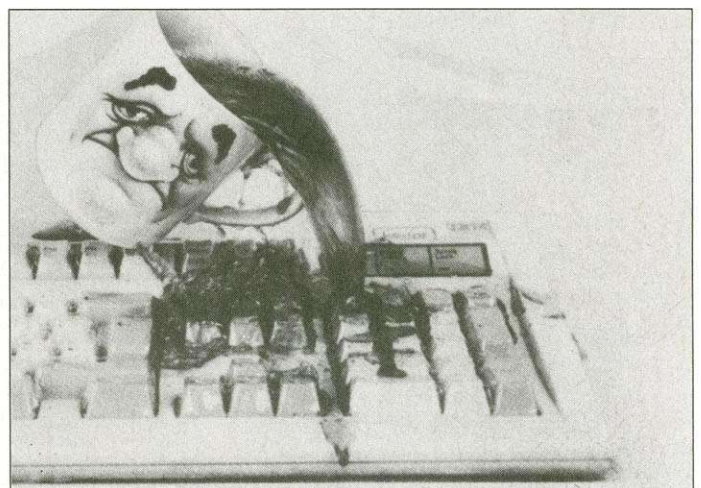
Keyboard-Cover schützt vor ungewollter Tastaturnässe

Da sitzt man vor seinem Computer und entspannt sich bei einer Tasse Kaffee oder Tee, und schwups, ist es passiert: Der wertvolle Inhalt Ihrer Tasse ergießt sich über die Tastatur! Das nachfolgende Lamentieren hat da auch keinen Sinn, denn bleibende Schäden an Ihrem Eingabemedium sind gewiß. Dies muß jedoch nicht sein.

Die englische Firma Kador hat eine Tastaturfolie entwickelt, die problemlos über die Tastatur gelegt werden kann, sich den Tasten anpaßt und die Tastatur während des Arbeitens vor unliebsamen Überraschungen schützt. Die flexible Folie wird einfach über die Tastatur gelegt

und schützt somit die Platine vor Staub und Feuchtigkeit, während sie ganz normal weiterbenutzt werden kann. Man kann sich diese zweite "Tastaturhaut" so ähnlich wie eine Einmachfolie vorstellen, wie sie bei der Marmeladezubereitung verwandt wird. Die Folie namens Seal'n Type kostet für die Amigas knapp 11 englische Pfund, ist abwaschbar und kann mehrmals verwendet werden.

Bezugsadresse:
Kador
Unit 4
Pontcynon Industrial Estate
Abercynon, Mid Glamorgan
U.K. CF45 4EP
Tel.: (0443) 740281
Fax: (0043) 741559



Bildvorlage: Seal 'n Type

Schon am frühen Vormittag hatte sich eine enorme Menschenmenge vor der Halle 6 versammelt, die Einlaß begehrte. Jedoch öffneten sich die Messtoren erst gegen 10:00 Uhr für die Besucher. Leider hatte der Veranstalter es versäumt, darauf deutlich hinzuweisen, so daß einige die Geduld verloren und ihrem Unmut freien Lauf ließen. Allem Chaos zum Trotz zeigte sich in der Halle 6 schon auf den ersten Blick, daß der Besuch sich lohnen würde. Aussteller wie Gigatron, Ariolasoft, Vortex, Software 2000, Kupke und viele andere zeigten ihre Leistungsfähigkeit. Neues und Interessantes konnte, bis auf wenige Ausnahmen, nicht nur bestaunt, sondern auch sogleich gekauft werden. Der ausgesprochene rege Zuschauerandrang führte dann auch in der Halle zu abenteuerlichen Situationen, und wer sich wirklich einmal genau umsehen wollte, der mußte 'drängelfest' sein.

Das Angebot an den Ständen war weit gefächert. Hardware, Software und viel Zubehör waren zu sehen.

Gigatron kündigt AMIGA-Laptop an

Gigatron dürfte vielen Amiga-Besitzern durch die dort angebotenen Speichererweiterungen bereits bekannt sein. Neben den Speichererweiterungen von 512 kByte, 1 MByte, 1,5 MByte und 1,8 MByte für den Amiga 500, die allesamt innerhalb des Gerätes Platz finden, waren drei neue Artikel von besonderem Interesse.

Als sehr nützlich gestaltet sich die neue AMIGA-Festplatte der Firma Gigatron. Mit einer Plattengröße von 2,5 Zoll findet der Winzling selbst in einem AMIGA 500 noch ausreichend Platz. Mit einer Kapazität von 20, 40, und 120 MByte dürfte auch den unersättlichen Speicherfans genügend Speicherplatz zur Verfügung stehen. Besonders auffällig ist die kompakte und professionelle Verarbeitung. Mit einer Größe, die in etwa die Ausmaße einer Zigarettenschachtel hat (inkl. Controller!), ist die Festplatte nur als winzig zu bezeichnen. Die aufgesetzte Controllerplatine ist vollständig in SMD Technik produziert.

Angekündigt, aber leider zur Messe noch nicht fertigge-

stellt waren ein AMIGA Laptop, sowie ein neues Motherboard für den AMIGA 500. Immerhin stehen bereits die technischen Daten beider Applikationen zur Verfügung. Der Laptop wird bei einem Gewicht von zirka fünf kg wahlweise mit einem Plasma-Bildschirm oder einem LCD-Display zur Verfügung stehen. Intern stehen 1 MByte zur Verfügung, die auf insgesamt 4 MByte ausgebaut werden können. Die Ausstattung umfaßt weiterhin ein 3,5-Zoll-Laufwerk sowie die bereits angesprochene 2,5-Zoll-Festplatte mit 120 MByte. Eine spezielle Tastatur mit Trackball macht auch den sonst notwendigen "Auslauf" für die Maus überflüssig. Mit Hilfe der Akkus wird eine Betriebsdauer von zirka acht Stunden unterstützt. Für stationäre Anwendungen kann das AMIGA-500-Netzteil weiterverwendet werden.

Das angekündigte Motherboard könnte ebenfalls dem AMIGA-500-Besitzer das Leben erleichtern. Auch dort ist bereits ein Einbauplatz für die 2,5-Zoll-Festplatte vorgesehen. Zur Geschwindigkeitssteigerung ist ein Sockel für die Prozessoren 68020/68881 vorhanden, mit Hilfe eines Umschalters kann der Anwender dann unmittelbar auf den gewünschten Prozessor zugreifen. Intern kann das Board bis auf 64 MByte ausgebaut werden. Der trotzdem nicht benötigte Speicherschacht wird dann mit Hilfe eines Jumpers zum Interface-schacht.

Die Firma X-pert präsentierte einen "aufgemotzten" Amiga unter dem Namen Black Tower. Das Gerät befindet sich in größter Ausbaustufe innerhalb eines Untertisch- (Tower-) Gehäuses; Tastatur, Monitor und Gehäuse sind von schwarzer Farbe und umge-

ben das Ganze mit einem geheimnisvollen Touch.

Die Anlehnung an Steve Jobs Computerwunder NEXT konnte dabei nicht verschwiegen werden. In der Grundkonfiguration stand ein Amiga 2000B mit dem neuen 1 MByte Fat Agnus zur Verfügung.

Der X-pert-Stand: Von Speichergiganten, Farbenkünstlern und schwarzen Schönheiten

Ein schwarzes Metallgehäuse stellt für eventuelle Erweiterungen jede Menge Platz zur Verfügung. Für Kühlung wurde (im Gegensatz zur Messehalle) wenigstens im Black Tower gesorgt: zwei Lüfter verrichten dort ihre Arbeit.

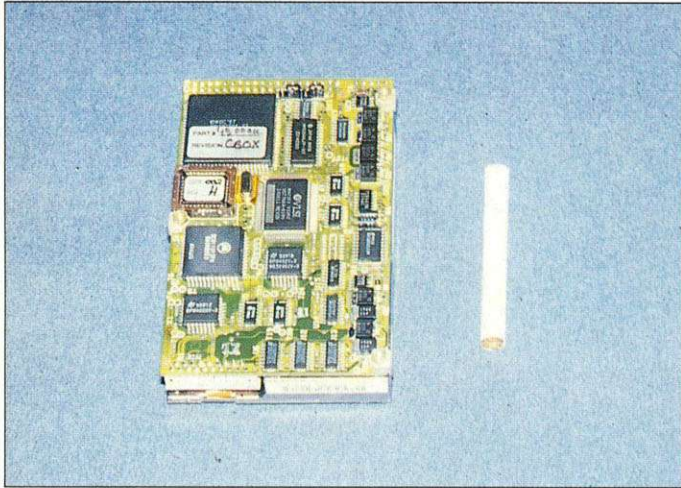
Wechselplatten war ein weiteres Argument des professio-

Die

AMIGA

Darauf haben die Amigaafreunde Deutschlands gewartet, die erste Messe die sich ausschließlich ihrem Computer widmete, fand vom 10.- 12.11. auf dem Kölner Messegelände statt. Händler, Hersteller und Anwender trafen sich in der Halle 6 zu einer stellenweise chaotischen, aber interessanten Ausstellung.





Die 2,5-Zoll-Festplatte zum internen Einbau in den AMIGA 500. Eine Zigarette demonstriert recht anschaulich die Größenverhältnisse

nellen Einsatzes vom Amiga. Auch hier konnte sich die Firma X-pert beweisen. Für knappe 2400 DM bot sie eine Sysquest-Wechselplatte mit 44-MByte-Cartridge an. In Zusammenhang mit dem Black Tower ein Gespann, das dem Heimanwender sicherlich genügen dürfte. Hinter dem Namen Tahiti verbirgt sich die erste wiederbeschreibbare Laser-Disk mit über 1 GByte (!) Speicherkapazität. Ebenso wie die Sysquest-Wechselplatte ist auch hier die Cartridge wie eine Floppy wechselbar. Die mittlere Zugriffszeit von 35 ms ist bei dieser Kapazität wohl eine sensationelle Leistung. Auch der Preis dafür ist sensationell: 12998 DM.

Farbenwunder Amiga könnte die Überschrift des nächsten X-pert-Artikels sein: 16 Mio. Farben werden gleichzeitig dargestellt. Der Name: Framebuffer. Bei dieser Karte handelt es sich um eine Art Video-Speicher, in dem die Videosignale sowohl extern (Videore-

korder, Bildplatte, Kamera etc.), als auch vom Amiga direkt gespeichert, eingefroren und angezeigt werden können. Durch die unglaubliche Speicherzeit von 1/15 Sekunde ist es möglich, direkt aus laufenden Filmen einzelne Bilder herauszuschnappen und in den Videospeicher abzulegen. Diese Bilder werden in den Originalfarben abgelegt. Dies heißt, daß bis über 16.000.000 Farben gleichzeitig möglich sind. Da der Amiga diese Farbenpracht allerdings nicht besitzt, kann per Knopfdruck die Farbzahl auf Amigaformat umgerechnet und als IFF-Bild abgespeichert werden. Dies ermöglicht eine noch nie dagewesene Qualität der digitalisierten Bilder.

Video, Video-Bildeffekte in Echtzeit

Ein tolles Video-Effekt-Gerät gab es am Stand von Intelligent Memory zu sehen. Dieser Digitizer mit Namen 'LIVE!' brachte hervorragende Ergebnisse auf einen Bildschirm: Während im Hintergrund das Video einer bekannten Popgruppe lief, wurden im Vordergrund die Standbesucher gezeigt. Der 'LIVE!'-Digitizer besteht im wesentlichen aus einer Steckkarte für den A2000. Er kann in s/w, mit 16, 32 oder 4096 Farben digitalisieren. Besonders hervorzuheben ist, daß der Digitizer in Realtime (Echtzeit) Bilder bearbeiten kann, dazu gehört auch das Verfremden von digitalisierten Bildern. Ausgestattet mit einem Video-Mischpult konnten die Aussteller die Besucher des Stan-

des ganz schön beeindruckten, die Vorführung war fast schon eine perfekte Videoschau.

Der 'LIVE!'-Digitizer ist für 1398 DM zu haben.

Sound- und Musik-Mache – Von professionellen Geräten bis zu Heimwerkzeugen

Auch die Sounderzeugung war ein Höhepunkt auf der Amiga '89. Hier wurde gezeigt, was in dem Amiga steckt. Von der Midi-Station, die im professionellen Rahmen Anklang findet, bis hin zu Musikprogrammen für den Heim-Musiker war allerhand vertreten. Die Softwarefirma Turtle Byte präsentierte ein neues Musikprogramm namens SIDMON. Es handelt sich dabei um einen Musikeditor mit dem innerhalb kurzer Zeit Eigenkompositionen in Studioqualität erstellt werden können. Die Bedieneroberfläche ist sehr funktionell gestaltet und dabei äußerst leicht zu bedienen. Mit Hilfe einer Play-Routine können Kompositionen in selbst erstellte Spiele oder Intros eingebaut werden. Als Höhepunkt dieses Programms dient eine Option, mit der die ADSR- (Attack-Decay-Sustain-Release) Werte einzeln einstellbar sind. Weiterhin ist es möglich, beliebige Wellenformen zu mischen. Auch der Preis ist relativ gering: zirka 90 DM. Wie uns angekündigt wurde, soll es bald eine MIDI-Version geben, die Sounds über eine geeignete MIDI-Schnittstelle an Keyboards transportiert.

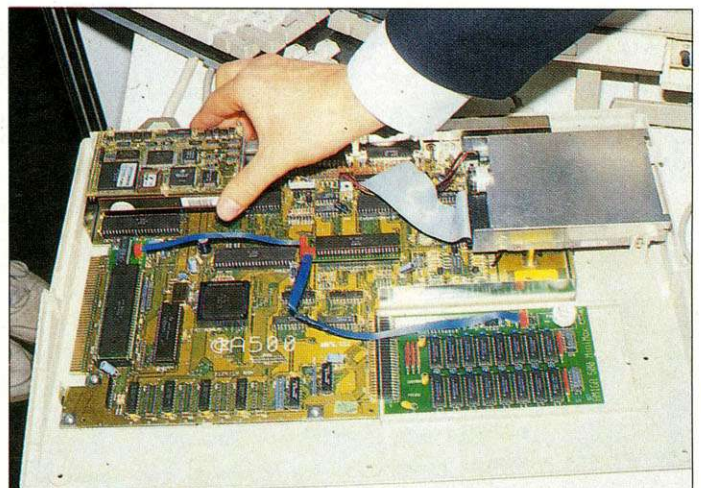
Turtle Byte brachte außer dem SIDMON noch ein Sprachenlexikon und Vokabeltrainer unter dem Namen 'Fit in Sprachen' heraus. Das Programm wurde von Pädagogen erstellt und komplett in Assembler geschrieben.

Ebenso neu ist der 'Playfield Designer', ein Programm zum Erstellen von Spielfeldern in beliebiger Größe, mit dem auch horizontales oder vertikales Scrollen des erstellten Spielfeldes möglich ist. Das Programm enthält 256 verschiedene Bausteine in 16 Farben und ist kompatibel zu Deluxe-Paint-Grafiken. Ein eigenes Malprogramm ist integriert, jede Funktion läßt sich problemlos mit der Maus steuern.

Der Preis für den Vokabeltrainer und auch für den 'Playfield Designer' liegt bei zirka 60 DM. Wir wollten uns eigentlich etwas ausführlicher einem Midi-Stand widmen, wo ein Amiga 500 seine Arbeit verrichtete. Ein zweiter Blick war unmißverständlich: da hatte sich doch glatt ein Atari ST auf die Amiga-Messe verirrt! Dieser diente als Midi-Server, wohingegen der Amiga für die Software zuständig war.

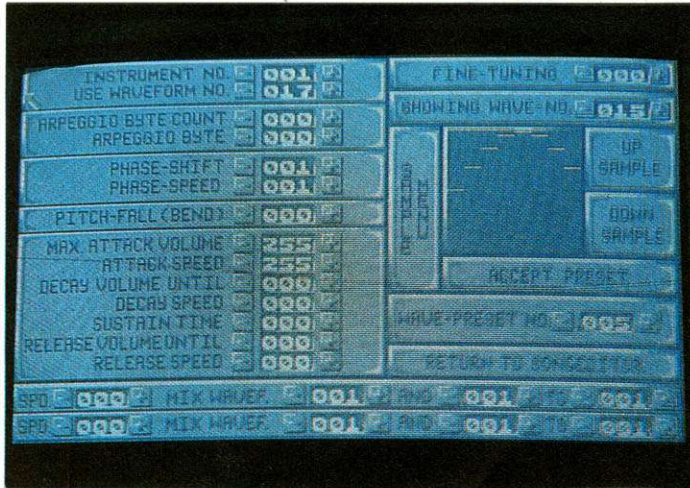
Ganz nebenbei bemerkt...

Auch das leidige Thema Viren kam zu Tage. Ein Aussteller konnte schon nach einer Stunde einen dieser "ungebetenen Gäste" verbuchen. Um diesem Übel vorzubeugen, hatten viele Aussteller zur einfachsten Präventivmaßnahme gegriffen: Sie klebten die Laufwerksöffnungen ein-



Aufgrund der geringen Größe findet die 2,5-Zoll-Festplatte auch innerhalb eines AMIGA 500 Platz

'89 in Köln



SIDMON heißt das neue Sound-Programm von Turtle-Byte, das bald auch in einer MIDI-Version erscheinen soll

fach mit Klebestreifen ab. Inwieweit dies geholfen hat, ist jedoch nicht bekannt, da bei dem Andrang auf die Messestände keinerlei Übersicht mehr möglich war. Sollte man den Gerüchten Glauben schenken, so durfte man davon ausgehen, daß die brandneue Software auf der am Sonnabend (angeblich) stattfindenden Copy-Party schon in Crackerkreisen herumging.

Btx – Der Informationsdienst der Post kommt auch zum Amiga

Auch Btx hatte Einzug in einige Stände gehalten, allem voran sind hier die Software-Decoder MultiTerm und Btx-Term zu nennen. Die Software-Decoder benötigen zum Betrieb an Btx nichts weiter als ein serielles Schnittstellenkabel und die Btx-Box der Post, daß Btx-Modem DBT-03, oder ein Modem. Modems

wurden ebenfalls auf den Ständen gezeigt, darunter welche mit einer Übertragungsrate von bis zu 2400 Baud. Btx auf dem Amiga bietet eine interessante Perspektive, zumal der Amiga dem CEPT-Standard folgen kann, er verfügt von sich aus über die Möglichkeit, 4096 Farben darzustellen. So fanden sich auch viele Interessenten an den Ständen ein; bevorzugte Übungsobjekte waren grafische Darstellungen und das Auffinden von Telefonnummern naher Verwandter. Nach wie vor stellen Spiele für den Amiga-Anwender ein hochinteressantes Thema dar, kein Wunder, die Maschine ist letztendlich einer der besten 'Spielcomputer', die es gibt. Auf der Amiga '89 war auch Ariolasoft mit einem Stand vertreten. Dort hatten wir Gelegenheit zu einem Gespräch mit zwei Mitarbeitern des Bitmap Brothers Teams. Diese Programmierer haben mit Xenon II einen echten Hit programmiert. Natürlich woll-



Eine schwarze Schönheit präsentierte sich auf dem Messestand von X-pert – der Black Tower

ten wir wissen, wie es bei den Bitmap Brothers weitergeht. Momentan befindet sich ein Spiel in der Entwicklung, das eine Mischung aus Action und Abenteuer werden wird und den Arbeitstitel Cadaver trägt. Jedoch interessierte uns vielmehr, wie die Bitmap Brothers arbeiten und wie sie es schaffen, derartig gute Spiele abzuliefern. Grundsätzlich wird in Teams gearbeitet und auf die Testphase besonderes Augenmerk gelegt. Während beim eigentlichen Programmieren wenig Aufregendes passiert, beginnt in der Testphase ein Prozeß, an dessen Ende ein fertiges Spiel steht. Während des Testes allerdings stehen Spieltester und Programmierer in enger Korrespondenz. Denn hier wird aus einem Programm ein Spiel gemacht. Der Schwierigkeitsgrad in jeder Phase des Spiels muß stimmen.

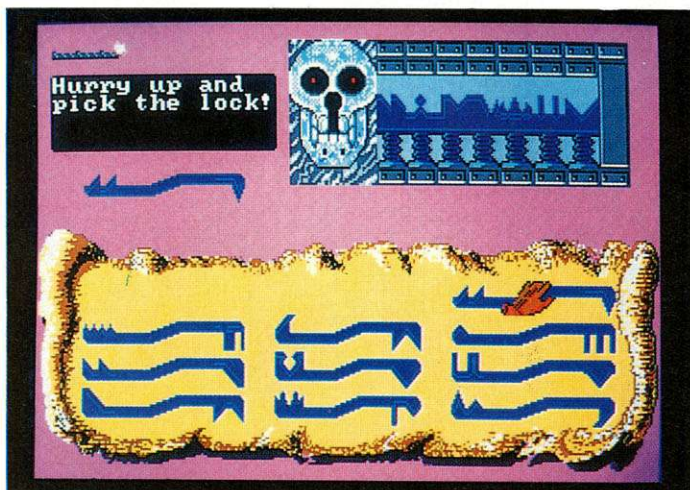
Rushware stellte ebenfalls aus und präsentierte neue Produkte. Besonders interessant dürften die Dungeons & Dragons Rollenspiele sein. Hillsfar, ein Abenteuer in einer mittelalterlichen Stadt, präsentiert sich komplett in deutscher Sprache. Zwar hätte man an der Grafik noch einiges verbessern können (man merkt ihr deutlich die EGA-Herkunft an), aber trotzdem ist Hillsfar ein feines Rollenspiel geworden. Wie von Rushware zu erfahren war, werden auch alle folgenden 'Dungeons & Dragons'-Produkte als Amiga-Versionen und vor allen Dingen in deutscher Sprache erscheinen. Turtle Byte Software hatte sich anlässlich dieser Messe große Mühe gegeben und

stellte gleich mehrere Spiel- und Anwendungs-Neuerscheinungen vor. Darunter erschienen uns insbesondere Cosmo Ranger, Sol 2000 AD und My Funny Maze gut gelungen zu sein. Sol 2000 ist ein Actionspiel in dem es um einige Raumstationen geht, die nicht mehr das tun, was sie eigentlich sollten.

Das Indiana Jones Grafik Adventure für den Amiga wurde auf dem Rainbow-Arts-Stand vorgestellt. Auch hier wurde das komplette Programm und auch alle Packungsbeilagen in Deutsch gestaltet. Scheinbar sind die Zeiten der rein englischen Spiele endlich vorbei. Ansonsten hatten sich die deutschen Softwarehersteller eher bedeckt gehalten, was sicherlich auch daran lag, daß bis kurz vor der Messe niemand wußte, ob die Show nun ein Erfolg oder ein Flop wird. Als sich am Abend des ersten Veranstaltungstags abzeichnete, daß die Amiga '89 ein echter Erfolg werden würde, bemühten sich noch einige weitere Aussteller nach Köln, wie z.B. Reline Software.

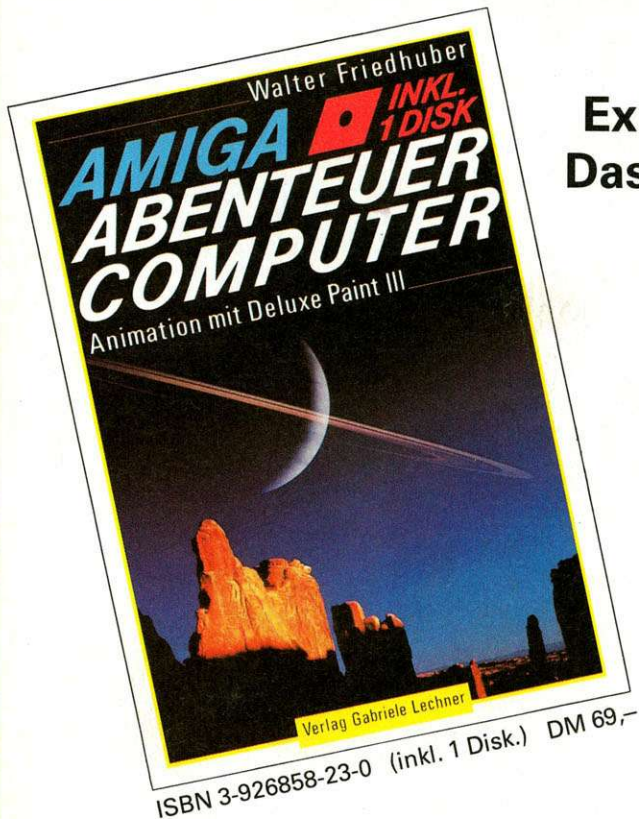
Alles in allem bot die Amiga '89 mehr als erwartet. Dank des regen Besucherstromes wurde die Messe für die Benutzer des Amiga tatsächlich das Podium für Neuheiten und Informationsaustausch. Auch wenn die Organisation an manchen Stellen der Veranstaltung sehr zu wünschen übrig ließ, konnte von Seiten der Aussteller ein positives Fazit gezogen werden. Schon heute ist die Amiga '90 beschlossene Sache, auch wenn es bisher keine Informationen über den Veranstaltungsort gibt.

(hs/jb/mm/br)



Screenshot Hillsfar Amiga

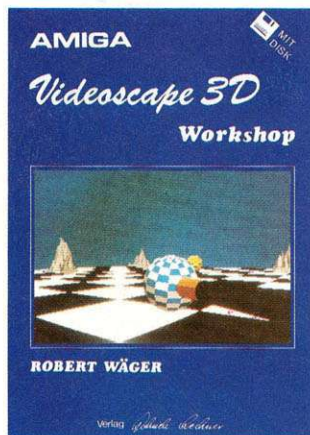
Abenteuer Computer



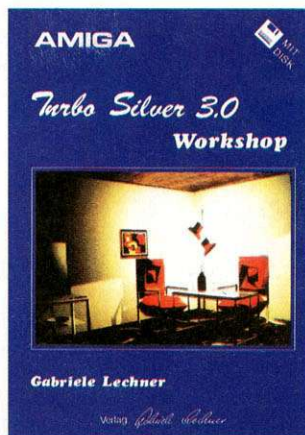
Exklusiv auf der Ami Expo Das Buch zu Deluxe Paint III

Das Buch ist eine unerschöpfliche Fundgrube für alle, die aus dem Alltag ausbrechen und eigene Ideen verwirklichen wollen. Hier werden Techniken gezeigt, die sowohl dem Anfänger als auch dem Fortgeschrittenen die faszinierende Welt der Computeranimation und -illustration auf unterhaltsame Weise näherbringen. Die Grundlagen des Programms werden anhand detailliert ausgearbeiteter Beispiele erläutert: Actiongeladene 2D- und 3D-Animation, Video-Überspieltechniken, ein wahres Feuerwerk an Effekten, Tricks – wie man sie bisher nur aus dem Kino kennt.

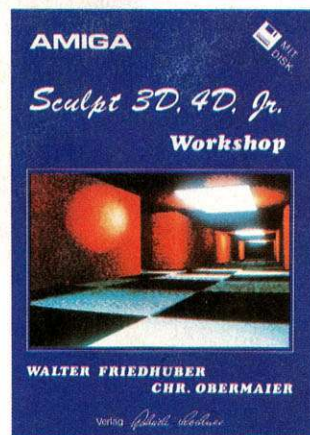
Die Amiga Workshop Reihe



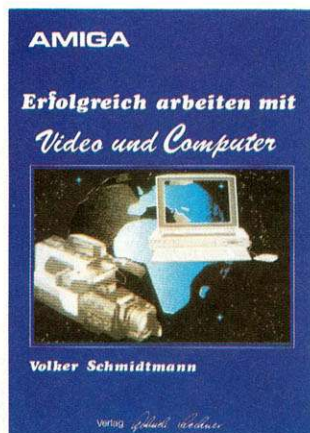
Videoscape 3D Workshop
ISBN 3-926858-03-6
248 S., inkl. 1 Disk **DM 59,—**



Turbo Silver 3.0 Workshop
ISBN 3-926858-12-5
330 S., inkl. 1 Disk **DM 69,—**



Sculpt 3D, 4D, Jr. Workshop
ISBN 3-926858-10-9
340 S., inkl. Disk **DM 69,—**



Erfolgreich arbeiten mit
Video und Computer
ISBN 3-926858-14-1
450 S., **DM 69,—**

Weitere Verlagserzeugnisse

Professionelles Arbeiten mit Deluxe Paint II
ISBN 3-926858-04-4, 540 S., über 200 Abb.
Disk-Set dazu (2 Disketten)

DM 69,—

DM 39,—

Im Brennpunkt: THE DIRECTOR
ISBN 3-926858-19-2, 112 S.

DM 29,80

Computermalschule Fantasy 204 S., inkl. 2 Disk

DM 59,—

Computermalschule Landschaften 160 S., inkl. 2 Disk

DM 59,—

Computermalschule Trickfilmzeichnen, 180 S., inkl. 2 Disk

DM 59,—

Alle Verlagserzeugnisse können direkt beim Verlag (+ Porto) oder über den Buchhandel bezogen werden.

Österreich exklusiv
Intercomp
Heldendankstr. 24
6900 Bregenz
Tel. 05574/273 44

Verlag Gabriele Lechner

Planegger Straße 1 · 8000 München 60
Telefon 0 89 / 834 05 91



WORLD OF COMMODORE



Elektronikshow in Valley Forge



Foto: Ausstellungskatalog Japanische Plakate

Valley Forge wird oft als das Silicon Valley des Ostens bezeichnet – dank der vielen Software-Entwicklungs-Firmen, die hier in dieser verschlafenen Region von Pennsylvania zu finden sind. Es ist zufällig auch der Standort der letzten WORLD-OF-COMMODORE-Show. WORLD präsentiert nicht nur das Neueste an Amiga-Computer-Produkten, sondern es ist auch der letzte Ort, wo noch neue Soft- und Hardware für den Dauerbrenner C-64 gesehen werden kann (da Commodore selbst diese Show sponsort...).

Wir gehen zum Convention Center, das unterhalb des Raddison Hotels gelegen ist – warten geduldig darauf, daß die Show offiziell eröffnet wird, und gehen hinein. Zusammen mit uns warten um die 1000 andere – viele von ihnen Spielefans, die gehört haben, daß es hier um Spaß geht – nicht Produktivität.

Innerprise bauen noch ihren Stand auf, aber es hat sich schon eine Menschenmenge versammelt, um BATTLE SQUADRON zu erproben. Es ist keine Frage, welchem Genre dieses Spiel zugeordnet werden muß: Shoot'em up. Ihr Fahrzeug fliegt vor und zurück über die Länge des Bildschirms, während senkrecht scrollende Landschaften eines mechanisierten Planeten vorbeiziehen. Landschaften, die Feinde in einer konstanten Geschwindigkeit ausbrüten, die dann versuchen, Sie zu vernichten – in einem Spiel, das die beste Höchstgeschwindigkeits-Animation darstellt, die es je für den Amiga gab.

Die Handlung ist einfach. Sie und, wenn erwünscht, ein Partnerschiff müssen alle feindlichen Mächte vernichten, die den Planeten Terrania heimsuchen. Das Fahrzeug kann sich über die gesamte Länge des Bildschirms bewegen und wird mit dem Joystick kontrolliert (auf Feuer drücken, um zu schießen). Wenn erst einmal alles ruhig geworden ist, ist es Zeit, zum inneren Kern vorzudringen, um die Arbeit zu beenden. Es gibt sogar ein Signal, das Ihnen zeigt, wo der Eingang ist. Sie müssen natürlich vor ein paar Problemen auf der Hut sein, bevor Sie eintreten können.

Neue Spiele – Ohne Extras geht nichts mehr

Am Anfang der Liste erscheint eine Vielzahl von fremden Fahrzeugen. Sie kommen auf den Bildschirm; alleine, in Paaren oder in

Gruppen. Und sie schießen natürlich auf Sie – und stürzen sich in Kamikaze-Manier auf Ihr Schiff. Währenddessen tauchen von unten selbst-angetriebene Geräte auf, die auf Sie feuern. Die meisten von ihnen schützen sich vor wiederholten Schüssen. Sie sehen zwar lustig aus, wenn sie sich öffnen, aber sie singen nicht "Happy Birthday".

Um das Ganze noch interessanter zu gestalten, fügten die Programmierer noch ein paar Panzer hinzu, die auf einem Schienennetz operieren und mit einer beweglichen Kanone ausgestattet sind. Mit diesen ist natürlich auch nicht zu spaßen, denn Sie haben einen mächtigen Frontlaser, der alles pulverisiert, was sich ihnen in den Weg stellt. Manchmal erscheinen besondere Zeichen in dem Schutt, die nicht nur Punkte und Schutzschilder bringen, sondern auch vier spezielle Extrawaffen verleihen. Das Symbol für jede Waffe trägt eine bestimmte Farbe (rot, grün, blau, gelb). Durch die Aufnahme mehrerer Symbole einer Farbe wird die Feuerkraft der Extrawaffe gesteigert. Jede Waffe schießt auf eine andere Art – am meisten hat uns die eine begeistert, die ein Rad voller zerstörerischer Energieteilchen freigibt. Diese Teilchen schwingen um das Schiff herum wie ein wild gewordenes Jo-Jo und lassen alles wegschmelzen wie Wachs unter einem Lötkolben. Ein weiterer Höhepunkt ist es, wenn der gesamte Bildschirm in einer besonders starken Explosion aufgeht. Die Soundeffekte sind wirklich gut. Die Musik hat genau die richtige Menge an dröhnendem Baß.

In einer ganz anderen Art präsentiert sich PERSIAN GULF INFERNO. Sie spielen hier ein Kommando, das eine Ölplattform von einer Gruppe böser Terroristen befreien muß. Während Sie sich durch die Levels der Plattform bewegen, müssen Sie einen Wissenschaftler finden, der als Geisel festgehalten wird – denn er ist der einzige, der die Atomwaffe entschärfen kann, die irgendwo versteckt ist. Die Figur, die Sie kontrollieren, bewegt sich in alle Richtungen, steigt Leitern hinauf und hinab und kann sogar eine Vorwärtsrolle machen, bevor sie aufsteht und mit ihrer Waffe feuert. Es treiben sich eine Menge böser Gestal-



Foto von M. M. Rosenthal

Ein neues Ballerspiel. Battle Squad macht auf den ersten Blick einen guten Eindruck

ten herum, die schnell gefunden werden müssen – wenn sie da sind, teilen sie es Ihnen mit, indem sie auf Sie schießen! Und sie schreien in irgendeiner unverständlichen Sprache.

Sie beginnen mit zwei Sprengladungen, um geschlossene Türen zu öffnen, und mit einem Handgewehr mit zwei Magazinen. Es ist lebenswichtig, mehr Munition zu finden, und es hilft Ihnen sicherlich, wenn Sie eine Schrotflinte bekommen. Andere Extras enthalten eine Aufzugskarte, die das ermüdende Klettern zwischen den Levels der Plattform ausschaltet.

Die Grafiken sind klein. Das ist auch gut so, denn Sie sehen Blutspuren auf den Zielen, die Sie treffen (um es höflich auszudrücken). Dasselbe gilt natürlich auch für Sie. Dies ist kein Diplomaten-

oder Strategiespiel. Hauptsächlich geht es darum, sich schnell zu bewegen (die Bombe explodiert innerhalb von zwei Stunden in normaler Zeit) und alles zu beseitigen, das sich Ihnen in den Weg stellt. Die Soundeffekte sind grausig genau, und das Trappeln kleiner Füße (die große Gewehre transportieren) übertrönt Ihre eigenen Schritte.

Ein El Dorado für Commodore User

Sierras SPACE QUEST III ist augenblicklich für den Amiga fällig. Die zwei Programmierer von Andromeda stecken fürchterlich in der Klemme: sie wurden gekidnappt und müssen nun schreckliche Software schreiben. Nur Roger Wilco (mit der Hilfe von Ihnen und der Maus) kann sie retten. Tolle Grafik, feine

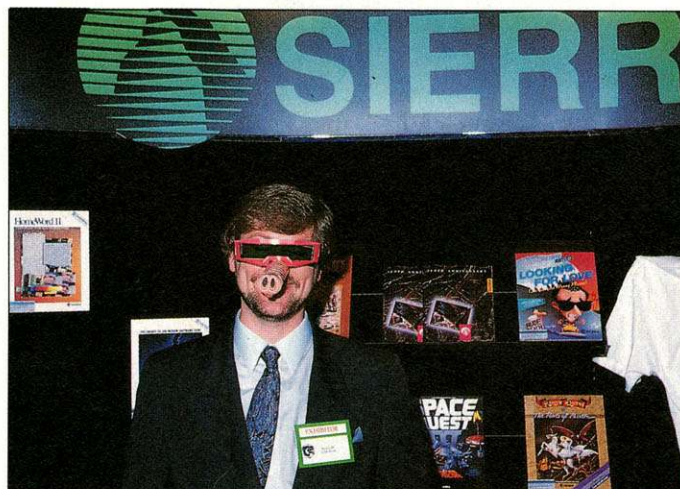


Foto von M. M. Rosenthal

Kauziger Humor ist eines der Sierra-Markenzeichen – auch auf ihrem Stand auf dieser Show

Soundeffekte und ein auf seine Weise außergewöhnlicher Humor erwarten Sie. Neue C-64-Hardware ist sehr rar – aber in der WORLD können Sie sie finden. Hier wurden einige Neuheiten für diesen Computer vorgestellt.

Zum Beispiel JiffyDOS, ein neues Betriebssystem für den 64er und die 1571-Floppy. Es wird auf ROM-Chips geliefert, die anstelle der ursprünglichen eingebaut werden. JiffyDOS vereinfacht das Anzeigen von Directories, das Formatieren von Disketten und andere Aktionen.

Mit GEODESK 2.0. liegt nun die neueste Version der grafischen Benutzeroberfläche für den C-64 vor. Neben den bekannten Desktop-Utensilien ist nun auch ein spezielles Grafikprogramm, GEO-PAINT, enthalten. Busy Bee stellte TALKER 64 vor. Dies ist die sprechende Version

des 'Write Stuff'-Textverarbeitungsprogramms. Es benutzt SAM (Software Automated Mouth), ein Programm, das schon vor einigen Jahren auf 8-Bit-Computern für Furore sorgte. Neun Stimmen, die vom Anwender definiert werden können, und viele Hilfsmenüs sorgen für eine einfache unproblematische Anwendung. Ursprünglich wurde TALKER 64 für Erzieher und Behinderte geschrieben, ist aber durch seinen Leistungsumfang für jedermann interessant.

Festplatten für den 64er zeigte die Firma Creatives. Diese Platten nutzen die 3,5-Zoll-Technologie und werden an den seriellen Bus des C-64 angeschlossen. Die Platten sind mit 20 oder 40 MBytes Speicherkapazität erhältlich. Sie wünschen sich innovative neue Hardware für den Amiga? Von VIVA wer-



Foto von M. M. Rosenthal

Die Jetsons-Zeichentrickhelden erobern den Computerscreen

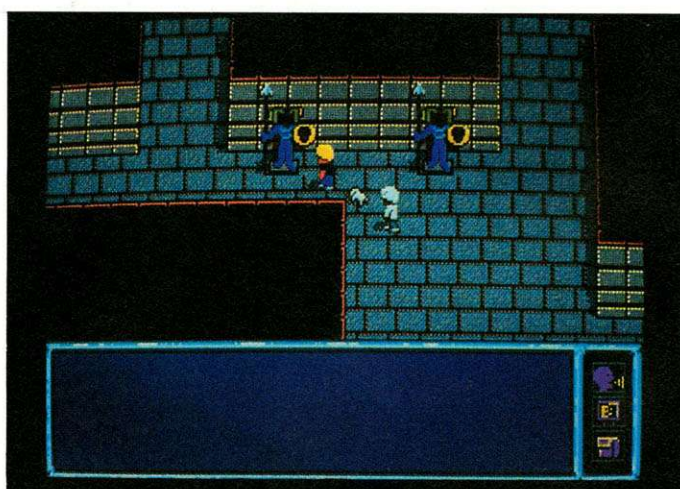


Foto von M. M. Rosenthal

Erleben Sie mit Jonny Quest und seinen Freunden die tollsten Abenteuer. Ein neues Action-Spiel für den Amiga

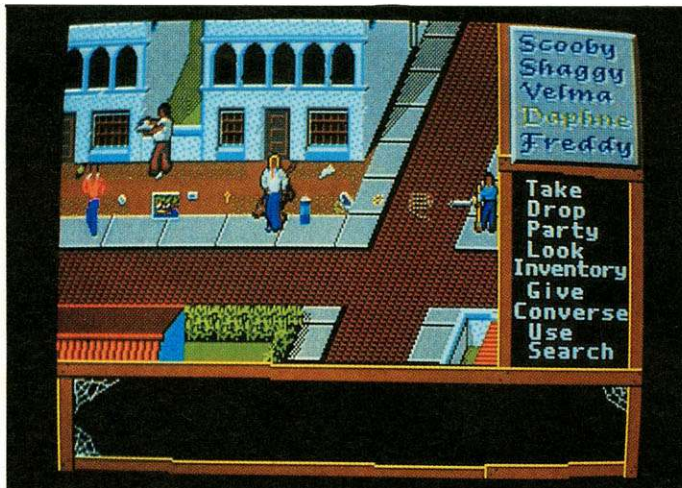


Foto von M. M. Rosenthal

Noch eine Zeichentrickserie, aus der ein Computerspiel gemacht wurde: Scooby Doo

den Sie nicht enttäuscht werden. Es ist ein Hyperware-Produkt, das Stapelspeicher benutzt, um Daten zu präsentieren, aufgebaut auf einer Serie von Informationen, auf die nichtlinear zugegriffen werden kann. Mac-User kennen schon seit einiger Zeit Hypercard – aber sie bekommen nicht die Farben und den Sound wie wir. Stapelspeicher zu bauen ist einfach und wird mit Ikonen gemacht. Besonders elegant wird es, wenn das Programm mit einem CD-Rom verbunden wird (dies erfordert ein besonderes Interface). VIVA läuft nur auf einem One-Meg-System, es erfordert 3 MByte, um ordentliche Programme zu schaffen. Übrigens, sehen Sie mal in den beiliegenden Weinführer. Tolle Soundeffekte und viele passende Bilder (schade, daß einige der Auflistungen unser Budget übersteigen).

Ein Monitor zum Anfassen macht die Maus überflüssig

Noch überraschender ist TOUCH SCREEN, der sich im futuristischen Design präsentiert. Er kann mit VIVA benutzt werden, aber es besteht auch die Möglichkeit, eigene Programme anzupassen. Die Bildröhrenoberfläche von TOUCH SCREEN ist berührungsempfindlich und reagiert auf einen Finger wie eine Maus. Professionelle Anwendungen sind reichlich vorhanden: das Auswählen von Musikvideos in Geschäften, Einkaufsauskünfte usw. Das macht aus dem Amiga ein

echtes Podium für Präsentationen. Der Preis von zirka 1000 \$ ist zwar relativ hoch, bedenkt man jedoch, daß man nebenbei einen guten RGB-Monitor erhält, so bewegt sich der Preis immer noch im akzeptablen Rahmen. Micro-Illusions präsentieren ihre Reihe interaktiver Bildgeschichten, die auf einigen wilden Zeichentrickproduktionen der 60er Jahre basieren. Sicherlich werden Sie sich noch an die Serie 'Die Jetsons' erinnern, die vor einigen Jahren auch bei Ihnen in Deutschland lief. In diesem Spiel verkörpern Sie George Jetson, ehrbares Mitglied der Gesellschaft, Familienvater mit einem recht eigenwilligen Verhältnis zu Sohn und Tochter (sie, Tochter im Teenageralter, er neunmalkluger 10jähriger Sohn). Und Jane, Ihre geliebte Frau, die Sie auf die Palme bringt. Zusätzlich ein lohnender und befriedigender Job in Sprockets Fabrik, wo Sie weiß was alles für Ihren ekligen Chef Spacely tun. Oh, und natürlich noch Astro, der Hund.

Die Jetsons als Computerspiel – Wiedersehen mit den beliebten Comic-Helden

Das Spiel hat ein grundlegendes Ziel: Überleben. Das ist sehr schwierig, denn Sie müssen keine außerirdischen Gestalten daran hindern, tödliche Raketenschüsse abzugeben – nein, Sie müssen versuchen, Ihren Job zu behalten und mit Boß Spacely auszukommen.

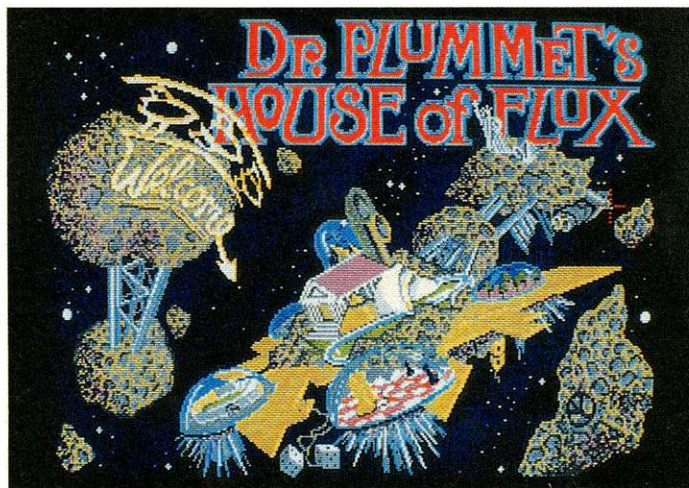


Foto von M. M. Rosenthal

Dr. Plummets House of Flux, hier dürfen Sie nichts, aber auch gar nichts ernstnehmen

Als Grafik-Adventure aufgebaut, gibt es Fenster, durch die Sie blicken können, um der Welt ins Auge zu sehen. Das größte Fenster zeigt Ihnen, wo Sie sind und wo sich Dinge befinden, die Sie unter-

suchen und aufnehmen müssen. Durch kurzes Anklicken können Sie in diese Dinge hineinsehen und sie drehen. Ein Verzeichnis, das Sie mitführen können, zeichnet auf, was Sie bereits gefunden ha-

Ein paar Worte mit Dave Needle

Sie kennen Dave Needle vielleicht nicht, aber Sie benutzen sein Werk gerade jetzt. Das bedeutet, daß der Amiga auf Ihre Eingabe in sein Werk wartet. Needle hat die Hardware für den Rechner entwickelt. Er wurde Meister im "kludging" genannt, die Technik, Plastik- und Metallteile so zusammenzubringen, daß daraus eine Maschine wird, die auf Ihre Befehle reagiert. Seit seinen Tagen bei Commodore war Needle damit beschäftigt, andere Meilensteine im Hardwarebereich zu schaffen: Die Atari-LYNX-hand-held-Farb-LCD-Spielmaschine, die ursprünglich von Epyx entwickelt wurde ist ebenfalls sein Werk. EPYX ist seitdem den Weg alles Sterblichen gegangen – und Needle hat sich neuen Projekten zugewandt. Wir trafen uns mit ihm, um ihm ein paar Fragen zu stellen.

Was denken Sie heute über den Amiga?

Needle: Das ist eine interessante Frage. Ich beginne gerade zu mögen, was ich über den Rechner höre. Als ich Commodore verließ, war ich nicht gerade erbaut davon, wie gewisse Dinge gehandhabt wurden – ich habe vieles davon verdrängt. Aber in den letzten Monaten ist viel getan worden, um den Rechner zu fördern.

Sie denken also, daß Commodore es jetzt geschafft hat?

Needle: Das ist schwer zu sagen. Teilweise scheint das Unternehmen das Richtige zu tun: die Werbung in den Staaten hat eine Menge verbessert – viele Software-Handelsketten verkaufen den Amiga jetzt genauso gut wie die Software. Aber es könnte noch besser sein.

Was meinen Sie damit?

Needle: Vieles scheint nur bruchstückhaft voranzugehen. Man braucht einen umfassenden und gesamten Überblick über das, was der Rechner tut und wofür er hier in den Staaten den Verbrauchern nützlich sein kann. Dann erst kann es dem potentiellen Käufer vermittelt werden.

Mit Dave Needle sprach Marshall M. Rosenthal



Produktfoto Beeshu Zoomer Joystick

ben. Probieren Sie eine auf Action basierende Bilder Geschichte, die sich um einen Jungen namens JOHNNY QUEST dreht. Nehmen Sie seinen Freund und Mitstreiter Hadji, Johnnys Hund Bandit, Dr. Benton und letztendlich den guten alten Race hinzu – und Sie haben die Bestandteile eines Adventures, von dem jeder Junge träumt. Ein Spiel mit einem solchen Materialreichtum, das noch dazu verschiedene Spielgenres in sich vereint, kann das überhaupt schlecht ausfallen? Adventure-Fans können die Maus benutzen, um herauszufinden, was in den geheimnisvollen Landstrichen vor sich geht, die es zu entdecken gilt. Auch Action/Arcade-Liebhaber kommen auf ihre Kosten, denn überall brauen sich Schwierigkeiten zusammen: Kampfszenen, versteckte Gänge, verschlossene Türen, die darauf warten, gewaltsam geöffnet zu werden. Je mehr Sie über das Spiel wissen, desto weniger ist es ein Geheimnis für Sie – leicht werden Sie es nicht haben, das Rätsel zu lösen. Hadji hilft Johnny mit

seinen magischen Kräften, die manchmal sogar funktionieren (aber leider nur selten, wenn wir uns recht erinnern). Bandit ist immer da, um hilfe reich seine Pfote auszustrecken – und wenn alles nichts mehr nützt, schalten Sie den Telekommunikator an Ihrem Handgelenk ein. Der größte Teil der Handlung findet im größten Bildschirmfenster statt. Direkt darunter ist ein Teil, in dem Sie Worte und Taten auswählen können. In großen Teilen des Spiels wurden übergroße Grafiken verwendet. Aber was noch wichtiger ist: die Animation (in 3D-Perspektive) bewegt sich sehr schnell. Einige Szenen wurden direkt aus dem ursprünglichen Comic digitalisiert. Der Fortgang der Geschichte wird über Textausgaben dargestellt, die Aktionen des Spielers erfolgen über Icons. Mit Jonny Quest können Sie ein richtiges Abenteuer erleben. Einmal unbeschwert tagträumen, sich durch einen Dschungel schwingen oder den Weltraum von bösen Buben befreien. Jetzt gibt es

noch SCOOPY DOO. Sie können dabei sein, wenn eine Gruppe von Freunden versucht, Anhaltspunkte zu finden, um eine Reihe von Abenteuern zu bestehen. Helfen Sie Shaggy, das Geheimnis des Meerestotopfes der Azteken zu lösen, besuchen Sie mit Velma eine unheimliche Geisterstadt, folgen Sie Freddy in ein Skiparadies und treffen Sie Daphne in Daymans unheimlichem Geisterhaus. Besondere Merkmale hat dieses "erschreckende" Spiel genug: 3D-Perspektive und verschiedene Arcade-Folgen. Eine der besten Spielsequenzen ist die Flucht vor dem absonderlichen Schneemenschen (die Gang rennt sich in bester Comic-Manier selbst im Weg herum). Zwei Schwierigkeitsgrade geben Ihnen die Möglichkeit, auch als Ungeübter Erfolg zu haben. Während des Spiels zeigt eine Übersichtskarte permanent an, wo Sie sich gerade aufhalten. Scooby Doo spielt in einem ziemlich unerforschten Landstrich, in dem es von Gefahren nur so wimmelt. Nur gut, daß jedes Mitglied der Truppe die Führung übernehmen kann.

Sarkasmus, Sex und fiese Robots – und was der Softwaremarkt sonst so hergibt

Testen Sie zum Schluß Dr. Plummet's House of Flux, das sich als ein Spiel mit wenig Ernst und viel Sarkasmus präsentiert. Es versetzt Sie auf einen zum Raumschiff ausgebauten Asteroiden. 28 bizarr und humorvoll gestaltete Bildschirme verteilen sich auf vier wählbare Aufträge, in deren Verlauf Sie jede Menge fieser Außerirdischer und einer ganzen Menge anderer unangenehmer Überraschungen in den Hallen Dr. Plummet's Festung begegnen werden. Die Verpackung bezeichnet JIGSAW als das "ultimate elektronische Puzzle". Tatsächlich kann man mit diesem Programm, das für einen oder zwei Spieler ausgelegt wurde, viel Spaß haben. JIGSAW zerlegt IFF-Bilder in einzelne Teile und mischt diese, so daß sie dann wie ein Puzzle wieder zusammengesetzt werden können. Der Anwender kann eigene Bilder in diesem Format benutzen und Sie von JIGSAW in ein Puzzle

verwandeln lassen. Auch die Anzahl der Teile ist variabel. Ein Puzzle mit vier Teilen ist schnell zusammengesetzt, aber wie sieht es mit 200 Teilen aus? Je nach Schwierigkeitsgrad mischt und verdreht JIGSAW die Bildteile. Ein eng bemessenes Zeitlimit sorgt dafür, daß Sie sich nicht zu lange Zeit lassen.

AMI von Free Spirit ist ein nützliches Werkzeug für alle die, die sich fragen, was gerade jetzt in ihrem Diskettenlaufwerk vor sich geht. Das Herz des Programms besteht aus vier Funktionen. Alles spielt sich auf einem Hauptbildschirm ab. Die Testabschnitte enthalten eine automatische Voreinstellung, die die verschiedenen Spuren überprüft, und eine Funktion, die die Rotationsgeschwindigkeit (r.p.m. – Umdrehungen pro Minute) des Antriebsmotors und die Stellung der Diskette im Verhältnis zum Antrieb regelt. Da AMI nicht kopiergeschützt ist, kann es auf der Festplatte installiert werden und mit einer besonders geeichten Diskette für Vergleiche benutzt werden.

Ebenfalls von Free Spirit ist die Trilogie von Text-Interaktiv-Spielen für Erwachsene. Dort taucht Captain Brad Stallion in Titeln wie "Sex Vixens from Space" und "Planet of Lust" auf. Das neueste Werk ist "Bride of the Robot", wo Sie Miss Galaxy vor einem sexbesessenen Roboter retten müssen. Man bewegt die Maus über eine Symbolleiste im unteren Teil des Bildschirms, wo Sie auch Befehle eingeben müssen. Auf dem Hauptbildschirm können Sie alles gut verfolgen (es sind leider keine digitalisierten Bilder, sondern nur 32-Lowres-Grafiken). Es ist alles ein bißchen unanständig, aber nicht sehr.

Unüberschaubare Menschenmassen an jedem Stand machen es unmöglich, die Spiele, die dort vorgestellt werden, einmal mit dem Joystick in der Hand auszuprobieren. Morgen werden wir die Show noch einmal besuchen. Vielleicht haben wir ja dann die Gelegenheit, einmal einen Joystick zu ergattern. (hs)

Einige Worte zum Autor:
Marshal M. Rosenthal ist ein Fotograf und Schriftsteller aus New York, dessen Arbeiten in aller Welt erscheinen. Er veröffentlicht auch "The Imagination Workshop", eine Zeitschriftenbeilage in Großbritannien, die einen Blick auf amerikanische Computer- und Elektronikunterhaltung wirft.

AMIGA

new  art

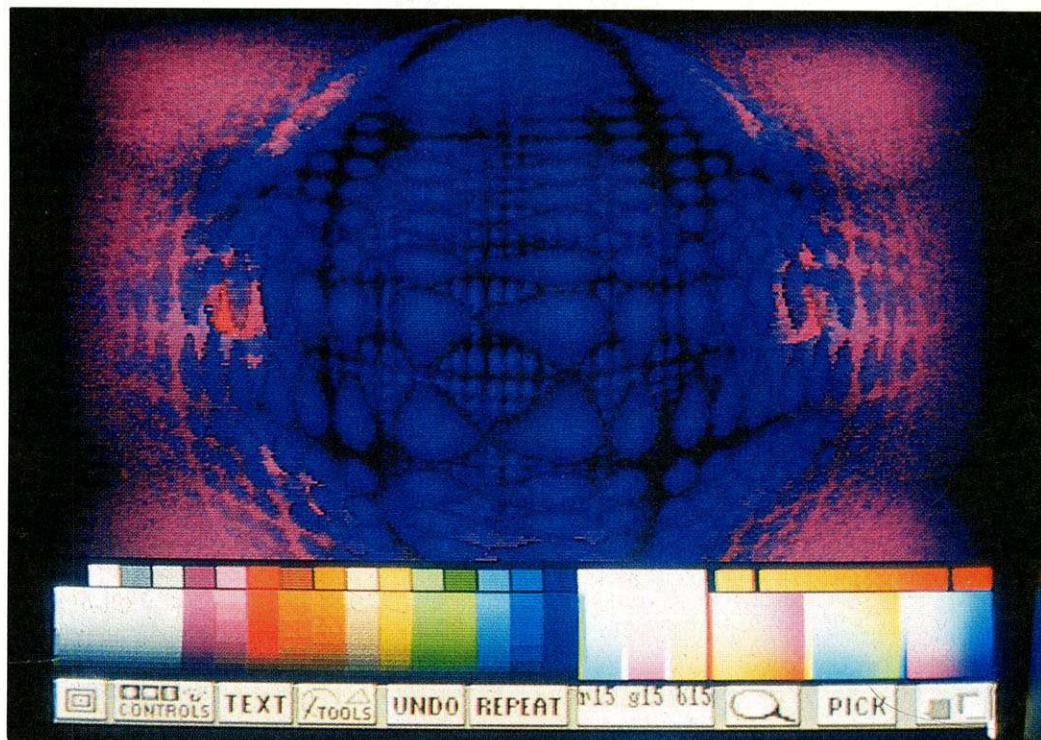
LIMITED EDITION



LIMITIERTE AUFLAGE
DESIGNED BY STEFANIE TÜCKING

Amiga 500 NEW ART.
Starke Farben und flippiges
Design. Der absolute Hit.
Für Leute, die immer wissen
wo's langgeht. Ab sofort
im Handel.
Nur in limitierter Auflage.


Commodore
ZUKUNFTSKOMPATIBEL



Mit wenigen Handgriffen zaubert Digi Paint 3 die tollsten Farbenspiele auf den Monitor

Digi Paint

Die grafischen Fähigkeiten des Amiga lassen sich auf die unterschiedlichste Art und Weise nutzen. Fast für jede Anwendung findet sich ein Grafik-Mode, in dem sich das Gewünschte adäquat realisieren läßt. Insbesondere der HAM-Mode (Hold-And-Modify) erweist sich durch die Möglichkeit, 4096 unterschiedliche Farben gleichzeitig auf dem Bildschirm darzustellen, als sehr leistungsfähig. Mit Digi Paint 3 liegt nun ein Grafikprogramm vor, das ein einfaches und effektives Arbeiten in diesem Mode ermöglicht.

Wer sich mit seinem Amiga schon einmal an der Digitalisierung grafischer Vorlagen versucht hat, wird wissen, daß eine Nachbearbeitung solcher Bilder mitunter bitter nötig ist. Das haben auch die Programmierer von NewTek erkannt (NewTek zeichnet auch für den Video-Digitizer Digiview Gold verantwortlich) und entwickelten Digi Paint, das nun in einer noch leistungstärkeren Version 3 vorliegt.

Zunächst muß gesagt werden, daß sinnvolles Arbeiten mit Digi Paint erst ab einem MByte-Speicher möglich ist. Zwar läuft Digi Paint auch auf 512-kByte-Computern, kann aber dann nur mit großen Einschränkungen benutzt werden. Zwar beinhaltet Digi Paint alle grundsätzlichen Funktionen, die ein Grafikprogramm ausmachen, seine wahre Stärke zeigt es aber erst,

wenn es darum geht, mit vielen (4096) verschiedenen Farben zu arbeiten. Die grundsätzlichen grafischen Werkzeuge sind eher spartanisch ausgefallen. Da gibt es freihändiges Zeichnen, eine Viereck-Option, Box, Kreis, Ellipse und Füllen.

Mit Digi Paint 3 4096 Farben voll im Griff

Hilfsmittel zum Zeichnen sind auch vorhanden. So existiert eine sehr gute Lupenfunktion und die Option, beliebige Bildausschnitte als Zeichenstift zu definieren. Bis hier unterscheidet sich Digi Paint 3 nur wenig von vergleichbaren Produkten, wenn da nicht die Möglichkeit bestünde, mit elf unterschiedlichen Arbeitsmodi zu arbeiten. Jeder dieser Modi beeinflusst die Wirkungsweise der Zei-

chenwerkzeuge. Wirklich interessant ist die Arbeit mit dem <Hot-Spot-Lokator>.

Wenn Sie z.B. eine räumlich wirkende Kugel zeichnen möchten, auf die von links oben Licht einfällt, so daß der Ball rechts unten im Schatten liegen muß, können Sie Digi Paint 3 mit dem <Hot-Spot-Lokator> entsprechende Anweisungen geben. Wurden die Einstellungen getätigt, braucht der Anwender nur noch die gewünschte Farbe anzuwählen und mit der <Circle>-Funktion einen Kreis zeichnen. Den Rest erledigt Digi Paint.

Ein weiteres leistungsstarkes Werkzeug ist <Transparency>. Mittels dieser Option können die Ränder jedes gezeichneten Objektes stufenweise ausgeblendet werden. Wird diese Option z.B. in Verbindung mit der Kreis-Option verwendet, erscheint ein Kreis, dessen äußere Ränder

durchscheinend erscheinen, so daß durch die Randbereiche des Kreises hindurch dahinterliegende Motivelemente zu sehen sind.

Dank leistungsstarker Werkzeuge ein überzeugendes Programmpaket

Auch mit selbstdefinierten Bildausschnitten läßt sich mit Digi Paint 3 einiges anstellen. So kann ein Bildausschnitt über eine beliebige Form (Kreis, Rechteck) gewickelt werden. Digi Paint berechnet automatisch die Wölbung, die das Motiv annimmt, wenn sie über die Form gewickelt wird. Außerdem läßt sich jeder definierte Bildausschnitt mit dem <Hot-Spot-Lokator> verknüpfen.

Digi Paint 3 beinhaltet auch das Utility Transfer 24, ein Programm, das unabhängig von Digi Paint 3 arbeitet. Transfer 24 ermöglicht den Import und Export von IFF-Dateien. Ferner können mit Transfer 24 die Paletten fertiger Bilder weiter modifiziert werden. Über Optionen wie Dither, Line Art und Black/White ist es sogar möglich, fertige Bilder generell zu verändern. Diese Effekte wirken global auf das ganze Bild. Wird z.B. das Mischverhältnis der Farben verändert, berechnet Transfer 24 das komplette Bild, entsprechend der Veränderungen, neu.

Weniger schön sind jedoch die Rechenzeiten, die der Amiga hier an den Tag legt. Für einen Kreis, der zwei Drittel des Bildschirms füllt, müssen schon fünf bis zehn Minuten Rechenzeit veranschlagt werden.

(hs)

AMIGA DOS Blitzlicht

Name: Digi Paint 3
Hersteller: NewTek
Quelle: Fachhandel
Preis: 199,- DM

Positiv:

- leistungsstarke Werkzeuge
- übersichtlich aufgebaut
- Transfer 24, starkes Utility zur Bild-Nachbearbeitung

Negativ:

- sehr zeitaufwendig
- Handbuch nur in Englisch

Zoetrope

Animationen spielend einfach



Seit Walt Disney 1932 den ersten abendfüllenden Zeichentrickfilm in die Kinos brachte, hat sich die Animation zu einem eigenständigen Medium entwickelt. Während sich Animationen früher nur durch den Einsatz horrend teurer Hardware realisieren ließen, leisten Heimcomputer heutiger Tage in diesem Bereich Dinge, über die Mr. Disney nicht schlecht gestaunt hätte.

Mit Zoetrope liegt nun ein neues Animationspaket für den Amiga vor, dessen Leistungsdaten beeindruckend sind.

Animation ist im Grunde nichts anderes als eine Illusion. Viele verschiedene Bilder sind jeweils für einen kurzen Moment zu sehen, wodurch die Eindruck einer Bewegung entsteht. Basierend auf diesem Grundprinzip arbeitet auch Zoetrope. Jede Animation besteht aus einer Animationssequenz, die wiederum aus einer Serie von Einzelbildern, sogenannten Frames, besteht. Der Anwender kann nun entweder jedes einzelne Frame per Hand zeichnen,

oder aber auf eines der leistungsstarken Animationswerkzeuge, die Zoetrope beinhaltet, zurückgreifen.

Animation klassisch oder technisch, das ist hier die Frage

Der Benutzer kann theoretisch bis zu 4096 Frames für eine Animation benutzen. In der Praxis fällt die Anzahl der möglichen Frames, je nach Speicherplatz, geringer aus.

Ist die Anzahl der Frames definiert, kann mit dem Erstellen einer Animation begonnen werden. Mittels der in Zoetrope enthaltenen Zei-

chenwerkzeuge können Grafiken unmittelbar in die Animationssequenz hineingezeichnet werden. Es wurde auch an eine Möglichkeit gedacht, den Bildinhalt eines Frames in andere zu übernehmen.

Animation kann aber auch anders aussehen. Z.B. ein Rechteck, das sich dreht und wendet. Hier brauchen nicht alle Frames einzeln mit der Hand gezeichnet zu werden. Eine einmal erstellte Form kann durch einen speziellen Zoetrope-Programmteil über jede beliebige Achse gedreht, gekippt, gewendet, vergrößert oder verkleinert werden. Wenn dies einzeln nicht genügt, der kann auch alle gleichzeitig aktivieren. Entsprechend wild sind dann die Kapriolen, die das Rechteck beim Abspielen der Animation auf dem Bildschirm vollführt.

Neben der Option, Flächen frei zu bewegen, stellt Zoetrope noch andere Werkzeuge zum automatischen Erstellen von Animationen zur Verfügung. Über die sogenannten Pixeleffekte kann eine Form hin- und hergerüttelt oder in wellenförmige Bewegung versetzt werden. Auch bei diesen Effekten reicht es aus, ein einzelnes Frame zu zeichnen. Die entsprechende Bewegung und die daraus resultierenden Veränderungen errechnet Zoetrope automatisch.

Das eigentliche Arbeiten mit Zoetrope gestaltet sich flott und komfortabel. Jedes Kommando kann über die Maus oder über einen Tastendruck aufgerufen werden. Nicht nur das Handbuch ist komplett in sauberem Deutsch gehalten und ermöglicht dem Anwender eine schnelle Orientierung, auch das eigentliche Programm präsentiert sich durch und durch in Deutsch.

Nach der Arbeit mit Zoetrope stellt sich irgendwann die Frage, wie fertige Animationen ohne Zoetrope weiterverwendet werden können. Auch daran wurde gedacht. Mit Zoetrope und Quickrif können

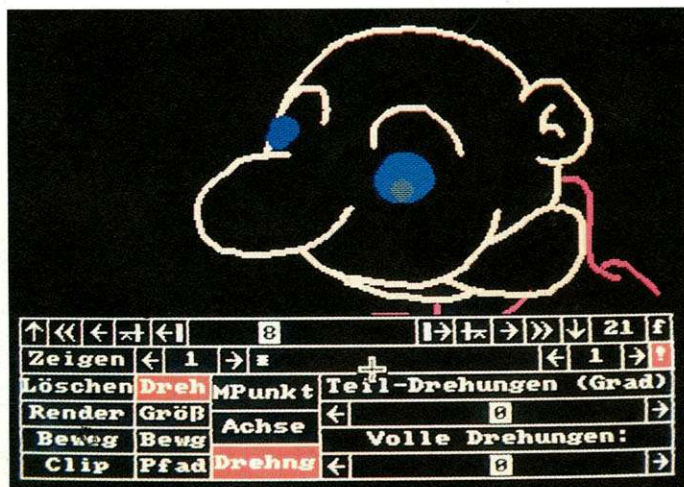
Animationen abgespielt werden, ohne auf Zoetrope zurückzugreifen. Beide Programme dürfen im nichtkommerziellen Rahmen frei kopiert und weitergegeben werden.

Animation ohne Zoetrope – Was passiert mit fertigen Trickfilmen?

Weniger komfortabel gestaltet sich die Arbeit mit Zoetropes Zeichenwerkzeugen. Jedoch ist es möglich, Bilder mit einem guten Grafikprogramm (DeLuxe Paint etc.) zu erstellen. Wurden diese Bilder mit der Extension 'Pic' abgespeichert, können sie problemlos in Zoetrope eingeladen werden. Alles in allem ist Zoetrope ein ausgereiftes Animationssystem, das nach einer gewissen Einarbeitungsphase flottes Arbeiten ermöglicht.

Den Disney-Studios können Amiga-Anwender mit Zoetrope zwar keine Konkurrenz machen, wer sich trotzdem an Computeranimation und Trickfilmen versuchen will, findet in Zoetrope das geeignete Werkzeug dafür.

(hs)



Als die Bilder laufen lernten. Animation auf dem Amiga, mit Zoetrope leichtgemacht

AMIGA DOS Blitzlicht

Name: Zoetrope
Hersteller: GFA Systemtechnik
Quelle: Fachhandel
Preis: 198,- DM
Besonderheiten: benötigt mindestens 1 MByte Speicher

Positiv:

- Programm und Handbuch in Deutsch
- gut strukturiert
- viele Effekte frei benutzbar
- Grafik kann importiert werden

Negativ:

- magere Zeichenoptionen

Endlich keine Listings mehr abtippen!

Nicht bei allen Programmen ist es mit drei Zeilen getan – gute Routinen und praktische Funktionen brauchen ihren Platz. Und bisweilen lassen sich auch lange Datenblöcke nicht vermeiden, ganz zu schweigen von Hexdumps und Assemblerlistings.

Schonen Sie Ihre Augen und schlagen Sie sich nicht die Nacht mit Abtippen um die Ohren. – Auf der Databox zum Amiga DOS-Heft finden Sie alle Listings als ASCII-File: passend für jeden Texteditor, den Amiga-BASIC-Interpreter, Makro-Assembler oder einen Compiler für C und Modula-2.

Alle Programme sofort nutzen

Da ist er nun endlich – der Trick oder das Programm, auf das Sie schon so lange gewartet haben! Zu allem Unglück ist das Listing aber in Modula-2 oder C, jedenfalls in einer Sprache, zu der Sie keinen Compiler haben, um ein lauffähiges Programm herzustellen.

Auch in diesem Fall hilft Ihnen die Databox von Amiga DOS aus der Patsche: Neben den Quelltexten im ASCII-Format finden Sie jeweils auch das fertige, lauffähige Programm. Sie brauchen es nur von der Databox-Diskette aus zu starten.

DATA

Alle Listings und Programme auf Diskette –

Keinen Ärger mehr mit Tippfehlern

Wer kennt das nicht, wenn das Programm nach dem Eintippen nicht läuft oder der Rechner gar abstürzt. Besonders gemein sind auch Fehler, die erst nach Wochen bei einer bisher nicht gebrauchten Funktion zu Tage treten, oder wenn der Druckfehlerteufel am Werke war.

Zermartern Sie sich nicht den Kopf, bis Sie die falsche Zahl im Datafeld gefunden haben. – Alle Dateien auf der Databox zur Amiga DOS sind vom Autor und der Redaktion auf Fehlerfreiheit geprüft und im dazugehörigen System "probegelaufen".

Sofort anfangen – ohne Zeitverlust!

Für was Sie sich auch interessieren, Interna des Amiga, Anwendungen, oder die Beispielprogramme der Werkstätten im Amiga DOS-Heft: Bevor Sie richtig an die Arbeit gehen können müssen Sie erst die benötigten Programme eingeben, korrigieren und zum Laufen bringen.

Sparen Sie sich diesen ermüdenden, unnötigen Aufwand an Zeit und Konzentration. Mit der Databox für Ihr Amiga DOS-Heft können Sie sofort den Dingen auf den Grund gehen, die Sie wirklich interessieren.

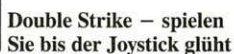


**2 Disketten
voller
Programme**

BOX

Computer einschalten - Diskette einlegen - los geht's

Basic
C
Modula-2
Assembler



Inland:		Ausland:	
Einzelpreis	24,- DM	Einzelpreis	24,- DM
zzgl. Versandkosten	4,- DM	zzgl. Versandkosten	6,- DM
Endpreis	28,- DM	Endpreis	30,- DM

DMV-Verlag · Postfach 250 · 3440 Eschwege



Basic			
C			
Modula-2			
Assembler			
Lauffähiges Programm			
Inhalt der Databox Amiga DOS 1/90			
X	X		ConText Amiga – eine super Textverarbeitung mit WYSIWYG
X	X		3D Shoot'em Up – mit zahlreichen Extrawaffen und packender Action!
X	X		Viruswürger – mit dem Ihre Virenprobleme bald der Vergangenheit angehören!
	X		Soundeditor – komplett in ASSEMBLER!
X	X		Transfer AMIGA-PC und C-64
X	X	X	IFF Handling
X	X	X	ASSEMBLER in BASIC einbinden
X	X		BASIC-Kurs – alle Beispielprogramme
	X	X	Maschinencode in M2-Amiga
	X		DELUXE PAINT 3 Werkstatt – alle Beispiele
	X	X	Aufsteigerwissen ASSEMBLER – alle Beispielprogramme

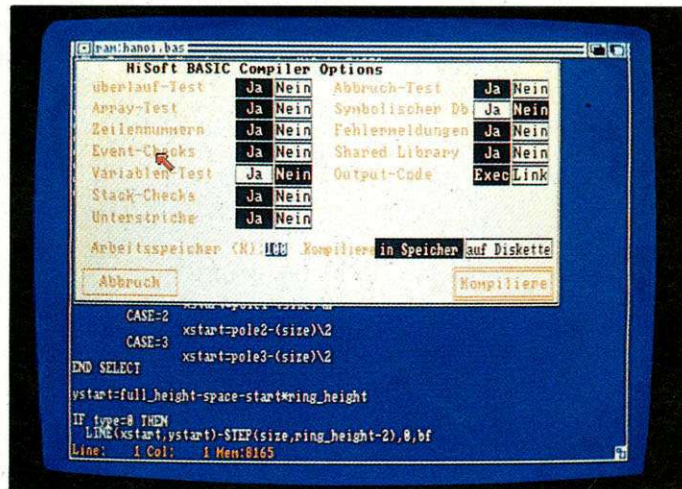
Das AmigaBASIC mag zwar viele Amigabesitzer begeistern, jedoch hat es auch einige Schwachstellen. Da wäre neben den bereits erwähnten Mankos die Tatsache, daß (weil Interpretersprache) die erstellten Programme nur dann lauffähig sind, wenn der Amiga auch auf den BASIC-Interpreter bzw. die AmigaBASIC-Libraries zugreifen kann. Die Firma HiSoft hat sich nun die Aufgabe gestellt, einen BASIC-Compiler zu erstellen, der diese Schwachstellen aus dem Weg räumen soll.

Dieser Compiler soll unter AmigaBASIC erstellte Programme problemlos compilieren und auch sogenannte "Stand-alone-Programme" erzeugen. HiSoft geht sogar noch einen Schritt weiter, indem auch BASIC-Dialekte, die auf anderen Computern erstellt wurden (z.B. Microsofts Quick BASIC und Borlands Turbo BASIC für MS-DOS-Rechner), ohne größere Umschreibungen auf dem Amiga importiert und compiliert werden können.

Der Inhalt der beiden nicht kopiergeschützten Disketten besteht aus einem benutzerfreundlichen Editor, dem Compiler und den Library-Dateien für den Compiler. Die zweite Diskette enthält zusätzliche Dateien und Programme für den fortgeschrittenen Programmierer. Weiterhin zeigen einige Demo-Programme die Effektivität des Compilers.

Das AmigaBASIC erhält neue Befehle: schneller, einfacher, leistungsfähiger

Die Befehlssyntax des HiSoft-BASIC entspricht im großen und ganzen der des AmigaBASIC, wobei der Befehlssatz des AmigaBASIC neue Features hinzubekommen hat. So sind unter anderem jetzt auch rekursive Programmtechniken, also sich selbst aufrufende Programmteile und Funktionen, möglich. Außerdem sind DO...LOOP- und SE-



Das übersichtliche Compiler-Menü – Alle Optionen sind durch einfaches Anklicken per Maus erreichbar

Dem BASIC Beine gemacht

BASIC - eine Sprache, mit der sich bestimmt jeder Einsteiger schon einmal vertraut gemacht hat, ist in manchen Programmiererkreisen eine oft verpönte Computersprache, da sie zum einen relativ langsam, und zum anderen für unstrukturierte Programmierung prädestiniert ist. Mit dem HiSoft-BASIC-Compiler soll sich dies ändern.

LECT...CASE-Schleifen implementiert worden.

Was die Portierbarkeit von anderen Rechnern angeht, sind jedoch einige Abstriche zu machen. Beim AmigaBASIC traten derweil zwar auch kleinere Probleme auf, die aber bei Beachtung einiger Regeln umgangen werden können. Schwierigkeiten beim Compilieren von AmigaBASIC-Programmen können dann entstehen, wenn AmigaBASIC-Variablen verwendet werden, die die gleiche Bezeichnung wie eines der reservierten Wörter des HiSoft-BASIC oder von Programmlabels besitzen. Hier macht sich der Nachteil bemerkbar, daß HiSoft-BASIC bei Variablenamen und reservierten Wörtern nicht zwischen Groß- und Kleinschreibung unterscheidet. Dies ist

jedoch für geübte BASIC-Programmierer recht einfach zu umgehen.

Ein Editor, der die Eingabe zum Kinderspiel macht

Der seitenorientierte Editor erweist sich als äußerst bedienungsfreundlich, da er Intuition unterstützt; die Amiga-typischen Eigenschaften, wie Windows, Pulldown-Menüs und die Mausbenutzung werden also übernommen, was natürlich nicht bedeutet, daß man das Arbeiten mit der Tastatur vergessen kann: Alle Befehle lassen sich auch von der Tastatur aufrufen.

Bei der Bearbeitung der ASCII-Dateien im Editor sind Blockoperationen, das Ausdrucken ganzer Texte bzw. Textteile sowie die Suchen- und-Ersetzen-Funktion möglich. Des weiteren kann der Compiler direkt aus dem Editor aufgerufen werden. In dem Compiler-Menü können Sie bequem per Mausklick Voreinstellungen (z.B. Fehlermel-

dungen anzeigen lassen) vornehmen. Hier kann man den Output-Code festlegen (lauffähiges Programm oder linkbare Objektdatei) oder entscheiden, ob der Compilerprozeß im Speicher vorstatten gehen bzw. die compilierte Datei auf Diskette abgespeichert werden soll. Durch das Abspeichern von linkfähigen Objektdateien lassen sich z.B. BASIC-Programme mit Assembler- und C-Routinen verbinden.

Compiler-Dialogboxen – Alles auf einen Blick

Wir ließen es uns nicht nehmen, einen Vergleich zwischen einer AmigaBASIC-Routine und einem compilierten File vorzunehmen. Hierbei ließen wir zuerst eine Variable innerhalb einer FOR...NEXT-Schleife 100000 Schritte durchlaufen. Der zweite Test beinhaltet eine WHILE...WEND-Schleife mit derselben Anzahl von Schritten. Tabelle 1 zeigt das Ergebnis. Anhand eines Tutorials im Handbuch, kann man leicht das Programmieren unter HiSoft-BASIC und den Umgang mit dem Compiler erlernen. Das Handbuch wurde übrigens so konzipiert, daß man sich anhand von Problemstellungen (unter anderem Türme von Hanoi) und deren Beseitigung langsam vorarbeiten kann.

(br)

AMIGA DOS Blitzlicht

Name: HiSoft-BASIC Compiler
Hersteller: HiSoft
Bezugsquelle: Markt & Technik
Hardwareanforderungen:
 - Amiga 500, 1000, 2000
 - mind. 512 kByte RAM
 - mind. 1 Diskettenlaufwerk
 - Kickstart 1.2 oder höher
Preis: 179,- DM

Positiv:
 - schneller und benutzerfreundlicher Editor
 - BASIC-Programme können mit Assembler- oder C-Routinen verbunden werden
 - umfangreiches deutsches Handbuch

Negativ:
 - bedingte Kompatibilität zu Amiga-fremden BASIC-Dialekten

Test 1:
 FOR...NEXT-Schleife mit 100000 Durchläufen
 AmigaBASIC: 40.827
 Compilat: 14.587

Test 2:
 WHILE...WEND-Schleife mit 100000 Durchläufen
 AmigaBASIC: 195.651
 Compilat: 53.290

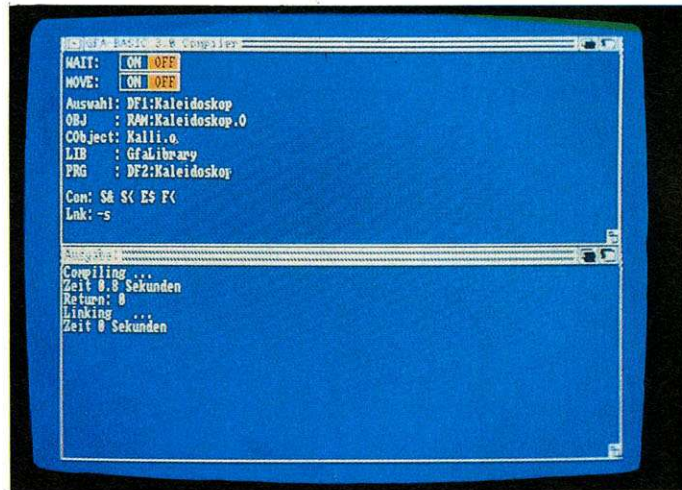
Wer bequem auf dem Amiga in BASIC programmieren will, der legt sich das GFA-BASIC zu. Nicht nur die einfachen Befehle sind es, die das AmigaBASIC erblassen lassen, nein besonders hervorzuheben ist die Geschwindigkeit, mit der GFA-BASIC-Programme ausgeführt werden. Doch GFA-Systemtechnik setzt dem noch einen drauf: Der Compiler zu GFA-BASIC macht die BASIC-Programme noch fixer. Außerdem ist nun endlich möglich, GFA-BASIC-Programme als "Stand-Alone" abzuspeichern, also die Dateien lauffähig zu machen, ohne daß sie auf den Interpreter oder die Datei GFABASRO zurückgreifen müssen.

Benutzerfreundlichkeit wird großgeschrieben

Die Handhabung des Compilers bzw. Linkers erweist sich als äußerst benutzerfreundlich, da alle Optionen direkt per Mausklick erreichbar sind. So können alle Voreinstellungen bequem modifiziert werden.

Im Sets-Menü läßt sich unter anderem festlegen, ob vor dem Öffnen einer Datei beim Kompilieren und Linken auf einen Tastendruck gewartet werden soll oder nicht. Der Sinn liegt darin, daß man einen Diskettenwechsel vornehmen kann (wenn man nur ein Laufwerk besitzt).

Eine weitere wichtige Voreinstellung liegt mit dem g3move-Befehl zugrunde. Er bewirkt, daß der Compiler sparsamer mit dem verfügbaren Speicher umgeht. Zu Beginn des "normalen" Kom-



Auf den GFA-BASIC-Compiler haben sicherlich schon viele gewartet. Wie schon beim Interpreter wurde hier sehr viel Wert auf Benutzerfreundlichkeit gelegt

GFA-BASIC Compiler Geschwindigkeit ist keine Hexerei!

GFA-BASIC ist ein Stichwort, das Atari ST- und Amiga-Programmierern das Herz höher schlagen läßt. Für luxuriöses BASIC-Programmieren ist diese Sprache prädestiniert. Um die ohnehin schon schnellen Programme noch schneller werden zu lassen, braucht man den GFA-BASIC-Compiler.

ginn des "normalen" Kompilierens, also wenn der g3move-Befehl inaktiv ist, wird der Source-Code in den Speicher geladen, das Kompilat wird dabei bis zum Absichern im Speicher gehalten. Falls man ein langes Programm kompilieren will, kann es aber passieren, daß der Arbeitsspeicher nicht ausreicht; hier hilft der g3move-Befehl: Wird dieser Befehl aktiviert, so wird jede übersetzte Prozedur aus dem Speicher entfernt, und es steht wieder wichtiger Speicherplatz zur Verfügung.

GFA-BASIC-Programme schnell wie der Wind: der Compiler macht's möglich

Der GFA-BASIC-Interpreter stellt von Grund auf 64 kByte Arbeitsspeicher zur Verfügung. Wenn Sie nun ein Programm einladen wollen, das mehr Speicher benötigt, müssen Sie mehr Speicherplatz reservieren. Der Compiler umgeht dieses Problem, in-

dem er das gesamte RAM belegt (minus 16 kByte für sich). Es lassen sich also auch längere Programme problemlos kompilieren, ohne daß eine Speicherreservierung vorgenommen werden muß.

Ein großer Vorteil des Linkers liegt darin, daß auch sprachfremde Routinen wie C-Funktionen und C-Libraries mit eingebunden werden können. Will man GFA-BASIC-fremde Dateien hinzulinken, so muß man ein spezielles, der Diskette beigelegtes Programm ablaufen lassen, da der Compiler keine standardmäßigen Objektdateien erzeugt.

Der Linker ist weitgehend kompatibel zu ALINK bzw. BLINK, was natürlich bedeutet, daß auch Assemblerprogramme, die z.B. mit dem A68k (Public Domain: u.a. Fishdisk 186) erstellt wurden, gelinkt werden können.

In dem Menüpunkt File kann man jeden Zwischenschritt, also Kompilieren, Linken und Austesten der lauffähigen Programme auch einzeln anwählen. Auch die Optionen zum Kompilieren und Linken lassen sich in einem Menüpunkt direkt mit der Maus anwäh-

len. Hier kann man z.B. den Speicherplatz bestimmen, den das Programm belegen soll, Errormeldungen unterbinden oder den Debug-Modus aktivieren.

Nun stellt sich die Frage, wie schnell das kompilierte File im Gegensatz zu dem "normalen" Interpreter-Programm ist. Dazu ließen wir eine Variable verschiedene Schleifen mit jeweils 100000 Schritten durchlaufen und stellten die Ergebnisse des GFA-Compilers denen des GFA-Interpreters gegenüber. Wie unsere Tests ausgingen, zeigt Tabelle 1.

Kleines Manko: Der Compiler erkennt Fehler bei Feldindizes nicht. Überschreitet eine Variable die ihm zugewiesene Dimensionierung, so bricht das kompilierte Programm mit einer Guru-Meditation ab. Der Interpreter erkennt diesen Umstand zwar und bringt auch eine Fehlermeldung "Feldindex zu groß", somit wäre es sinnvoll, alle Programme, die kompiliert werden sollen, zuerst einem Testlauf im Interpreter-Modus zu unterziehen. Ein Umstand, auf den im Handbuch jedoch ausführlich hingewiesen wird und bei fortgeschrittenen Programmieren leicht erkannt und umgangen werden kann. Auf der Diskette ist noch genügend Speicherplatz vorhanden, so daß man sowohl die Interpreter-, als auch die Compiler-Dateien auf einer Diskette unterbringen kann. Dies ist sicherlich anzuraten, nicht nur aus dem Grunde, da man aus dem Compiler-Menü die Option des direkten Interpreter-Aufrufes besitzt.

(br)

Test 1:
FOR...NEXT-Schleife mit REAL-Variable
Interpreter: 6.727
Compiler: 4.407

Test 2:
FOR...NEXT-Schleife mit INTEGER-Variable
Interpreter: 3.433
Compiler: 1.007

Test 3:
REPEAT...UNTIL-Schleife
Interpreter: 40.877
Compiler: 1.347

Test 4:
DO...LOOP-Schleife
Interpreter: 40.900
Compiler: 1.293

Tabelle 1: Testreihe

AMIGA DOS Blitzlicht

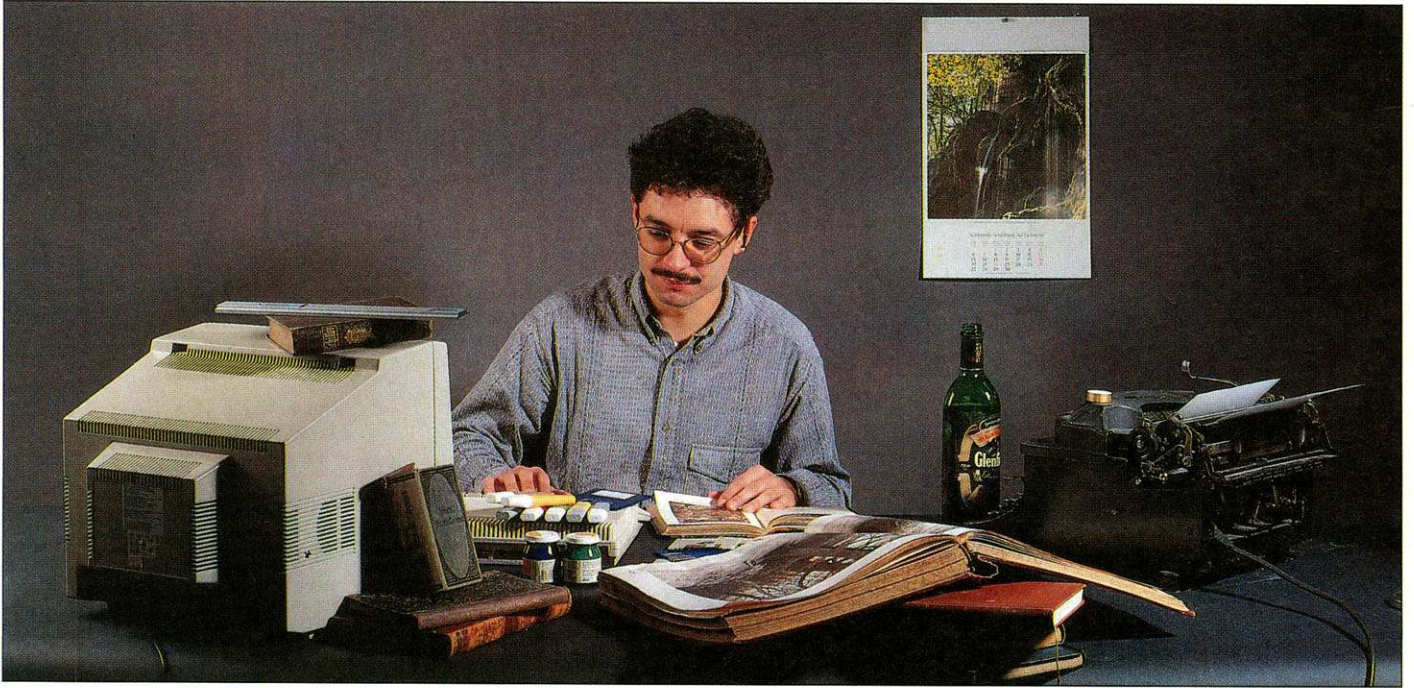
Name: GFA-BASIC-Compiler
Hersteller: GFA Systemtechnik
Preis: ca. 99,- DM

Positiv:

- Linker weitgehend kompatibel zu ALINK und BLINK
- einfache Handhabung durch Mausunterstützung
- Einbinden von C-Funktionen und C-Libraries möglich
- Ausreichendes deutsches Handbuch

Negativ:

- Dürftige Fehlerbeschreibung beim Linken
- Keine Fehlererkennung bei Feldindizes



Ich habe ein Malprogramm geschrieben

Bernd Blitter hat einen Amiga. Er programmiert in BASIC, und es macht ihm Spaß. Herr Blitter ist clever, denn er hat das Programm nicht in die Schublade gelegt, sondern er hat es an die AMIGA DOS geschickt. Dort fand man das Programm so gut, daß man es unbedingt haben wollte. Die haben es dann abgedruckt und Herrn Blitter **1500,- DM** für das TOP-Programm des Monats überwiesen.

Übrigens: Andreas Becker hat einen Disk-Monitor eingeschickt. Herr Becker hat gute Aussichten, den Jackpot in Höhe von **2000,- DM** zu knacken, der für das beste Listing des Monats außerdem zur Verfügung steht.

Auf jeden Fall sind sowohl Herr Blitter als auch Herr Becker bei der großen Sonderaktion "LISTING DES JAHRES 1990" dabei. Hier wählen die Leser die drei besten Listings, auf deren Autoren zusätzlich ein **5000,- DM** wartet. Senden Sie uns Ihr Programm (egal, ob Spiel, Anwendung, Soft- oder Hardware- tip) inklusive ausführlicher Beschreibung auf Diskette. Fügen Sie Ihrem Schreiben eine kurze Notiz bei, aus der hervorgeht, daß Sie der Autor des Programms sind und dieses bislang keinem anderen Verlag angeboten haben.

Auf geht's:

- Fertigmachen
- Eintüten
- Abschicken an:

DMV-Verlag
Redaktion AMIGA DOS
Stichwort: Listing des Monats
Postfach 250
3440 Eschwege

Floppys für AMIGA

komplett anschlussfertig, TEAC- und NEC Markenlaufwerke, stabiles Metallgehäuse, voll kompatibel, abschaltbar, durchgeführter Bus, 5,25"-Laufwerke mit 40/80 Track-Schalter, optional Write-Protect-Schalter (20.-DM)

3.5" mit Bus	222.-
3.5" digi mit digit. Trackdisplay und Bus	259.-
5.25" mit Bus	255.-
5.25" digi mit digit. Trackdisplay und Bus	309.-
3.5" intern NEC 1037A für A2000	189.-

Disketten

Disketten 3.5" 2S2D 10 Stck.	17,90
Disketten 5.25" 2S2D 10 Stck.	5,99

Festplatten

mit Omti - Controller (bis 480 KB/sec) und Autobootmodul

AMIGA 2000:

32 MB 5.25" (autoboot), 65 ms	799.-
42 MB 5.25" (autob.), 65 ms, mit Seagate Controller	899.-
65 MB 5.25" (autoboot), 28 ms	1249.-
32 MB 3.5" (autoboot), 40 ms	899.-
50 MB 3.5" (autoboot), 40 ms	1049.-
66 MB 3.5" (autoboot), 28 ms	1295.-
FileCard 30 MB, (autoboot), 40 ms	1095.-
FileCard 50 MB, (autoboot), 40 ms	1295.-

AMIGA 500/1000:

32 MB A500	895.-
42 MB A500 mit Seagate Controller	995.-
50 MB A500	1049.-
65 MB A500	1295.-

Stalter Computerbedarf GmbH

Gartenstr. 17 - 6670 St. Ingbert

☎06894/2012

2400 Baud MODEMS



TORNADO 2400 E

Tischgerät incl. Steckernetzteil
für alle Rechner mit RS 232/V.24

399,-

MAXMODEM 2400 MNP 5

Wie oben, mit MNP5 Übertragungs-
protokoll bis 4800 bps eff. Geschw.

599,-

PC-Karte, halbe Länge, COM 1:
bis COM 4: konfigurierbar

349,-

Lieferung per Nachnahme, komplett mit engl. Handbuch und Telefonkabel (USA). 1 Jahr Garantie, Rückgaberecht innerh. von 8 Tagen ohne Angabe von Gründen.

Leistungsmerkmale:

2400, 1200, 600, 300 Baud CCITT V.22 bis V.22/V.21, 1200 und 300 Baud Bell 212a/103, kompatibel mit HAYES SMARTMODEM 2400 (AT-Kommandos), automatische Wahl (Ton oder Impuls), Autoanswer, Konfiguration speicherbar, basierend auf INTEL Chipsatz.

Der Betrieb dieser Modems am öffentlichen Postnetz in der BRD und in West-Berlin ist verboten und unter Strafe gestellt.

Carl Schewe (GmbH & Co.), Abt. Modems

Essener Str. 97, 2000 Hamburg 62

Tel. (0 40) 5 27 03 21, Fax (0 40) 5 27 66 54

AMIGA-SOFTWARE

African Raiders	59.90	Dragon Ninja	79.90
Airball	59.90	Dragon Spirit	59.90
A.P.B.	59.90	Emerald Mines 2	39.90
Archipelagos	59.90	Exolon	49.90
Artic Fox	79.90	Falcon F16 Mission Disk	59.90
Balance of Power 1990	29.90	F.O.F.T.	89.90
Batman The Movie	79.90	Footb.Man.2 Exp.Kit.dt.	44.90
Battletech	69.90	Football Director	59.90
Beach Volley	79.90	Foundation Waste	29.90
Bio Challenge	69.90	Fugger	59.90
Bismarck	79.90	Galactic Conquerer	79.90
Black Magic	59.90	Garfield Winter's Tail	69.90
Breach	59.90	Gunship	79.90
Bundesligamanager	59.90	Hawkeye	69.90
Butcher Hill	59.90	Iron Tracker	59.90
Castle Warrior	59.90	Joan of Arc	59.90
Chariots of Wrath	79.90	Kickoff	69.90
Chinese Karate	59.90	King Arthur	59.90
Colossus Chess 10.0	59.90	Kingdom of England	79.90
Cosmic Pirate	79.90	Kult	79.90
Crazy Cars 2	79.90	LED Storm	69.90
Cybernoid 2	69.90	Legend of Djelen	69.90
Demon's Winter	59.90	Leonardo	59.90
Die Suche n.d. Vogel.	69.90	Lizenz zum Töten	59.90
Division	29.90	Lords o.th. Rising Sun	99.90
Dominator	59.90	Manhunter NY	89.90
		Maniax	29.90
		Marble Madness	29.90

**SOFORT-BESTELLUNG
PER TELEFON
09 11 / 28 82 86**

Mayday Squad	59.90
Millenium 2.2	79.90
Murder in Venice	79.90
Netherworld	49.90
Newzealand Story	79.90
Night Dawn	79.90
Olimperium	59.90
Paperboy	59.90
Paranoia Komplex	79.90
Personal Nightmare	89.90
Phantom Fighter	19.90
Phobia	79.90
Populous	69.90
Populous Scenery Disk	39.90
Precious Metal nurA500	79.90
Premier Collection	79.90
Prof. Music Assistant	29.90
Red Heat	89.90
Rick Dangerous	69.90
Roll-Out	59.90
Running Man	79.90

Run The Gauntlet	79.90
RVF Honda	69.90
Shogun	79.90
Shoot'em Up Constr.Kit	89.90
Silkworm	59.90
SK City	99.90
State of the Art	59.90
Skrull	59.90

PROGRAMME!

Sonix	139.90
Sorcerer Lord	79.90
Sound System Mark 2	69.90
Space Harrier	59.90
Star Wars Trilogy	79.90
Steve Davis' Snooker	59.90
Super Quintett	69.90
Superscramble Sim.	69.90
Superski	59.90
Testdrive 2 The Duel	79.90
Testdrive 2 Car Disk	25.90
Testdrive 2 Cal. Chall.	25.90
The Champ	79.90
The Story so Far 3	59.90
Thunderbirds	79.90
Timescanner	79.90
Tim and Struppi	59.90
Tom and Jerry	79.90
Tracksuit Manager	49.90
Triple X	59.90
Ult. Sound Tracker 2.0	99.90
Vindex	59.90
Voyager	79.90
Wanderer 3D	59.90
War in Middle Earth	79.90
Wayne Gretzky Icehock.	69.90
Wicked	79.90
World Class Leaderboard	29.90
World Tour Golf	29.90
Xenon 2 Mega Blast	79.90
Xybots	59.90
Yuppie's Revenge	79.90
Zak Mac Kracken dt.	69.90
Zork Zero	69.90

DAS Taktik-
Spiel mit Super-
Grafik: ARCHIPELAGOS
Unser Preis:
29.90

Public-Domain-Software

Wir haben über 2000!!! Disketten aus (fast)
allen Gebieten und PD-Serien, die wir ab
DM4.00 weitergeben Inhaltsverzeichnisse und
Kurzbeschreibungen auf:
2 Disketten für AMIGA... 10.00

Jetzt sehen Sie sich
mal hier um:

Armageddon Man.29.90	Goldrunner.....29.90
Black Jack.....29.90	Incr.Shr.Sphere...29.90
Black Shadow...29.90	N. M. Grand Prix...29.90
Bombuzal.....29.90	Quantox.....29.90
Craps Academy...29.90	Strike F. Harrier.29.90
Dark Castle....29.90	Tetra Quest.....29.90

BESTELLUNG + INFO ANFORDERUNG

☐ Hiermit bestelle ich für den Computer

☐ Nachnahme (+ Kosten 5,90)

☐ Vorkasse und Scheck (+ Kosten 2,50)

☐ Ich möchte ein kostenloses Gesamtinfo über Software für meinen Computer.

Bitte Anschrift nicht vergessen _____ Unterschrift

AMIGA DOS 1/90

T.S. Datensysteme · Denisstraße 45 · 8500 Nürnberg 80

Alle Preise sind unsere Ladenpreise. Bei Versand berechnen wir anteilige Selbstkosten:
bei Vorkasse mit Scheck: DM 2,50, bei Versand per Nachnahme DM 5,90 je Sendung.

T.S. Datensysteme

DENISSTRASSE 45 · 8500 NÜRNBERG 80 · TELEFON 09 11 / 28 82 86



Das Unsichtbare sichtbar machen

'Pro Access' läßt tief(er) blicken

So ein Spielchen auf dem Amiga ist je nach Ausführung spannend, aufregend, zur Konzentration anregend, lustig und noch einiges andere mehr. Was aber geschieht hinter den Bildschirmkulissen, sprich im Speicherbereich des Computers? Eine Frage, die mit dem Modul 'Pro Access' beantwortet werden soll.

Alle Programme eines Computers haben einen Nachteil, wir sehen nur das Endergebnis auf dem Bildschirm, ohne eine Eingriffsmöglichkeit – es sei denn, das Programm entammt unseren eigenen geistigen Ideen.

Normalerweise ist dies auch gut so, bei vielen Spielen allerdings kommt uns der Umstand, nur drei Leben für den Computerhelden zu haben, gar nicht so recht. Was also tun? Beim guten alten C-64 war es kein großes Problem, ein kleiner POKE-Finder reichte meistens völlig aus. Beim Amiga allerdings wird es schon schwieriger, einfache POKEs dieser Art werden wohl kaum erfolgreich sein. Dieser Umstand war wohl der Anlaß für die Firma Boehm und Zanger aus Lahr, ein externes Modul für die Amigas 500 und 1000 zu entwickeln,

welches laufende Programme einfriert und Einblick in den Ablauf des Programms gibt.

'Freezen', 'Bremsen' und 'Trainen' – Hilfe für geplagte Spieler

Das 'Pro Access'-Modul präsentiert sich in einem kleinen Pultgehäuse. An der einen Seite sitzt ein Platinenstecker zum Anschluß an den Amiga-internen Expansion-Port. Ein Potentiometer (regelbarer Widerstand) mit angeschlossenem Ein-/Ausschalter, ein Taster und zwei Leuchtdioden sind die einzigen Elemente, die sich dem Benutzer präsentieren. Das Potentiometer dient als Bremse mit Ausschaltung. Sie kann bei besonders schnellen Spielsequenzen genutzt werden, bei

denen es oft auf Zehntelsekunden in der Reaktion ankommt: Die Bremse verlangsamt den Prozessor in seiner Arbeitsgeschwindigkeit fast stufenlos, so daß Kollisionen zwischen Objekten besser beobachtet werden können. Der Taster dient zur Einschaltung des 'Pro-Access-Editors', dazu gleich mehr.

Die beiden Leuchtdioden haben Signalfunktion, die grüne LED signalisiert ein software- und hardwaremäßig abgeschaltetes Modul, mit anderen Worten: Ist das Modul betriebsbereit, leuchtet die LED nicht, wird es über den Editor abgeschaltet, leuchtet sie. Die rote LED leuchtet für die aktivierte Bremse.

Mitgeliefert werden zum Modul eine Programmdiskette mit einem Konvertierungsprogramm und ein Handbuch in Deutsch mit einer Referenzkarte über die Befehle.

Das Arbeiten mit dem Modul gestaltet sich auch für Anfänger recht einfach: Nur an den Expansion-Port anschließen, und schon kann es losgehen. Unser Versuchsobjekt Nummer eins war das Firebird-Spiel "Rick Dangerous". Der Held der Story hat insgesamt sechs Bildschirmleben zur Verfügung, eigentlich recht viel, wenn nicht seine (Sprite-)Gegner wären.

Wie man aus drei kurzen Leben den Unsterblichkeitsfaktor 1 macht

Nach dem Start des Spiels wurde das Modul mittels Taste aktiviert, es versetzt den Anwender vom Spiel in einen Editor, der verschiedene Befehle zuläßt. Uns interessierte jetzt als erstes der Trainer, der in der Lage sein soll, nach verschiedenen Angaben dem Spiel die Möglichkeit zu nehmen, Zähler für Leben oder Energie ablaufen zu lassen. Der erste Befehl, der hier zur Geltung kommt, heißt T(rainer). Durch Eingabe der zu suchenden Zahl, hier die 6, wird der Speicher nach dieser Zahl durchforstet, und zwar der Speicher, in dem das Spiel steht. Der Trainer meldet sich nach Eingabe von 'TS !6' (!6 für dezimal 6, hexadezimale oder binäre Zahlen sind ebenfalls erlaubt) binnen kürzester Zeit mit einer Endadresse wieder. Nun verläßt man den Editor mit 'X' und befindet sich wieder im Spiel. Man läßt die Spielfigur ein Leben verlieren, geht über den Taster wieder in den Editor und gibt ein 'T !5' (ein Leben weniger). Nach wenigen Augenblicken gibt der Editor eine Adresse aus, wo sich der Wert in dieser Weise verändert hat. Diese Prozedur wird noch einige Male durchgezogen, natürlich mit dem gerade aktuellen Wert, irgendwann kristallisiert sich dann eine bestimmte Adresse heraus. In diese Adresse kann man mittels des 'M'-Befehls (Memory) gehen, das heißt, sie editieren und den Wert verändern. Meistens hat man schon nach wenigen Versuchen Glück. Aber Vorsicht, zuviel des Guten ist ungesund, wie man unschwer an Bild 4 erkennen kann.

Die so ermittelten Adressen lassen sich in einem 'Memory Counter' eintragen, sie wird dann jedesmal wiedergefun-



Inh.
P.Engels

Spitzen- Zubehör für AMIGA-PC's

128K RAM-Karte für XT-Karte: 248,00 DM

Kurze Karte mit 128K Ram. Somit haben Sie 640K DosRam und können z.B. Ventura, F&A und D-Base-IV anwenden !!!

XT-Multifunktions Karte 299,00 DM

Multi-IO-Karte für XT-Karte mit Ser., Par., Uhr & 128K Ram

EGA-Karte 640 * 480 289,00 DM

Modi: EGA 640*350 / 640*480 / HGC / CGA - 256K Ram

Dual-Video-Karte 138,00DM

Modi: CGA/HGC/MDA 100% AMIGA-XT kompatibel

Genius GM-6000 Maus (hochauflösend): 99,00 DM

30 MB XT-File-Card / 60ms 768,00 DM

50 MB XT-File-Card / 40ms 948,00 DM

Die File-Cards können für AMIGA & XT partitioniert werden.
Lieferung incl. ausführlicher Soft- & Hardware Installations-Anleitung

Fordern Sie unsere kostenlose Preisliste an

SKy-Ware, Postfach 1331, 5308 Rheinbach
Telefon: 02253/2667

Spiele für Amiga

Neu! GREAT COURTS Neu!

DIE Tennis-Simulation für alle Amiga!
74,80 DM

Wir bekommen täglich Neuheiten für den Amiga!
Fordern Sie doch einfach mal die neueste Preisliste an,
Sie werden überrascht sein!
Hier ein paar Beispiele:

TITEL	PREIS	TITEL	PREIS
Rock'n Roll	69,90	Gran Prix Circuit	74,80
Xenophobe	74,80	Hillsfar	74,80
SimCity (dtsh)	89,90	King Arthur	74,80
Batman	74,80	Laser Squad	59,90
Chinese Karate	59,90	LittleComp.People	34,80
Create-A-Shape	98,00	Paper Boy	59,90
FS II-Hawaii Od.	44,80	Space Quest III	99,00

COMPY/SHOP

Gneisenastr. 29
4330 Mülheim/Ruhr
02 08-49 71 69 + 49 61 78

Haben Sie Fragen zum Thema AMIGA und AMIGA DOS?

Dann rufen Sie doch einfach mal an!

Achtung! Ab 1. Januar 1990 gelten die in den Klammern angegebenen Telefonnummern!

Das AMIGA-DOS-Team freut sich darauf, Ihnen bei Fragen zum AMIGA und zu diesem Magazin Hilfestellung zu leisten. In diesem Sinne steht Ihnen das AMIGA-DOS-Team an jedem **Dienstag** in der Zeit von **17:00 bis 20:00 Uhr** zur Verfügung, um Ihre Fragen zu beantworten. Wählen Sie einfach eine der nachfolgenden Nummern, um direkt den betreffenden Redakteur zu erreichen:

Jürgen Borngießer:
05651/8009-14
(05651/809-742)

Bernhard Rinke:
05651/8009-15
(05651/809-743)

Heinrich Stiller:
05651/8009-13
(05651/809-741)

Markus Matejka:
05651/8009-12
(05651/809-740)

Wir freuen uns auf Ihren Anruf!
Ihr AMIGA-DOS-TEAM





Abb. 1: Das 'Pro Access'-Modul am Amiga 500; zwar etwas unscheinbar, aber mit enormen Fähigkeiten

den ohne daß eine erneute Sucherei nötig wäre.

'Pro Access' hält aber eine weitere Methode bereit, das Spiel einfacher zu machen. Der 'TFD'-Befehl kann die ermittelte Adresse mit dem dazugehörenden Zählerbefehl eliminieren, so daß man quasi unsterblich (natürlich nur im Spiel) wird. Und schon ist man gerade bei schwierigen Spielen fein raus!

In den Tiefen eines Programmspeichers – nicht nur für Profis

Der Trainer ist allerdings nicht alles, was 'Pro Access' zu bieten hat. Gehen wir die einzelnen Punkte doch einmal durch:

Das Modul enthält eine eigene Formaterroutine für Disketten, allerdings auch nur in

einem Spezialformat. Auf diesen Disketten kann ein 'gefreesztes' (eingefrorenes) Programm abgespeichert und wieder geladen werden. Die Änderungen, die im Speicher vorgenommen wurden, werden dabei berücksichtigt, so daß man mühelos eine geänderte Version des Originalprogramms auf Datenträger bringen kann. Nach dem Laden geht das Programm dort weiter, wo der Freezer 'zugeschlagen' hat.

Bilder und Sounds aus Programmen können ebenfalls isoliert und auf Datenträger gebracht werden, die Bilder lassen sich zu Diashows verbinden oder auf den Drucker von üblichen Druckprogrammen bringen.

Programme oder der Speicher lassen sich analysieren und auch auf eventuell vorhandene Viren untersuchen, ein Vorteil, der gerade dem viren-

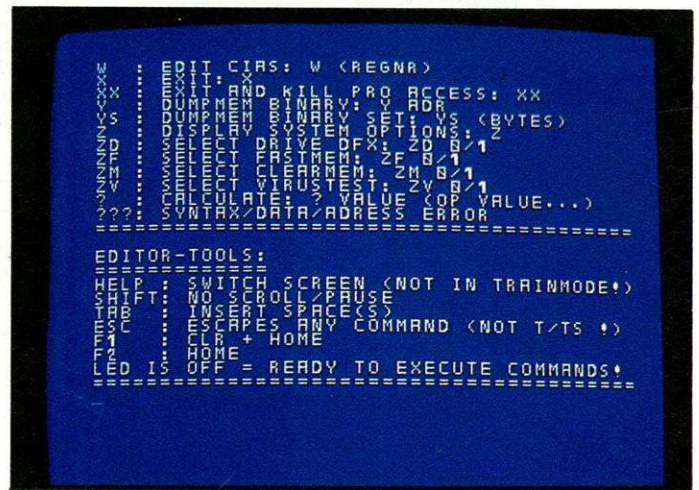


Abb.2: Der Editor des Systems meldet sich nach Tastendruck. Hier eine Auswahl der enthaltenen Befehle

geplagten Amiganer aufhören lassen sollte. Speichererweiterungen und Laufwerke werden automatisch erkannt, die Erweiterungen lassen sich vom Modul aus abschalten.

Ein interner Spriteeditor ermöglicht das einfache Erstellen von Sprites.

Dies sind die Hauptpunkte, die man über das Modul sagen kann, jedoch hat man immer noch nicht alles über 'Pro Access' erfahren. Allein die Befehle des Editors sind da schon ein Kapitel für sich.

Die Speicherinhalte lassen sich als Zahlenlisten oder direkt als Zeichencode ausgeben. Eine Suche nach bestimmten Zeichenfolgen ist ohne Mühe durchführbar, der Suchstring kann dazu auf verschiedene Weise eingegeben werden. Speicherbereiche können miteinander verglichen werden und binär

oder hexadezimal auf dem Bildschirm dargestellt werden. Für die Profis existiert der 'U'-Befehl, mit weiteren Parametern können freier Speicher, die Task-Liste, die Interrupt-Liste, die Library-Liste, Device-, Resource-, und Port-Liste aufgezeigt werden. Eingriffe in die CIA-Register sind vom Modul ebenfalls möglich. Ein Assembler und ein Disassembler gehören ebenfalls zur Ausstattung von 'Pro Access'.

Das neue Diskettenformat und seine Vorzüge

'Pro Access' ist in der Lage, Disketten mit einem eigenen Format zu formatieren. Diese Disketten sind allerdings nur unter 'Pro Access' zu lesen und zu beschreiben. Was uns ziemlich überrascht hat, ist

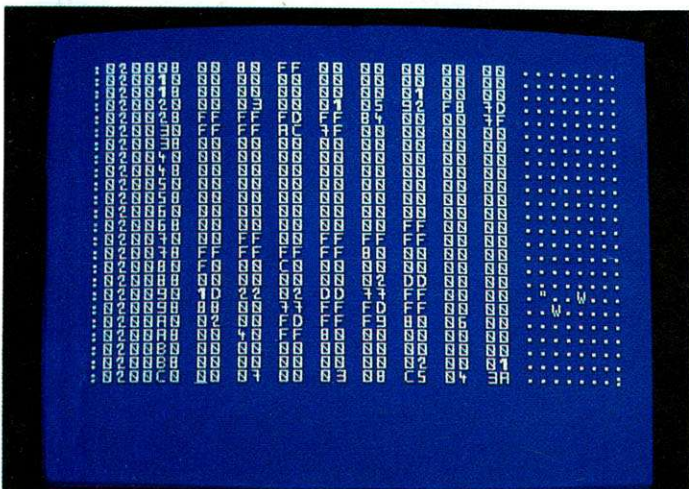


Abb.3: Während ein Spiel läuft, kann man es durch besagten Tastendruck 'einfrieren', sich in aller Ruhe den Speicher ansehen und nach Kollisionsabfragen, Zählern oder ähnlicher Unbill absuchen...

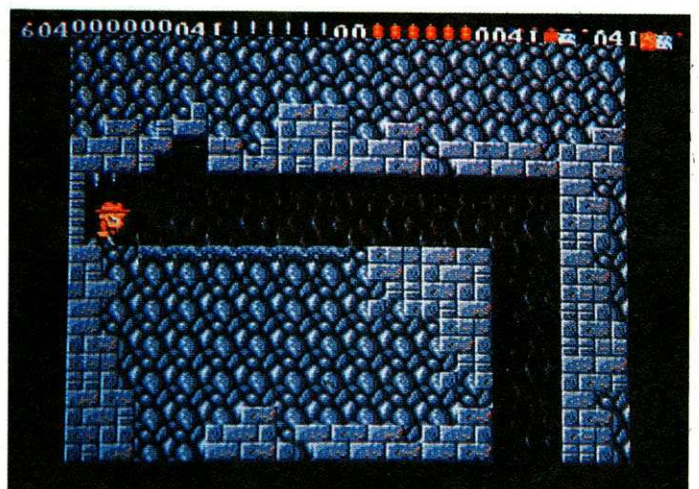


Abb.4: ...bis man etwas in dieser Richtung gefunden hat. Nun kann fröhlich verändert werden, falsche Werte führen allerdings zu seltsamen Bilderscheinungen wie hier bei 'Rick Dangerous'

die Geschwindigkeit des Formatierens, knappe 3 (DREI!) Sekunden brauchte das Modul, um die eingelegte Diskette für sein System vorzubereiten. Und damit ist schon einer der Vorteile für das eigene System klar: Die hohe Geschwindigkeit der Diskoperationen. Die weiteren positiven Punkte sind die hohe Kapazität der formatierten Diskette, laut Handbuch können Maximalspeicherblöcke von 940 kByte abgelegt werden, sowie eine hohe Datensicherheit. Kopien von Softwareprogrammen sind auch nur mit dem Modul lauffähig, da die Programme in der neuen Form abgespeichert werden.

Auf der beiliegenden Diskette befindet sich ein Konvertierungsprogramm, welches die auf den FDOS (so heißt das System) abgelegten Bilddateien auf normale DOS-formatierte Disketten überträgt. Diese Bilddateien können durch den Freezer den Programmen einzeln entnommen werden und auf den Disketten zu Diashows zusammenge-

setzt werden. Programmdateien können allerdings nicht kopiert werden, dagegen können Speicherbereiche gesamt auf FDOS-Disketten gebracht und auch überspielt werden.

'Pro Access' – fähiges Modul für neue und alte 'Speichergucker'

Das Modul hat uns Spaß gemacht, zu bemängeln gibt es eigentlich nichts. Jede Funktion wurde einwandfrei abgearbeitet, auch das Abspeichern und Laden von FDOS-Disketten. Einen Minuspunkt wäre vielleicht für das Gehäuse zu vergeben, beim A500 hatten wir ein bißchen Angst um den Expansionsport, weil das Gerät nicht hundertprozentig waagerecht auf dem Tisch stand. Ansonsten gibt es wirklich nichts zu bemängeln, auch das mitgelieferte Handbuch ist recht ausführlich.

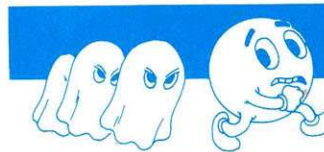
Das Gerät ist im Augenblick nur am A 500 und A 1000 ein-

setzbar, ob es eine Version für den A 2000 oder höher geben wird, stand bei Redaktionsschluß noch nicht fest. Fest steht nur eines, ein PROACCESS für den A 2000 müßte völlig anders aussehen, zu denken wäre an eine Steckkarte.

Der Preis ist mit 249,- DM nicht gerade niedrig, jedoch lohnt sich die Investition, denn Profis bekommen mit dem Modul ein Gerät zur Programmkontrolle in die Hand, das es in sich hat, und Einsteiger können sich mit 'Pro Access' prima in ihren Amiga einarbeiten.

Und als weitere Erkenntnis kann man sagen: Spiele werden mit 'Pro Access' noch interessanter.

(jb)



AMIGA DOS Blitzlicht

Name: Pro Access

Funktion: Stoppt und ermöglicht das Untersuchen von Programmen mit vielen Eingriffsmöglichkeiten

Anbieter: Boehm und Zanger Hard- u. Software, Heidenburgstr.13, Postfach 1163 Tel.:07822/8272

Preis: 249,- DM

Positiv:

- gesamter Speicher einsehbar
- Trainermodus auch für Einsteiger leicht verständlich
- ausführliches und verständliches Handbuch
- schnell anschließbar
- arbeitet mit jedem Programm (Ausnahmen sind uns nicht bekannt)
- abschaltbar ohne Entfernen
- integrierte Prozessor-Bremse

Negativ:

- Anschluß am Expansions-Port etwas wackelig
- relativ hoher Preis

***** hima computertechnik ***** WEIHNACHTSANGEBOTE

Laufwerke für Amiga

- | | |
|--|-----------------|
| 3 1/2"-Amigalaufwerk, extern, abschaltbar und mit durchgeführtem Bus, amigafarbenes Metallgehäuse | DM 219,- |
| 3 1/2"-Amigalaufwerk, intern, für Amiga 2000/500 | DM 179,- |
| 5 1/4"-Amigalaufwerk, extern, abschaltbar und mit durchgeführtem Bus, 40/80 Track, amigafarbenes Metallgehäuse | DM 269,- |

Speichererweiterungen

- 512-KB-Erweiterungen für den Amiga 500, intern,
- mit akkugepuffter Echtzeituhr
 - aufgebaut mit 1 MegaBit-Chips
 - bei Bedarf abschaltbar
- 1,2 oder 4 MB
- DM 199,-**

Händleranfragen und Angebote erwünscht
hima computertechnik Velbert
Burgstr. 13 · 5620 Velbert 1 · Telef: 0 20 51/5 67 67



Von DF0: nach DF3:

Amiga-Zusatzlaufwerke

– Was sie haben, was sie können...

Kurz nach Erwerb eines Amiga fällt dem Neubesitzer gleich eine Sache unangenehm auf: Der Betrieb mit einem Laufwerk kann ganz schön entnervend sein. Nicht nur, daß sich das Kopieren der Systemdiskette zu einem Problem entwickelt ('Please insert volume...'), auch das Arbeiten mit vielen kommerziellen Programmen gestaltet sich recht mühsam, wenn die Software auf mehreren Disketten verteilt wurde. Was macht also der gestreßte Ein-Laufwerk-Benutzer? Er schaut sich nach Laufwerk Nummer 2 um!

Zuerst einmal ist es wichtig, eine Unterscheidung zwischen den einzelnen Amigas zu machen: Während der A 2000 nur Einbau-Laufwerke benötigt, welche als einziges Extra nur für den Amiga selektiert wurden, benötigen A-500- oder A-1000-Besitzer externe, also nicht in der Zentraleinheit vorhandene Floppies. Beim Kauf kommt es deshalb weniger darauf an, welcher Laufwerkstyp nun der beste für den Amiga ist, sondern eher, wie gut es ausgestattet ist.

"Aber ein Laufwerk soll doch nur Disketten lesen und schreiben, was hat die Ausstattung damit zu tun?" – Eine Frage, die gerade von

Neulingen recht oft gestellt wird, wollen wir anhand unserer Marktübersicht an dieser Stelle beantworten. Zuvor jedoch noch ein paar Überlegungen zum 'idealen' Laufwerk:

1.) Die Floppy sollte sich abschalten lassen. Warum? Nun, jede Floppy belegt einen gewissen Speicherplatz im Amiga, der vom Hauptspeicher abgezweigt werden muß. Programme, die besonders viel Speicherplatz verbrauchen, reagieren beim Amiga 500 in der Grundausstattung mit 512 kByte auf ein zweites Laufwerk entweder mit der Meldung, dieses zu entfernen, oder (noch schlimmer) gar nicht! Ein Laufwerk am Port an- und abstecken ist nicht gerade vorteilhaft, weder für den Besitzer noch für den Portstecker. Ein abschaltbares Laufwerk hat da unübersehbare Vorteile: ein kleiner Daumendruck – und das Problem kann als gelöst angesehen werden.

2.) Ein durchgeschleifter Bus ist sinnvoll! Warum? Der Amiga kann von Haus aus bis zu vier Laufwerke gleichzeitig bedienen. Sehr schön, nur, was macht man, wenn das Zweitlaufwerk den Anschluß besetzt? Der durchgeschleifte Anschluß innerhalb einer Floppy läßt weitere Diskettengeräte zur Geltung kommen, also ein nicht zu unterschätzender Vorteil.

3.) 40/80-Track-Umschaltung – ein sinnvoller Zusatz für 5,25-Zoll-Laufwerke.

Warum? Der Amiga 2000 besitzt eine offene Systemarchitektur, das heißt, er kann unter anderem das Betriebssystem MS-DOS benutzen, ein



Abb.1: Die beiden Laufwerke der Firma Konyo, 'Senator 3.5' und 'Master-5A-1'



Abb.2: Das Alcomp 5,25-Zoll- und das HK 3,5-Zoll-Laufwerk



Abb.3: Die Firma Stalter bietet diese beiden Laufwerke an

Betriebssystem, welches vor allem auf Personal Computern (PC) Anwendung findet. Die Disketten werden dabei mit 360 kByte in 2 × 40 Tracks formatiert.

Gleichzeitig wollen wir aber die 5,25-Zoll-Floppy für unseren Amiga und 'sein' Betriebssystem nutzen, das heißt, wir brauchen die 5,25-Zoll-Disketten im 880 kByte-Format mit 2 × 80 Tracks. Was also ist leichter, als zwischen diesen beiden Betriebsmodi mit einem Schalter umzuschalten?

Dies waren drei der wichtigsten Extras, über die man beim Kauf eines Laufwerkes nachdenken sollte.

Um Ihnen einen kleinen Überblick über das große Angebot an Laufwerken zu geben, haben wir die Produkte einiger Firmen stellvertretend für die große Masse einmal unter die Lupe genommen.

'Senator' und 'Master' – solide Technik

Die beiden ersten Kandidaten kommen von der Firma Konyo. Es handelt sich dabei um das 5,25-Zoll-Laufwerk 'MASTER-5A-1' und das 3,5-Zoll-Laufwerk 'Senator'. Beide Laufwerke sind in soliden Metallgehäusen aufgebaut, ebenso die Selektierungselektronik. Beim 3,5-Zoll-Gerät handelt es sich um ein Slim-Line-Laufwerk mit einer Bauhöhe von einem Zoll (25,4 mm). Der 'Senator' besitzt einen durchgeschleiften Laufwerksbus, somit können weitere Geräte angeschlossen werden. Das Anschlußkabel

zum Amiga ist mit 70 cm zwar nicht allzu lang, aber in den meisten Fällen ausreichend. Die Floppy besitzt zwar einen Selektionsschalter an der Rückseite, muß jedoch immer als erstes externes Laufwerk an den Floppy-Port des Amiga angeschlossen werden. Weitere Geräte können dann an den Laufwerksanschluß der Floppy angeschlossen werden.

'Master-5A-1' ist ein 5,25-Zoll-Laufwerk mit Knebelverschluss. Auch dieses Laufwerk sitzt in einem Metallgehäuse mit Lüftungsschlitzen. An der Rückseite befindet sich ein weiterer Laufwerks-Port zum Anschluß weiterer externer Floppies. Ein 40/80-Track-Umschalter ermöglicht das wahlweise Arbeiten unter Amiga-DOS oder, in Verbindung mit einer PC-Karte, unter MS-DOS. Das Format der Disketten liegt beim Amiga-Betrieb bei 880 kByte, beim PC-Karten-Betrieb bei 360 kByte. Zusätzlich zum Laufwerk wird ein Netzgerät mitgeliefert, welches das Gerät mit den benötigten Spannungen und Strömen versorgt.

Beide Laufwerke arbeiten sehr geräuscharm, wobei das 5,25-Zoll-Gerät etwas mehr an Arbeitsgeräuschen erzeugt als das 3,5 Zoll. Die Gehäusefarbe ist dem Amiga angepaßt, die Geräte selbst sind sehr solide aufgebaut.

Stalter MegaTronic – platzsparend und leise

Die MegaTronic-Laufwerke von der Firma Stalter ent-



Abb.4: Die Rückseite des Alcomp-Laufwerkes bietet viele Auswahlmöglichkeiten

puppten sich als leiseste Vertreter ihrer Art. Die 3,5-Zoll-Floppy ist außerdem ein in den Abmessungen sehr kleines Gerät, ganze 18,4 × 10,5 × 3,5 cm (T/B/H) Platz beansprucht sie neben dem Amiga. An der Rückseite befinden sich ein Ein-/Ausschalter und ein Anschluß für weitere Laufwerke. Die niedrige Bauhöhe findet ihre Erklärung darin, daß es sich hierbei ebenfalls um ein Slim-Line-Laufwerk handelt. Das Anschlußkabel hat ebenfalls eine Länge von zirka 70 cm.

Die 5,25-Zoll-Floppy verfügt im Gegensatz zum Konyo-Laufwerk nicht über ein externes Netzteil, bezieht also den benötigten Betriebsstrom vom Amiga. Bei unserem Test an einem A 2000 traten zwar keine Probleme auf, diese entstehen jedoch spätestens dann, wenn die Floppy als DF2: an einem A 500 arbeiten soll, da dessen Netzteil nicht gerade eines der stärksten ist. An der Rückseite befindet sich ein 40/80-Track-Umschalter sowie ein Laufwerksbusanschluß. Beide Laufwerke sind in stabilen Metallgehäusen eingebaut und machen ebenfalls einen guten Eindruck. Optional sind die Stalter-Geräte auch mit Trackdisplays zur Anzeige der aktuellen Trackposition zu erhalten.

Alcomps Alleskönner und HKs Zweitläufer

Die Alcomp-Laufwerke unterscheiden sich eigentlich kaum von denen anderer Hersteller. Das 3,5-Zoll-Gerät

verfügt ebenfalls über einen Ein-/Ausschalter und einen durchgeschleiften Bus. Ein zweites, etwas teureres Laufwerk verfügt über eine Trackanzeige und einen Schreibschutz-Schalter.

Die 5,25-Zoll-Floppy besitzt einen 40/80-Track-Umschalter, einen Ein-/Ausschalter, einen durchgeschleiften Laufwerksbus und eine automatische Diskchange-Erkennung, die allerdings erst nach Herausnehmen oder Einlegen der Diskette reagiert. Beide Laufwerke sind ebenfalls in Metallgehäusen eingebaut und sehr lauffähig. Beim 5,25-Zoll-Modell macht sich die Absenkung des Schreib-/Lesekopfs allerdings durch relativ lautes Klacken bemerkbar. Trotzdem sind beide Geräte im Alltagsgebrauch ohne Probleme zu bedienen.

Von der Firma HK-Computer hatten wir das 3,5-Zoll-Gerät zu Besuch. Es verfügt ebenfalls über eine Busdurchführung und einen Ein-/Ausschalter. Auch seine Arbeitsweise ist mit der Bewertung 'sehr leise' zu charakterisieren.

So, das war sie also, unsere kleine Reise durch Laufwerksland. Leider können wir nicht alle angebotenen Laufwerke vorstellen, doch da die Unterschiede weniger in den Leistungsdaten als bei den Extras auftreten, sollte es Ihnen leichtfallen, sich ein Bild über das Angebot zu machen. Als kleine Hilfe finden Sie im Anschluß eine Marktübersicht mit Preisen, in der auch die vorgestellten Laufwerke noch einmal zu finden sind.

(jb)

Marktübersicht

ANBIETER	TYP	PREIS	BUSDURCHFÜHRUNG	EIN-/AUSSCHALTER
ALCOMP	3,5" 3,5" Profi 5,25" 3,5" + 5,25"	249,- 329,- 308,- 558,-	JA JA JA siehe 3,5 – 5,25	JA JA JA JA
ATLANTIS	3,5" (NEC) 3,5" 5,25	248,- 298,- 298,-	JA JA JA	JA JA JA
COMBITEC	3,5" 5,25"	278,- 368,-	JA JA	JA JA
COMPTEC	3,5" 5,25"	257,- 325,-	JA JA	JA JA
CPS	3,5" 5,25" 3,5" 5,25"	290,- 370,- 319,- 408,-	JA JA JA JA	JA JA JA JA
DATARAM	CEW 3,5" CEW 5,25" GOLEM 3,5 GOLEM 5,25	239,- 289,- 289,- 389,-	JA JA JA JA	JA JA JA JA
EUROSYSTEMS	3,5" 5,25"	265,- 299,-	JA JA	JA JA
FSE	3,5" 5,25"	229,- 259,-	JA (+ 10,-) JA (+ 10,-)	JA JA
GNE	3,5" 5,25"	239,- 289,-	JA JA	JA JA
HK – Computer	3,5" 5,25"	229,- 279,-	JA JA	JA JA
IDS	3,5" 5,25"	218,- 249,-	JA JA	JA JA
KUPKE	3,5" 3,5" 5,25" 5,25"	299,- 319,- 399,- 419,-	JA JA JA JA	JA JA JA JA
OTRONIC	3,5" 5,25" Trackdispl.	218,- 361,- 108,-	JA JA - - - - -	JA JA - - - - -
STALTER	3,5" 3,5" DIGI 5,25" 5,25" DIGI	222,- 259,- 255,- 309,-	JA JA JA JA	JA JA JA JA
VESALIA	3,5" Winner 3,5 Winner 5,25	189,- 229,- 269,-	JA JA JA	JA JA JA

Einige Adressen von Anbietern:

(erhebt keinen Anspruch auf Vollständigkeit)

ALCOMP GmbH
Glescher Weg 22
D-5012 Bedburg
Tel.: 0 22 72/20 93

GNE Grebe-Neumann-
Elektronik
Am Stein 10
D-5419 Raubach
Tel.: 0 26 84/5 56 6

COMBITEC Computer GmbH
Liegnitzer Str. 6-6a
D-5810 Witten/Ruhr
Tel.: 0 23 02/88 07 2

KUPKE Computertechnik
Burgweg 52a
D-4600 Dortmund 1
Tel.: 0 23 1/8 18 32 5-27

CPS Computertechnik GmbH
Hamburger Str. 283
3300 Braunschweig
Tel.: 0 53 1/3 30 66 3

ATLANTIS
Postfach 1141
D-5030 Hürth
Tel.: 0 22 33/41 08 1

DATARAM Computertechnik
Kampstr. 122
D-4370 Marl
Tel.: 0 23 65/69 14 31

COMPTEK-Versand
Obermörnterer Str. 1
D-4192 Kalkar 4
Tel.: 0 28 24/3 86 7

Das Diagramm zeigt eine Draufsicht auf ein Diskettenlaufwerk. Ein Disketten-Medium ist in der Laufwerkschachtel eingelegt. Ein Schreib-/Lesekopf ist über dem Medium positioniert und wird von einem Antrieb (Motor) angetrieben. Die Beschriftungen im Diagramm sind:

- SCHREIB-/LESEKOPF
- ANTRIEB
- DISKETTE

Oliver Heggelbach

Abtipphilfe

HexType V1.0

– Schnelle Eingabe von Listings

Dieses Programm dient zur Eingabe von Programm- oder Grafikdateien, die als Hexdumps (die beliebten Zahlenkolonnen) abgedruckt wurden.

In unserem Heft werden Sie öfters auf sogenannte HEXDUMP-Dateien stoßen. Dabei handelt es sich in den meisten Fällen um Programme in Assembler, C oder Modula2, deren Quellcode zu umfangreich für einen Abdruck wäre. Das vorliegende Programm macht aus diesen HEXDUMPS wieder vollständige Dateien, die je nach Beschaffenheit sofort lauffähig oder weiterverwendbar sind. Nach dem Starten des Programms unter AmigaBASIC stehen folgende Menüpunkte zur Verfügung:

1.) Neues Listing

Bevor Sie mit der Eingabe eines HexType-Listings beginnen

können, müssen Sie diesen Menüpunkt anwählen. Nach der Eingabe des Dateinamens, der Anzahl der Zeilen und der Länge der Datei (diese Angaben finden Sie in den ersten drei Zeilen des Hexdumps), gelangt man in den Eingabemodus.

2.) Listing speichern

Wenn Sie das Abtippen eines Listings unterbrechen wollen, können Sie mit diesem Menüpunkt Ihre bisherigen Eingaben auf Diskette (oder sonstwo) speichern. Als Dateiname wird der Name des Listings und die Endung ".l" verwendet.

3.) Listing laden

Laden eines unter Punkt 2 abgespeicherten Listings. Dabei

muß der Name des Listings angegeben werden.

4.) Datei erstellen

Aus einem vollständig abgetippten Listing wird die endgültige Datei erzeugt und abgespeichert. Dies kann (aufgrund des "schnellen" AmigaBASIC) eine Weile dauern. Nach Beendigung dieses Vorgangs wird die Länge der gespeicherten Datei ausgegeben.

5.) Programm beenden

Ein Punkt, der wohl keiner näheren Erläuterung bedarf.

Der Eingabemodus

Im Eingabemodus zeigt der Computer am linken Bildschirmrand die aktuelle Zeilennummer an (zum Beispiel "000:"), daneben erscheint ein Cursor. Nach der Eingabe der entsprechenden Zeile des Listings (**natürlich ohne Zeilennummer**) wird diese vom Programm überprüft. Ist die eingegebene Zeile zu kurz oder liegt ein Prüfsummenfehler vor, muß die Zeile noch einmal eingegeben werden. Nachdem das Listing vollständig eingegeben wurde, erscheint die Meldung

"Listing vollständig" und man kann nun die Datei mit "Datei erstellen" abspeichern.

Noch ein Hinweis: Die etwas merkwürdig anmutenden Punkte ("."), die in den HEXDUMP-Listings vorkommen, sind keine Druckfehler und müssen genauso abgetippt werden. Sie dienen zum Komprimieren von Daten, dadurch können die Listings (insbesondere bei Grafikdaten) teilweise erheblich verkürzt werden. Den Umgang mit HEXTYPE können Sie an unserem Listing DOUBLESTRIKE gleich mal ausprobieren, wir wünschen dazu gutes Gelingen.

(jb)

Wichtig !!!

Die Zeilennummern am Anfang der Programmzeilen dienen nur zur besseren Übersicht und sollten im BASIC-Editor nicht mit eingegeben werden. Sie können zur Eingabe auch den Checksummen benutzen, der sich ebenfalls in diesem Heft befindet.

Listings

```
000:543 REM
001:784 REM HexType v1.0
002:543 REM
003:052 REM Autor: Oliver Heggelbacher
004:543 REM
005:000
006:590 CLEAR,80000&
007:416 DEFINIT a-z
008:000
009:977 REM Bytes pro Zeile
010:828 bmax=16
011:000
012:984 DIM hd$(999):DIM SHARED bg,nlin
013:362 bg=0:fi=0:nlin=0
014:377 cu$=CHR$(32)+CHR$(8)
015:052 DEF FN dstell$(n$)=STRING$(3-LEN(n$),"0")+n$
016:000
017:714 WINDOW 2,"HexType v1.0"
018:822 MENU 1,0,1,"Projekt"
019:736 MENU 1,1,1,"Neues Listing"
020:070 MENU 1,2,1,"Listing speichern"
021:330 MENU 1,3,1,"Listing laden"
022:290 MENU 1,4,1,"Datei erstellen"
023:698 MENU 1,5,1,"Programm beenden"
024:174 MENU 2,0,1,""
025:302 MENU 3,0,1,""
026:430 MENU 4,0,1,""
027:382 ON MENU GOSUB Projekt
028:768 MENU ON
029:000
030:012 Warte:
031:204 WHILE NOT bg:SLEEP:WEND
032:852 WHILE fi:SLEEP:WEND
033:000
034:116 Eingabe:
035:871 PRINT FN dstell$(MID$(STR$(ze),2));";";
036:522 nlin=0:t1=0:k$="":ip$=""
037:274 Eingabe2:
038:568 WHILE k$<>CHR$(13) AND NOT nlin
039:676 COLOR 0,3:PRINT cu$;COLOR 1,0
040:430 SLEEP
041:495 PRINT cu$;
042:386 k$=INKEY$
043:872 IF k$=CHR$(8) AND t1>0 THEN
044:057 t1=t1-1:ip$=LEFT$(ip$,t1)
045:039 PRINT CHR$(8);cu$;
046:216 END IF
047:320 IF k$=CHR$(127) AND k$>CHR$(31) AND t1<bmax*3+3 TH
EN
048:982 t1=t1+1:ip$=ip$+k$
```

```
049:175 PRINT k$;
050:216 END IF
051:196 WEND
052:468 PRINT
053:741 IF nlin THEN Eingabe
054:448 IF LEN(ip$)<bmax*3+3 THEN
055:194 PRINT "Eingabefehler"
056:797 GOTO Eingabe
057:216 END IF
058:000
059:232 REM Checksumme ueberpruefen
060:258 ch=ze:hd$(ze)=" "
061:515 FOR n=0 TO bmax-1
062:653 w$=MID$(ip$,n*3+1,2)
063:110 hd$(ze)=hd$(ze)+w$
064:501 IF w$=".." THEN w=256 ELSE w=VAL("&h"+w$)
065:471 ch=(ch*2+w) MOD 4095
066:398 NEXT n
067:712 IF ch<>VAL("&h"+MID$(ip$,bmax*3+1)) THEN
068:158 PRINT "Eingabefehler":BEEP
069:797 GOTO Eingabe
070:216 END IF
071:829 ze=ze+1:IF ze<maxz THEN Eingabe
072:381 PRINT "Listing vollständig":fi=-1:GOTO Warte
073:000
074:543 REM
075:815 REM Menuesteuerung
076:278 Projekt:
077:579 mx=MENU(0):my=MENU(1)
078:510 IF mx<>1 THEN RETURN
079:421 ON my GOTO NListing,Speichern,Laden,Datei,Ende
080:982 RETURN
081:000
082:464 NListing:
083:654 CALL Sicher("Neues Listing",1,n):IF n=1 THEN RETURN
084:468 PRINT
085:146 INPUT "Name : ";lnam$
086:030 INPUT "Zeilen: ";maxz
087:456 INPUT "Laenge: ";dlen
088:641 PRINT STRING$(3*bmax+7,"-")
089:487 ze=0:bg=-1
090:982 RETURN
091:000
092:870 Speichern:
093:437 CALL Sicher("Listing speichern",0,n)
094:472 IF NOT bg THEN
095:094 PRINT "Kein Listing im Speicher":BEEP
096:982 RETURN
097:216 END IF
098:147 OPEN lnam$+".l" FOR OUTPUT AS #1
099:122 PRINT #1,lnam$
100:558 PRINT #1,maxz
101:480 PRINT #1,dlen
```



```

102:353 PRINT #1,ze
103:317 PRINT #1,fi
104:175 FOR n=0 TO ze-1
105:701 PRINT #1,hd$(n)
106:398 NEXT n
107:043 CLOSE #1
108:530 PRINT ze;"Zeilen gespeichert"
109:982 RETURN
110:000
111:506 Laden:
112:950 CALL Sicher("Listing laden",1,n):IF n THEN RETURN
113:640 INPUT "Name des Listings : ",n$
114:099 OPEN n$+".1" FOR INPUT AS #1
115:010 INPUT #1,lnam$
116:502 INPUT #1,maxz
117:424 INPUT #1,dlen
118:089 INPUT #1,ze
119:053 INPUT #1,fi
120:175 FOR n=0 TO ze-1
121:477 INPUT #1,hd$(n)
122:398 NEXT n
123:043 CLOSE #1
124:641 PRINT STRING$(3*bmax+7,"-")
125:007 bg=-1
126:356 IF fi THEN PRINT "Listing vollständig"
127:982 RETURN
128:000
129:368 Datei:
130:061 CALL Sicher("Datei erstellen",0,n)
131:632 IF NOT fi THEN
132:934 PRINT "Listing nicht vollständig":BEEP
133:982 RETURN
134:216 END IF
135:467 OPEN lnam$ FOR OUTPUT AS #1
136:626 ze=0:sp=0:ln=0
137:952 WHILE ln<dlen

```

```

138:475 GOSUB GetW
139:992 IF w$=".." THEN
140:221 GOSUB GetW:by=w
141:475 GOSUB GetW
142:049 PRINT #1,STRING$(w,CHR$(by));
143:429 ln=ln+w
144:091 ELSE
145:159 PRINT #1,CHR$(w);:ln=ln+1
146:216 END IF
147:196 WEND
148:043 CLOSE #1
149:202 PRINT ln;"Bytes gespeichert"
150:982 RETURN
151:000
152:256 GetW:
153:341 w$=MID$(hd$(ze),sp*2+1,2):w=VAL("&h"+w$)
154:967 sp=sp+1
155:619 IF sp=bmax THEN sp=0:ze=ze+1
156:982 RETURN
157:000
158:196 Ende:
159:726 CALL Sicher("Programm beenden",1,n):IF n THEN RETURN
160:590 MENU OFF
161:464 WINDOW CLOSE 2
162:674 MENU RESET
163:500 END
164:000
165:021 SUB Sicher(te$,si,n) STATIC
166:599 nlin=-1
167:169 COLOR 3:PRINT CHR$(13);CHR$(13);te$:COLOR 1
168:136 IF si AND bg THEN
169:908 INPUT "vorhandenes Listing wird geloescht - Sich
er (j/n) ";n$
170:275 IF n$="j" THEN n=0 ELSE n=1
171:216 END IF
172:568 END SUB

```

Oliver Heggelbacher

CheckBAS V1.0

Der AmigaBASIC-Checksummer

In der AMIGA DOS finden Sie eine Menge an BASIC-Listings vornehmlich in Amiga-BASIC geschrieben. Um Ihnen die Eingabe etwas zu erleichtern stellen wir Ihnen 'CheckBAS V1.0 vor. Dieses Programm dient dazu, Tippfehler beim Abtippen von BASIC-Listings aufzudecken und so die Fehlersuche zu vereinfachen.

AmigaBASIC-Listings haben den Vorteil, sich recht gut in einer Zeitschrift abdrucken zu lassen. Leider hat der BASIC-Editor den Nachteil, daß man kaum eine Überprüfung der eingegebenen Zeile vornehmen lassen kann, es sei denn, man wechselt den Blick bei jedem Buchstaben zwischen Monitor und AmigaDOS. Zur besseren Eingabe geben wir Ihnen ein Hilfsmittel in die Hand, daß Ihnen zumindest die Arbeit erleichtert: Den AmigaBASIC-Checksummer.

Anwendung des Programms

Nachdem man ein BASIC-Listing ganz normal (das heißt ohne die zwei sechsstelligen Zahlen am Anfang einer Zeile) unter AmigaBASIC eingegeben hat, speichert man es mit SAVE "Dateiname",a (als ASCII-Version) ab. Dann startet man

das 'CheckBAS'-Programm. Nach der Eingabe des Dateinamens werden jeweils 20 Zeilen der Datei angezeigt, danach wird auf einen Tastendruck gewartet. Bei der Ausgabe der Zeilen wird zuerst die Zeilennummer angezeigt, danach folgt die Prüfsumme und der Inhalt der Zeile. (z.B. 001:790 PRINT "Hallo") Nun überprüfen Sie die angezeigte Prüfsumme mit der des Listings in der Zeitschrift. Sollten diese nicht übereinstimmen, so notieren Sie sich die entsprechende Zeile und korrigieren Sie die Zeile, nachdem Sie das gesamte Listing überprüft haben.

Hinweise

1. Um ein Programm zu überprüfen, können Sie auch folgendermaßen vorgehen: Nachdem Sie das zu überprüfende Programm abgespeichert haben, lassen Sie es im Speicher stehen, starten ein zweites AmigaBASIC (wozu hat man

denn Multitasking) und laden in diesem AmigaBASIC das CheckBAS-Programm. Wenn Sie nun das LIST-Window des ersten AmigaBASICs in den Vordergrund holen, können Sie beim Auftreten eines Prüfsummenfehlers die entsprechende Zeile sofort korrigieren.

2. Wollen Sie, daß weniger als 20 Zeilen auf einmal angezeigt werden, so können Sie im

CheckBAS-Programm die Variable "zanz" mit einem anderen Wert belegen. Die entsprechende Stelle ist mit einer REM-Zeile markiert.

CheckBAS ist eingeschränkt auch bei anderen Listings nutzbar, wenn diese als ASCII-Files abgespeichert sind. Hier kommt es aufs Probieren an.

(jb)

Listings

```

000:543 REM
001:808 REM CheckBAS v1.0
002:543 REM
003:052 REM Autor: Oliver Heggelbacher
004:543 REM
005:000
006:416 DEFINT a-z
007:837 DEF FN dste11$(x)=STRING$(4-LEN(STR$(x)),48)+MID$(STR$(x),2)

008:000
009:008 REM Anzahl Zeilen
010:200 zanz=20
011:000
012:609 COLOR 3:PRINT "CheckBAS v1.0":COLOR 1
013:746 PRINT:INPUT "Name der zu überprüfenden Datei : ",qu$
014:468 PRINT
015:000
016:090 ln=0
017:971 OPEN qu$ FOR INPUT AS #1
018:059 WHILE NOT EOF (1)
019:168 LINE INPUT #1,ip$
020:522 az=0:ch=0
021:961 FOR n=1 TO LEN(ip$)
022:257 n$=MID$(ip$,n,1):IF NOT az THEN n$=UCASE$(n$)
023:260 IF n$=CHR$(34) THEN az=NOT az
024:552 IF az OR n$<>CHR$(32) THEN ch=(ch*2+ASC(n$)) MOD 1000
025:398 NEXT n
026:229 PRINT FNdste11$(ln);":";FNdste11$(ch);CHR$(32);
027:689 COLOR 3:PRINT ip$:COLOR 1
028:472 ln=(ln+1) MOD 1000
029:392 IF (ln MOD zanz)=0 THEN
030:371 PRINT STRING$(20,45);"Taste drücken";STRING$(20,45)
031:756 WHILE INKEY$="":SLEEP:WEND
032:216 END IF
033:196 WEND
034:043 CLOSE #1
035:259 PRINT STRING$(21,45);"Listingende";STRING$(21,45)

```

Listing CheckBAS

Oliver Heggelbacher

Bis der Joystick glüht...

DoubleStrike – Rettet das Universum!!

Irgendwie ist das schon komisch – immer wenn irgendwelche Außerirdischen schlecht zu Mittag gegessen, Streit mit der Frau oder einfach nur miese Laune haben, kommt Ihnen der Gedanke, mal wieder das alte Mütterchen Erde und seine befreundeten Nachbarwelten zu überfallen und sämtliche Bewohner zu Stiefelputzern zu machen. Mit 'DoubleStrike' ist es nicht anders, hier gibt es alles das, was eines geübten Joystickbenutzers Herz höher schlagen läßt, inklusive Extrawaffen und Nervenkitzel.

Das Programm ist ein "Ballerspiel" im Stil von Space Harrier, mit dem Unterschied, daß hier bis zu zwei Spieler gleichzeitig zu Werke gehen und mit Hilfe diverser Extrawaffen die gegnerischen Formationen beseitigen können.

Vor dem Nervenkitzel kommt erst der Schweiß der Arbeit: Hinweise zum Abtippen

1) Die Listings 'DoubleStrike' (Listing 1) und 'DSGraph' (Listing 2) sollten Sie mit Hilfe des HexType-Eingabeprogramms (zu finden im 'Tips&Tricks'-Teil) abtippen.
2) Danach müssen Sie das Listing "DScalc" unter Amiga-BASIC abtippen und starten. Das Programm lädt nun die Datei "DSgraph" und erzeugt in der RAM-Disk ("RAM:")

die etwa 81 kByte große Datei "DSdata", was zirka 14 Minuten in Anspruch nimmt (Kaffeepause!). Diese müssen Sie nun in das Verzeichnis kopieren, in dem sich die "DoubleStrike"-Datei befindet.

3) Um das Spiel zu laden, muß vom CLI aus in das Verzeichnis gewechselt werden, in dem sich die Dateien "DoubleStrike" und "DSdata" befinden. Durch Aufruf von "DoubleStrike" wird das Spiel dann gestartet.

Zuerst wird die Anzahl der Spieler angezeigt und kann mit Joystick 1 verändert werden (links/rechts). Bei zwei Spielern muß ein zweiter Joystick vorhanden sein, das Spiel kann nicht mit Maus gespielt werden. Am oberen Bildschirmrand wird die Punktezahl, die Anzahl der Raumschiffe ("S:") und die Anzahl der Energiebomben ("B:") angezeigt. Diese Anzeige ist für Spieler 1 und 2 vorhanden ("PL1:" bzw. "PL2:").

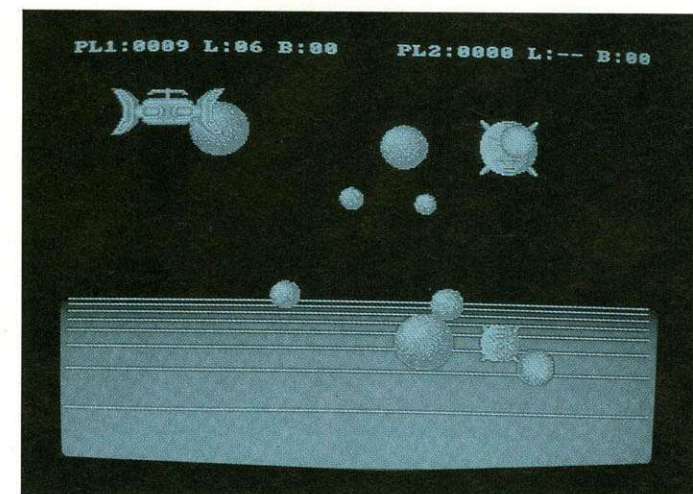


Abb.1: Ganz schön hektisch geht's im Weltraum zu, nicht nur sonderbare Kugelwesen wollen einem an den Kragen,...

Beim Spiel geht es um folgendes: Sie sollen fünf Planeten von merkwürdigen Invasoren befreien (mehr ist zur Hintergrundstory dieses Spiels nicht zu sagen). Die Feinde, die sich Ihnen dabei in den Weg stellen, müssen teilweise mehrmals getroffen werden, um vernichtet zu werden. Andere Objekte (z.B. Felsbrocken) können überhaupt nicht abgeschossen werden.

Auf einem Planeten begegnen Sie außerdem mindestens einem Mutterschiff, welches Sie mit mehreren Schüssen zerstören müssen. Haben Sie einen Planeten durchflogen, so müssen Sie auf dem Weg zum nächsten Planeten zahlreichen Felsbrocken, die Ihnen entgegenkommen, ausweichen. Nach dem Verlust eines Raumschiffs kann Ihr neuer Raumer für kurze Zeit nicht zerstört werden.

Von Zeit zu Zeit fliegen Kapseln mit einem "E" auf Sie zu. Diese verschaffen Ihnen nach dem Einsammeln bessere Waffen. Der Typ des Extras wird, solange die Kapsel auf Sie zukommt, am unteren Bildschirmrand angezeigt. Es gibt folgende Extrawaffen:

1) Power

Der Schuß verschwindet nach einem Treffer nicht.

2) Speed

ergibt eine höhere Schußgeschwindigkeit.

3) Shadow

Hiermit können Sie nun Schüsse von der Größe Ihres Raumschiffes abfeuern (kann allerdings nicht mit 'Speed' kombiniert werden).

4) Balls

kennzeichnet zwei Kugeln, die sich ober- und unterhalb

Ihres Schiffes befinden und nach dem Abfeuern zum Schiff zurückkehren.

5) Side

bringt zusätzliche Schüsse nach oben und unten.

6) Satel

erzeugt einen Satelliten, der sich unabhängig vom Raumschiff bewegt und Schüsse abfeuert. Der Satellit kann auch durch Berührung der Feinde vernichtet werden.

Von den Extras 4), 5) und 6) kann immer nur eines aktiv sein.

7) Bombs

Diese Extrawaffe verhilft Ihnen zu zwei Energiebomben, die Sie auslösen, indem Sie den Feuerknopf etwa eine Sekunde gedrückt halten.

Wenn's zu aufregend werden sollte: Eine kleine Pause tut gut!

Sollte Ihnen das Spiel zu hektisch werden, können Sie es mit der P-Taste unterbrechen. Durch Druck auf eine beliebige Taste wird das Spiel dann fortgesetzt.

So, das war's. Genug gesagt, laßt uns nun die Welten retten!

(jb)

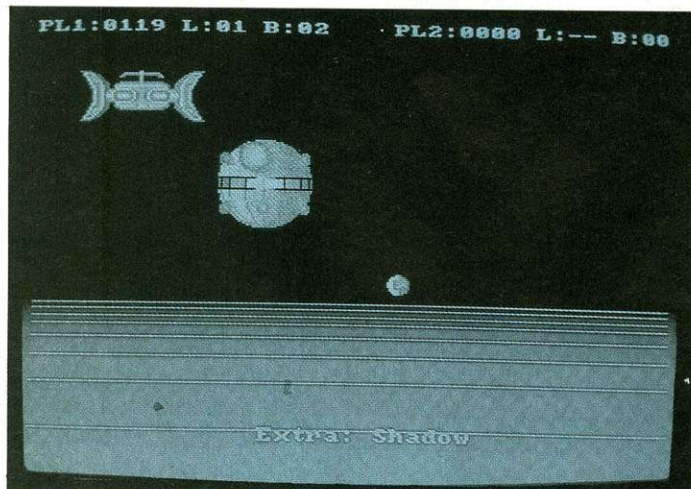


Abb.2: ...sondern auch Riesen-Raumschiffe und ähnlich garstige Sonderformen des (Amiga-) Lebens

Wichtig!!!

Beim BASIC-Programm sollten die Zeilennummern nicht mit eingetippt werden, da diese nur die Übersicht im Listing verbessern. Sie können zur Eingabe auch einen Texteditor und im Anschluß daran unseren Checksummer benutzen.

Listings

```

*****
*                               AMIGA DOS INFO                               *
*                               *                               *               *
* Programmname   : DoubleStrike                                           *
* Funktion       : Hauptprogramm                                           *
* Sprache        : HEXDUMP                                                *
* Besonderheiten : Muß mit HEXTYPE eingegeben werden! *
*****
000:00 00 03 F3 .. 00 07 03 .. 00 07 02 00 00 05 37 3C3
001:00 00 02 FF 00 00 04 E2 00 00 03 E9 00 00 05 37 A53
002:49 F9 00 00 03 86 61 00 01 EC 61 20 61 00 14 18 568
003:29 6C 05 BE 05 C6 61 00 08 BA 29 7C 00 00 0B 37 5E1
004:05 A6 61 00 02 30 61 00 02 90 60 EA 2C 78 00 04 8C3
005:20 3C 00 01 48 20 22 3C 00 01 00 02 4E AE FF 3A A35
006:29 40 05 CE D0 BC 00 00 EB 90 29 40 05 D6 06 80 ACD
007:00 00 03 70 29 40 05 CA 43 F9 00 00 02 51 42 80 F35
008:4E AE FD D8 29 40 05 AE 2C 40 24 3C 00 00 03 ED 7E0
009:22 3C 00 00 02 5D 4E AE FF E2 29 40 05 B2 22 00 51D
010:24 2C 05 CE 26 3C 00 0F FF 4E AE FF D6 22 2C 219
011:05 B2 4E AE FF DC 22 4E 2C 78 00 04 4E AE FE 62 B73
012:70 1D 22 3C 00 00 FF FF 51 C9 FF FE 51 C8 FF F4 587
013:4E AE FF 7C 20 3C 00 00 F0 00 22 3C 00 01 00 02 89F
014:4E AE FF 3A 29 40 05 BE 06 80 00 00 78 00 29 40 FC5
015:05 C2 20 3C 00 00 08 00 22 3C 00 01 00 02 4E AE 601
016:FF 3A 29 40 05 B6 43 F9 00 00 02 40 42 80 4E AE 355
017:FD D8 29 40 05 AA 2C 40 4E AE FE 38 20 6C 05 B6 E58
018:61 00 02 88 30 FC 01 80 42 58 29 48 05 BA 61 00 B89
019:00 D4 4D F9 00 02 DF F0 00 3D 7C 03 E0 00 96 2D 6C 16C
020:05 B6 00 80 42 6E 00 88 3D 7C 30 81 00 8E 3D 7C 3C8
021:30 C1 00 90 3D 7C 00 38 00 92 3D 7C 00 00 94 E5C
022:3D 7C 32 00 01 00 42 6E 01 02 42 6E 01 04 42 6E 0D7
023:01 08 42 6E 01 0A 3D 7C 83 C0 00 96 4E 75 45 F9 94B
024:00 00 0B B2 70 00 10 39 00 00 01 C0 FC 00 18 59E
025:45 F2 00 00 47 EE 01 82 70 06 36 DA 51 C8 FF FC 26F
026:36 1A 38 03 42 82 20 6C 05 BA 43 F9 00 00 01 86 1F4
027:43 E9 01 FF 42 80 42 81 12 11 E2 09 D2 3C 00 B0 002
028:64 0E 4A 02 66 0A 30 FC FF E1 30 FC FF FE 74 01 460
029:E1 49 12 3C 00 E3 30 C1 30 FC FF FE 30 FC 01 80 462
030:30 C3 B8 43 66 0A 3A 1A 38 1A D6 45 43 E9 FF E0 08B
031:D0 3C 00 10 64 C0 20 BC FF FF FF 19 7C 00 FF 8A8
032:00 07 4E 75 20 6C 05 BA 20 BC FF FF FF FE 42 2C 2DC
033:00 07 4E 75 41 F9 00 00 01 86 70 00 32 3C 00 80 29F
034:34 3C 00 FF C4 C1 28 00 D8 41 84 C4 10 C2 52 40 7BC
035:B0 7C 02 00 65 EA 42 44 41 F9 .. 00 04 13 F9 00 404
036:00 00 4A 20 3C 00 00 00 C2 32 3C 00 0F 3A 01 B4 A17
037:41 66 04 36 19 34 19 30 81 B9 58 1A 00 CA 3C 00 F55
038:03 66 04 0A 44 0F FF D2 43 C2 7C 0F FF 51 C8 FF EB6
039:E0 4E 75 61 00 0F 5E 61 00 01 5C 61 00 00 E6 61 903
040:00 10 A6 61 00 0F 54 41 F9 00 00 00 7E 61 00 11 B8C
041:4E 61 00 04 00 61 00 00 CC 61 00 00 DC 30 2E 00 0E5
042:0C 08 00 00 09 67 10 13 FC 00 01 .. 00 04 13 FC 99C
043:00 31 00 00 00 CE 08 00 00 01 67 10 13 FC 00 02 E53
044:.. 00 04 13 FC 00 32 00 00 CE 4A 39 00 BF 0E 1E8
045:01 6B 9E 4E 75 61 00 05 88 29 7C 00 00 02 64 05 5FC
046:A6 42 2C 00 04 61 00 0E CA 61 00 00 E8 61 00 06 19E
047:F2 61 00 0D 5A 4B EC 00 1A 47 ED 00 46 61 00 07 08E
048:B4 26 4D 4B ED 00 46 61 00 07 AA 61 00 10 18 41 D65
049:F9 00 00 00 CF 61 00 10 C4 61 46 61 00 0E BA 61 27F
050:00 03 70 61 00 03 92 61 38 61 4A 61 0C 10 2C 00 A12
051:02 B0 3C 00 C0 65 AE 4E 75 10 39 00 BF EC 01 67 4C2
052:18 E2 18 46 00 B0 3C 00 19 66 0E 42 39 00 BF EC 047
053:01 10 39 00 BF EC 01 67 F2 42 39 00 BF EC 01 4E 040
054:75 24 2E 00 04 C4 BC 00 01 FF 00 B4 BC 00 01 30 1D6
055:00 66 EE 4E 75 61 00 0C 10 4A 40 67 10 04 40 01 382
056:11 39 40 00 14 20 6C 05 BA 31 40 FF FE 2A 6C 05 D39
057:C6 61 14 2A 6C 05 BE 0A 14 00 FF 67 0A 2A 6C 05 7DE
058:C2 29 4D 05 C6 4E 75 70 02 32 3C 00 E0 24 00 20 A4B
059:6C 05 B6 61 06 51 C8 FF FC 4E 75 48 42 30 C1 30 55E
060:C2 54 41 48 42 30 C1 30 C2 54 41 06 82 00 00 28 0F3
061:00 4E 75 4A 2C 00 0E 66 00 00 F8 4A 2C 00 04 67 DCB
062:00 01 10 41 EC 01 06 24 6C 05 D6 42 80 10 2C 00 37C
063:05 48 E7 80 80 12 10 4A 01 67 64 6B 00 02 FE 4A 992
064:28 00 0B 67 06 52 28 00 0B 60 54 12 28 00 04 4A 88E
065:01 67 00 03 36 B2 3C 00 01 67 00 03 42 B2 3C 00 5C8
066:02 67 00 03 5E B2 3C 00 10 64 00 03 B4 4A 28 00 E7C
067:0C 66 08 42 10 53 2C 00 04 60 24 10 28 00 06 12 B7C
068:28 00 07 14 28 00 08 16 28 00 01 2F 08 61 00 0D 49B
069:1C 20 5F 4B EC 00 1A 61 14 4B ED 00 46 61 0E 4C AB8
070:DF 01 01 41 EC 00 12 51 C8 FF 88 4E 75 4A 2D 00 FF0
071:09 6A 5E 4A 10 6B 5A B4 3C 00 08 62 54 48 80 48 5CA
072:81 3C 28 00 0E E2 4E 90 46 18 2D 00 06 48 84 06 F1B
073:44 00 1C B8 40 6D 3A 3C 28 00 10 E2 4E 92 46 1A 567
074:2D 00 07 48 85 06 45 00 0B BA 41 6D 24 00 68 00 8CF
075:0E 04 44 00 38 B8 40 6E 18 D2 68 00 10 04 45 00 2B0
076:16 BA 41 6E 0C 00 10 00 09 67 08 1B 7C 00 02 00 E29
077:08 4E 75 42 10 53 2C 00 04 42 40 10 28 00 0A D1 93D
078:6D 00 04 61 00 0E A2 70 08 12 2D 00 14 60 00 0F 520
079:76 19 7C 00 FF 00 0C 22 6C 05 A6 41 EC 01 06 42 447
080:83 1C 19 19 46 00 0D BC 3C 00 01 66 06 13 D9 00 6B4
081:00 00 01 19 59 00 08 19 59 00 09 42 86 1C 11 4A D29
082:06 6B 00 00 F2 53 06 10 29 00 02 B0 3C 00 46 66 220
083:06 70 60 72 60 60 04 61 00 0D D4 1A 29 00 05 14 269
084:29 00 06 18 29 00 03 4A 04 6A 12 C8 7C 00 7F E3 FC9
085:4C 45 F9 00 00 0B 52 45 F2 40 00 78 03 31 40 00 1AD
086:0E 31 41 00 10 10 A9 00 01 11 69 00 02 00 01 11 B3B
087:69 00 04 00 05 11 69 00 07 00 09 11 69 00 08 00 FDC
088:0A 11 69 00 09 00 0C 11 69 00 0A 00 0D 42 68 00 FFO

```

```

089:02 11 44 00 04 11 45 00 0B 4A 04 67 3C B8 3C 00 8A5
090:02 67 4A DA 02 B8 3C 00 03 67 5A B8 3C 00 01 66 53C
091:16 61 00 02 92 11 47 00 06 61 00 02 7A 11 47 00 A40
092:07 11 7C 00 FF 00 08 41 E8 00 12 52 03 51 CE FF 018
093:8E 43 E9 00 0B 60 00 FF 44 42 28 00 08 11 45 00 CF5
094:06 11 42 00 07 11 7C 00 FF 00 08 60 DA 42 48 00 9A0
095:08 42 68 00 06 11 7C 00 FC 00 08 11 45 00 02 11 68C
096:42 00 03 60 C2 42 28 00 04 31 5A 00 06 11 7C 00 BDA
097:FF 00 08 60 B2 52 89 BC 3C 00 FF 66 06 43 F9 00 C75
098:00 02 64 29 49 05 A6 19 43 00 04 53 03 19 43 00 08A
099:05 10 2C 00 0D 4A 00 67 56 19 7C 00 3F 00 0E 41 129
100:F9 00 00 00 FA B0 3C 00 01 67 30 41 F9 00 00 01 2D5
101:08 B0 3C 00 02 67 24 41 F9 00 00 01 1A B0 3C 00 F8B
102:03 67 18 41 F9 00 00 01 3B B0 3C 00 04 67 0C 41 51C
103:F9 00 00 01 58 19 7C 00 FF 00 0E 29 48 00 10 B0 164
104:3C 00 01 67 00 FA F6 B0 3C 00 02 67 00 FB 74 4E 706
105:75 4A 2C 00 0E 67 0C 53 2C 00 0E 20 6C 00 10 61 083
106:00 0D 3A 41 F9 00 00 01 F6 43 EC 00 A6 4A 29 00 0FB
107:54 66 00 0D 28 4E 75 4A 2C 00 01 66 3C 41 F9 00 888
108:00 01 E4 61 00 0D 16 42 86 1C 2C 00 02 7A 03 4A CC4
109:06 6A 02 7C 7F 41 F9 00 00 01 D2 22 6C 05 CE 43 98C
110:E9 03 00 43 F1 60 00 11 51 00 02 61 00 0C EE 54 93C
111:06 51 CD FF DC 52 2C 00 02 4E 75 0C 10 00 FF 67 24D
112:14 52 10 11 7C 00 01 00 0C 4A 28 00 01 66 00 FD B76
113:1E 60 00 FD 4C 14 28 00 04 52 02 11 42 00 04 B4 54B
114:3C 00 10 67 1A B4 3C 00 08 64 04 0A 02 00 07 C4 0CD
115:3C 00 07 D4 3C 00 22 11 42 00 01 60 00 FC F0 42 C4A
116:10 53 2C 00 04 60 00 FC F4 14 28 00 05 95 28 00 C62
117:08 64 00 01 AC 53 28 00 0C 60 00 FC D2 14 28 00 A84
118:05 95 28 00 08 64 00 01 98 53 28 00 0C 61 00 00 2F4
119:D6 11 47 00 06 61 00 00 BE 11 47 00 07 60 00 FC 1E9
120:AE 10 28 00 06 12 28 00 07 14 28 00 08 0D 28 00 6DE
121:02 B0 3C 00 92 6D 36 B0 3C 00 06 E3 30 D2 28 00 30A
122:03 B2 3C 00 A6 6D 2C B2 3C 00 5A 6E 26 D4 28 00 E6A
123:05 B4 3C 00 84 65 22 B4 3C 00 FC 64 1C 11 40 00 CF3
124:06 11 41 00 07 11 42 00 08 60 00 01 34 44 28 00 7D8
125:02 60 CA 44 28 00 03 60 DA 44 28 00 05 60 DE 42 501
126:82 42 83 34 28 00 02 16 28 00 05 48 83 DA 43 31 989
127:42 00 02 C4 7C 00 00 FF 66 04 53 28 00 0C 92 3C 00 EOE
128:10 C2 7C 00 7F E1 49 D2 42 C2 FC 00 03 06 41 03 ECA
129:80 22 6C 05 CE 43 F1 10 00 11 51 00 06 11 69 00 BB3
130:01 00 07 11 69 00 02 00 08 4A 28 00 0C 66 00 00 1F6
131:D0 60 00 FB FA 61 0E BE 3C 00 60 6E F8 BE 3C 00 BE2
132:A0 6D F2 4E 75 2E 2C 00 16 EA 8F DE 6E 00 06 CE 303
133:FC D0 0D DE BC 00 05 B0 65 29 47 00 16 4E 75 61 7AA
134:00 00 8E 4B EC 00 1A 3A BC 00 0C 1B 7C 00 07 00 D68
135:02 42 2D 00 14 42 2D 00 03 2B 7C 00 00 00 06 00 970
136:0C 61 46 61 00 08 22 4B ED 00 46 3A BC 00 0A 1B F54
137:7C 00 06 00 02 1B 7C 00 01 00 14 1B 7C 00 01 00 926
138:03 2B 7C 00 00 00 EB 00 0C 61 1E 1B 7C 00 20 00 84C
139:07 10 39 .. 00 04 19 40 00 01 B0 3C 00 02 67 00 77A
140:0A E6 42 2D 00 09 4E 75 42 6D 00 04 1B 7C 00 06 1BF
141:00 12 42 6D 00 10 1B 7C 00 00 00 11 42 6D 00 06 0F3
142:1B 7C 00 FF 00 09 61 00 0A 9E 60 00 0A 8A 20 4C EB5
143:20 3C 00 00 05 A5 42 18 51 C8 FF FC 4E 75 45 4C 2F9
144:00 A6 08 28 00 00 00 0D 67 2E 61 00 FF 38 CE 3C EA6
145:00 1F 08 28 00 05 00 0D 67 04 CE 3C 00 07 BE 3C 99F
146:00 06 66 14 22 4A 74 03 4A 11 66 04 61 74 60 08 E2E
147:43 E9 00 0C 51 CA FF F2 08 28 00 01 00 0D 67 3E ED3
148:43 EA 00 3C 4A 11 67 08 43 E9 00 0C 4A 11 66 2E 987
149:61 50 61 00 FE F0 08 07 00 00 67 04 52 29 00 05 671
150:CE BC 00 00 00 0E 33 78 70 06 00 0A 60 10 FC 00 B2E
151:04 00 00 FC 00 04 FC FC 04 FC FC 04 04 04 08 28 FEA
152:00 02 00 0D 67 00 FA A6 43 EA 00 30 4A 11 66 00 E58
153:FA 9C 61 0E 12 BC 00 04 13 7C 00 03 00 05 60 00 260
154:FA 8C 12 BC 00 03 33 68 00 06 00 02 13 68 00 08 26F
155:00 04 42 69 00 0A 13 7C 00 04 00 05 08 28 00 03 6ED
156:00 0D 67 06 13 7C 00 08 00 05 08 28 00 04 00 0D FCE
157:67 06 13 7C 00 0C 00 05 13 7C 00 1A 00 01 4E 75 DFD
158:41 EC 00 A6 7E 07 4A 10 67 2C 0C 10 00 04 67 30 29C
159:61 00 03 7C 12 28 00 05 93 28 00 04 64 04 42 10 A17
160:60 14 2F 08 61 00 06 32 20 5F 4B EC 00 1A 61 56 54F
161:4B ED 00 46 61 50 41 E8 00 0C 51 CF FF CA 4E 75 901
162:10 28 00 02 12 28 00 03 4B EC 00 1A 61 10 4B EC 546
163:00 60 61 0A 11 40 00 02 11 41 00 03 60 B6 4A 2D 620
164:00 09 6A 10 B0 2D 00 06 6E 0C 6D 0E B2 2D 00 07 47C
165:6E 0C 6D 0E 4E 75 57 00 60 F2 56 00 06 EE 57 01 FD6
166:4E 75 56 01 4E 75 4A 2D 00 09 6A 44 B4 28 00 05 EAB
167:62 3E 48 80 18 2D 00 06 48 84 06 44 00 28 B8 40 5F5
168:6D 2E 48 81 1A 2D 00 07 48 85 06 45 00 16 BA 41 72C
169:6D 1E 04 44 00 50 B8 40 6E 16 04 45 00 2C BA 41 2A5
170:6E 0E 0C 10 00 10 67 00 06 1C 1B 7C 00 01 00 08 1AF
171:4E 75 4A 2D 00 09 67 00 02 48 6A 00 03 72 4A 2D 4A8
172:00 08 66 00 03 5A 10 2D 00 06 12 2D 00 07 34 15 FF2
173:34 36 20 00 08 02 00 09 67 04 90 3C 00 0C 08 02 914
174:00 01 67 04 0D 3C 00 0C 36 02 E2 4B B5 43 08 03 B69
175:00 00 67 04 D2 3C 00 0C 08 03 00 08 67 04 92 3C 224
176:00 0C 4A 2B 00 09 67 32 48 80 48 81 14 2B 00 06 669
177:16 2B 00 07 48 82 48 83 04 42 00 04 B4 40 6E 1A 5E1
178:06 42 00 80 B4 40 6D 12 04 43 00 1C B6 41 6E 0A 5ED
179:06 43 00 38 B6 41 6D 02 60 20 B0 3C 00 90 6D 0A 207
180:B0 3C 00 70 6E 04 1B 40 00 06 B2 3C 00 0A 6D 0A 756
181:B2 3C 00 5C 6E 04 1B 41 00 07 10 2D 00 06 12 2D 066
182:00 07 08 2D 00 00 00 0A 66 0A 42 02 16 2D 00 03 505
183:61 00 05 F8 41 ED 00 16 10 39 00 BF 0E 01 12 2D B5F
184:00 15 1B 40 00 15 14 2D 00 02 05 00 66 00 01 2A 9D1
185:05 01 67 00 02 18 42 2D 00 13 08 2D 00 03 00 10 649

```


Listings

```

186:67 30 47 E8 00 18 0C 2B 00 01 00 04 67 0C 47 E8 97A
187:00 0C 0C 2B 00 01 00 04 66 18 37 7C 00 1C 00 06 C3E
188:37 7C 00 1C 00 08 17 7C 00 10 00 04 17 7C 00 10 08A
189:00 05 4A 10 67 08 41 E8 00 0C 4A 10 66 66 61 00 1A0
190:08 46 31 6D 00 06 00 02 42 28 00 04 10 BC 00 01 6BD
191:11 7C 00 12 00 01 31 7C 00 16 00 06 31 7C 00 16 9F7
192:00 08 11 7C 00 18 00 05 08 2D 00 01 00 10 67 10 234
193:10 BC 00 02 11 7C 00 20 00 05 11 7C 00 1A 00 01 336
194:08 2D 00 02 00 10 67 1C 10 BC 00 07 11 7C 00 02 117
195:00 01 31 7C 00 40 00 06 31 7C 00 1C 00 08 11 7C 024
196:00 14 00 05 08 2D 00 05 00 10 67 20 41 ED 00 2E E70
197:4A 10 66 18 31 68 00 0E 00 02 11 68 00 10 00 04 F5C
198:11 7C 00 20 00 05 42 68 00 0A 60 36 08 2D 00 04 1E2
199:00 10 67 44 41 ED 00 2E 4A 10 66 08 31 7C FF F6 4D1
200:00 0A 60 0E 41 E8 00 0C 4A 10 66 2C 31 7C 00 0A D07
201:00 0A 31 6D 00 06 00 02 42 28 00 04 11 7C 00 0A 54E
202:00 05 10 BC 00 08 11 7C 00 12 00 01 31 7C 00 16 3BB
203:00 06 31 7C 00 16 00 08 41 ED 00 16 7E 03 61 0A F9A
204:41 E8 00 0C 51 CF FF F8 4E 75 14 28 00 04 18 28 6AD
205:00 05 10 10 4A 00 67 28 80 3C 00 05 67 56 80 3C OCC
206:00 06 67 54 80 3C 00 08 67 18 80 3C 00 09 67 24 ADD
207:D4 04 11 42 00 04 B4 3C 00 E0 65 00 01 DA 42 10 0C9
208:4E 75 61 6A 04 04 64 04 42 10 4E 75 11 42 00 04 690
209:60 00 01 C4 61 58 D4 04 B4 3C 00 22 65 10 B4 3C 56B
210:00 E0 62 0A 11 42 00 04 78 08 60 00 01 AA 44 28 2ED
211:00 05 60 F0 70 E4 60 02 70 1C D0 2D 00 07 11 40 DC5
212:00 03 11 6D 00 06 00 02 B4 3C 00 01 67 14 D4 04 628
213:66 10 4A 04 6A 06 74 01 78 08 60 06 78 F0 14 3C 4E7
214:00 F0 11 42 00 04 11 44 00 05 60 00 01 6A 10 28 6C5
215:00 02 12 28 00 03 D0 28 00 0A B0 3C 00 8B 6D 20 14A
216:B0 3C 00 75 6E 1A D2 28 00 0B B2 3C 00 9F 6D 16 8EE
217:B2 3C 00 61 6E 10 11 40 00 02 11 41 00 03 4E 75 E80
218:44 28 00 0A 60 E0 44 28 00 0B 60 EA 52 2D 00 13 197
219:0C 2D 00 19 00 13 67 04 60 00 FE FE 4A 2D 00 11 FAA
220:67 00 FE F6 53 2D 00 11 39 7C 11 10 00 14 70 05 OFA
221:61 00 06 8C 43 EC 01 06 42 84 18 2C 00 05 4A 11 A4B
222:67 18 6B 16 4A 29 00 0B 66 10 55 29 00 09 65 06 6BD
223:4A 29 00 09 66 04 61 00 01 B8 43 E9 00 12 51 CC A58
224:FF DE 61 00 05 90 60 00 FE B0 BE 3C 00 60 65 00 83C
225:FC B6 42 2D 00 0A 18 7C 00 FF 00 09 4E 75 70 04 904
226:12 2D 00 14 61 00 06 3E 1B 7C 00 01 00 09 1E 2D ECD
227:00 0A 52 07 1B 47 00 0A BE 3C 00 38 67 5E 62 CA 4E9
228:10 2D 00 06 12 2D 00 07 42 02 BE 3C 00 18 64 08 8C3
229:16 2D 00 03 61 00 03 14 58 00 61 20 51 00 59 07 F4E
230:65 16 61 18 D0 3C 00 10 59 07 65 0C 61 0E 90 3C 373
231:00 18 59 07 65 02 61 04 60 00 FE 3E 16 07 B6 3C BE2
232:00 20 64 16 B6 3C 00 10 64 04 0A 03 00 0F C6 3C 6C1
233:00 0F D6 3C 00 22 60 00 02 D2 4E 75 53 2D 00 12 28C
234:61 00 05 04 4A 2D 00 12 67 12 1B 7C 00 02 00 09 B1D
235:42 2D 00 08 42 2D 00 10 60 00 02 3E 42 2D 00 09 C9A
236:24 6D 00 0C 45 EA 00 07 14 FC 00 2D 14 BC 00 2D 890
237:53 2C 00 01 4E 75 10 28 00 02 12 28 00 03 48 80 4BF
238:48 81 16 02 D6 04 43 EC 01 06 42 84 18 2C 00 05 9CC
239:4A 11 67 58 6B 56 4A 29 00 08 66 50 1A 29 00 08 5FF
240:BA 02 65 48 BA 03 62 44 1A 29 00 06 48 85 3C 29 6F0
241:00 0E DC 68 00 06 E2 4E 9A 46 BA 40 6E 2E DA 69 B98
242:00 0E DA 68 00 06 BA 40 6D 22 1A 29 00 07 48 85 397
243:3C 29 00 10 DC 68 00 08 E2 4E 9A 46 BA 41 6E 0C E33
244:DA 69 00 10 DA 68 00 08 BA 41 6E 12 43 E9 00 12 243
245:51 CC FF 9E 2F 08 61 00 00 E0 20 5F 4E 75 08 2D 840
246:00 00 10 66 14 0C 10 00 05 67 0E 0C 10 00 06 752
247:67 08 0C 10 00 09 67 02 42 10 53 29 00 09 4A 29 9D4
248:00 09 67 08 70 06 61 00 04 D6 60 1E 70 03 0C 11 3DA
249:00 05 66 02 70 04 61 00 04 C6 42 40 10 29 00 0A 3AA
250:01 6D 00 04 61 0A 61 00 03 DE 4A 10 66 A6 4E 75 43E
251:12 11 12 BC 00 FF 42 29 00 04 B2 3C 00 06 66 16 6AB
252:70 07 14 3C 00 EF 43 E9 00 12 12 82 42 29 00 04 OAD
253:54 02 51 C8 FF F2 B2 3C 00 05 66 5A 2F 07 12 BC ACE
254:00 EE 70 07 14 3C 00 F0 36 29 00 06 18 29 00 08 BB6
255:59 04 43 E9 00 12 12 82 42 29 00 04 42 29 00 01 07F
256:33 43 00 06 13 44 00 08 61 00 F8 3A CE 3C 00 3F D8A
257:04 07 00 20 DF 29 00 06 E0 4F CE 3C 00 3F 04 07 53C
258:00 20 DF 29 00 07 54 02 51 C8 FF C8 50 2C 00 04 2B1
259:50 2C 00 05 2E 1F 4E 75 10 28 00 02 12 28 00 03 3EA
260:14 28 00 04 16 28 00 01 60 00 01 20 10 2C 00 08 5F5
261:12 2C 00 0C 92 00 65 06 19 41 00 0C 4E 75 19 7C 467
262:00 FF 00 0C 42 2C 00 08 42 87 1E 2C 00 09 E7 4F 81B
263:43 F9 00 00 02 08 43 F1 70 00 19 51 00 08 19 69 E82
264:00 01 00 0A 45 F9 00 00 02 00 24 A9 00 02 35 69 DDB
265:00 06 00 04 41 EC 00 A6 41 E8 00 54 10 BC 00 10 259
266:11 7C 00 42 00 01 61 00 F7 8C 11 47 00 02 61 00 F4B
267:F7 84 11 47 00 03 11 7C 00 FF 00 04 11 7C 00 03 76C
268:00 05 4E 75 42 10 0C 2C 00 06 00 09 67 00 00 92 BE9
269:61 26 10 2D 00 10 C0 2C 00 0B 80 2C 00 0A 1B 40 6FC
270:00 10 08 00 00 03 66 1C 08 00 00 05 66 3E 70 07 E9F
271:12 2D 00 14 60 00 03 6E 47 ED 00 2E 42 13 42 2B B31
272:00 0C 4E 75 47 ED 00 2E 16 BC 00 05 17 7C 00 5E 9FD
273:00 01 17 7C 00 01 00 04 17 7C 00 06 00 0C 17 7C 57C
274:00 5E 00 0D 17 7C 00 01 00 10 60 C2 45 ED 00 3A E08
275:14 BC 00 09 15 7C 00 55 00 01 35 6D 00 06 00 02 FC3
276:15 7C 00 20 00 04 35 7C 02 02 00 0A 15 7C 00 02 EC2
277:00 05 35 7C 00 24 00 06 35 7C 00 24 00 08 60 8E 586
278:54 2D 00 11 61 00 02 2E 60 84 41 EC 04 66 42 84 9A0
279:42 85 1A 2C 00 03 B8 05 67 26 B4 28 00 02 62 08 DE6
280:41 E8 00 04 52 04 60 EE 43 EC 04 66 3C 00 04 6AC
281:CC C5 43 F1 60 00 22 A9 FF FC 59 89 B1 C9 66 F6 B67
282:10 C0 10 C1 10 C2 10 C3 52 05 19 45 00 03 4E 75 419

```

```

283:2A 6C 05 C6 08 2E 00 0E 00 02 66 F4 2D 4D 00 54 B03
284:42 6E 00 42 42 6E 00 66 3D 7C 01 00 00 40 3D 7C 3A8
285:C0 14 00 58 4E 75 2A 6C 05 C6 41 EC 04 66 42 80 A6A
286:10 2C 00 03 4A 00 67 10 53 00 61 0E 41 E8 00 04 D76
287:51 C8 FF F8 42 2C 00 03 4E 75 42 81 42 82 42 83 770
288:43 F9 00 00 01 86 12 28 00 02 18 01 EA 0C 43 F1 OAC
289:10 00 12 11 14 10 16 28 00 01 48 82 48 83 C5 C1 ED1
290:C7 C1 E0 82 E0 83 06 42 00 A0 06 43 00 80 42 81 82D
291:12 28 00 03 D2 04 E7 49 22 6C 05 D6 43 F1 10 00 AEF
292:32 11 E2 49 94 41 32 29 00 02 E2 49 96 41 08 2E EC9
293:00 0E 00 02 66 F8 42 81 24 6C 05 CE 26 4A 32 29 575
294:00 06 D5 C1 2D 4A 00 50 32 29 00 04 D7 C1 2D 4B 2F6
295:00 4C 42 84 38 03 C8 FC 00 28 32 02 E6 49 08 81 635
296:00 0D 08 41 D8 8D 2D 44 00 54 2D 44 00 48 32 02 773
297:C2 7C 00 0F E1 49 E9 49 3D 41 00 42 82 7C 0F C2 C08
298:3D 41 00 40 42 6E 00 64 42 6E 00 62 3A 11 06 45 719
299:00 1F E8 4D 72 14 92 45 E3 49 3D 41 00 66 3D 41 06B
300:00 06 32 29 00 02 ED 49 82 45 3D 41 00 58 08 2E F84
301:00 0E 00 02 66 F8 2D 4B 00 4C 06 84 00 00 28 00 4D8
302:2D 44 00 54 2D 44 00 48 3D 41 00 58 08 2E 00 0E B22
303:00 02 66 F8 2D 4B 00 4C 06 84 00 00 28 00 2D 44 ACB
304:00 54 2D 44 00 48 3D 41 00 58 4E 75 C0 BC 00 00 010
305:00 FF E7 48 26 6C 05 D6 47 F3 00 00 30 13 32 2B C00
306:00 02 4E 75 2A 6C 05 C6 4A 2C 00 07 67 26 41 ED A29
307:3C 00 43 F9 00 00 01 86 70 00 10 2C 00 06 5B 00 C2A
308:C0 3C 00 3F 19 40 00 06 61 0C 06 40 00 40 B0 7C 3AD
309:02 00 65 F4 4E 75 42 81 12 31 00 00 E2 09 C2 FC D69
310:00 28 D2 88 08 2E 00 0E 00 02 66 F8 2D 41 00 54 BBF
311:42 6E 00 66 42 6E 00 42 3D 7C 01 FF 00 40 3D 7C 16B
312:00 54 00 58 4E 75 42 80 30 2D 00 04 24 6D 00 0C 3E0
313:72 01 60 22 42 80 10 2D 00 11 24 6D 00 0C 45 EA 682
314:00 0C 42 81 60 10 42 80 10 2D 00 12 24 6D 00 0C 05C
315:45 EA 00 07 42 81 4A 01 67 0C 8C FC 03 E8 61 05 DAO
316:80 FC 00 64 61 08 80 FC 00 0A 61 00 00 02 D0 3C OED
317:00 30 14 C0 42 40 48 40 4E 75 42 87 24 6C 05 CE E54
318:42 80 10 18 4A 00 67 2E 6B 2E 90 3C 00 20 E7 48 DB9
319:47 F2 00 00 43 F5 70 00 70 07 12 1B 83 11 0A 01 883
320:00 FF C3 29 28 00 C3 29 50 00 43 E9 00 28 51 C8 9D9
321:FF EA 52 47 60 CA 4E 75 42 47 10 18 1E 18 CE FC 43C
322:00 28 DE 40 60 BA 12 2D 00 14 70 00 08 2D 00 01 417
323:00 10 67 02 70 02 08 2D 00 02 00 10 67 02 70 01 01F
324:60 22 3D 7C 47 80 00 9A 29 78 00 70 05 D2 21 FC 93E
325:00 14 B2 00 70 3D 7C 0C 00 00 9A 4E 75 12 2D 992
326:00 14 52 01 42 42 03 C2 38 02 EF 4C 3D 44 00 9A 84A
327:3D 42 00 96 42 83 16 01 E9 0B 06 43 00 A0 47 F6 178
328:30 00 36 00 E7 4B 45 F9 00 00 00 02 45 F2 30 00 47C
329:42 83 36 1A D6 AC 05 CA 26 83 27 52 00 04 37 6A C82
330:00 04 00 08 42 83 36 2A 00 02 51 CB FF FE 08 C2 818
331:00 0F 3D 42 00 96 26 3C 00 00 00 C8 51 CB FF FE B08
332:30 44 00 9C 08 C4 00 0F 3D 44 00 9A 4E 75 48 E7 66A
333:C0 80 41 F9 00 DF F0 00 30 28 00 1E C0 7C 07 80 703
334:32 00 EE 49 31 41 00 96 31 40 00 9A 31 40 00 9C 010
335:4C DF 01 03 4E 73 10 00 04 03 EC 00 00 00 06 00 784
336:00 00 02 00 00 00 02 00 00 01 8C 00 00 01 F6 00 CB6
337:00 02 1A 00 00 11 E6 00 00 13 18 00 00 00 01 E30
338:00 06 14 34 00 00 00 1F 00 00 00 01 00 00 00 1C 4C3
339:00 00 5A 00 00 00 72 00 00 00 E8 00 00 01 60 675
340:00 00 61 68 00 00 02 20 00 00 02 6A 00 00 02 8C 11E
341:00 00 02 94 00 00 02 A2 00 00 02 AA 00 00 02 BE 796
342:00 00 02 F4 00 00 04 F2 00 00 05 36 00 00 06 02 477
343:00 00 06 24 00 00 06 30 00 00 06 3C 00 00 06 48 679
344:00 00 06 54 00 00 06 88 00 00 06 A2 00 00 06 BA 591
345:00 00 08 5E 00 00 08 86 00 00 08 96 00 00 10 56 6DF
346:00 00 10 6A 00 00 14 6C 10 00 06 03 F2 00 00 03 2C0
347:EA 00 00 02 FF 01 01 10 00 08 00 00 AA 00 30 00 335
348:00 08 00 00 C8 00 20 00 08 00 00 00 7C 00 28 20 C99
349:00 08 00 02 58 00 32 20 00 08 00 05 DC 00 40 00 3D8
350:00 08 00 01 90 00 18 10 00 08 00 00 7C 00 1C 30 C96
351:00 10 00 00 7C 00 38 30 00 10 00 00 A0 00 38 01 FEA
352:00 0F 0F 0F 0F 0F 00 00 10 0F F0 0F 00 00 F0 0F 78F
353:F1 0F 0F 0F 0F 0F 01 00 0F F0 0F 0F 0F 0F 00 14C
354:01 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 00 9B5
355:01 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 00 6E8
356:45 FF 10 5C 53 20 54 20 52 20 49 20 48 20 45 FF DA4
357:09 6C 62 79 20 4F 6C 69 76 65 72 20 48 65 67 67 04E
358:65 6C 62 61 63 68 65 72 FF 0E 7C 28 63 29 20 31 9F4
359:39 38 39 20 44 4D 56 FF 0F A8 50 6C 61 79 65 72 168
360:73 3A 20 31 FF 00 00 50 4C 31 3A 30 04 20 4C F97
361:3A 2D 2D 20 42 3A 30 30 FF 16 00 50 4C 32 3A 00 06D
362:30 04 20 4C 3A 2D 2D 20 42 3A 30 30 00 FF 0F F0 47B
363:50 6C 61 6E 65 74 66 61 6C 6C 00 FF 0F 0F 45 6E 78A
364:74 65 72 69 6E 67 20 53 70 61 63 65 00 FF 05 F0 489
365:42 6F 6E 75 73 72 6F 75 6E 64 3A 20 43 6F 6C 6C DB7
366:65 63 74 20 74 68 65 20 62 61 6C 6C 73 00 FF 07 A23
367:F0 57 61 72 6E 69 6E 67 3A 20 4D 6F 74 68 65 72 477
368:73 68 69 70 20 61 68 65 61 64 00 FF 0A 4C 2D 2D 234
369:43 4F 4E 47 52 41 54 45 44 41 54 49 4F 4E 53 2D 669
370:2D FF 0C 5C 59 6F 75 20 68 61 76 65 20 77 6F 6E 0D1
371:20 74 68 65 FF 0D 64 47 4F 4C 44 45 4E 20 50 4F 1FE
372:54 41 54 4F 45 FF 06 74 66 6F 72 20 62 65 61 72 525
373:69 6E 67 20 61 6C 6C 20 66 69 76 65 20 6C 65 76 3E1
374:65 6C 73 FF 04 7C 6F 66 20 74 68 69 73 20 69 6E 840
375:63 72 65 64 69 62 6C 79 20 73 74 75 70 69 64 20 782
376:67 61 6D 65 00 FF 0D 00 2D 2D 2D 47 20 41 20 4D 222
377:2D 45 2D 2D 00 FF 0D 9A 2D 2D 04 4F 20 56 20 DA2
378:45 2D 52 2D 2D 00 FF 0E DC 45 78 74 72 61 3A AD5
379:2D 07 00 00 FF 01 50 6F 77 65 72 00 FB 02 53 BCE

```



```

380:70 65 65 64 00 FD 04 53 68 61 64 6F 77 CF 08 42 445
381:61 6C 6C 73 00 D7 10 53 69 64 65 00 00 E7 20 53 B66
382:61 74 65 6C 00 FF 00 42 6F 6D 62 73 00 07 72 61 54B
383:70 68 69 63 73 2E 6C 69 62 72 61 72 79 00 64 6F 565
384:73 2E 6C 69 62 72 61 72 79 00 44 53 64 61 74 61 A15
385:00 01 00 00 00 07 03 32 01 06 05 06 01 01 04 00 930
386:FE 00 00 00 08 03 32 01 04 00 07 01 02 04 09 FE 540
387:00 04 00 08 04 3A 10 01 00 20 01 02 04 00 02 03 9C3
388:32 01 02 80 20 02 03 04 01 FE 00 02 01 04 04 3A CE8
389:10 01 80 08 02 02 FF 00 04 02 32 10 FF 01 08 01 1CD
390:02 FF 09 FE 00 00 00 04 03 32 12 01 80 08 01 02 5F3
391:02 01 02 04 3A 16 01 00 80 04 06 FF 04 FE 00 02 87F
392:06 08 04 3A 11 01 E0 04 02 02 03 00 FE 00 00 00 309
393:0A 04 3A 18 02 7E 0D 01 02 FF 00 01 04 3A 00 04 E7F
394:00 00 04 04 FF 02 FE 00 02 01 08 04 3A 14 02 FO AE1
395:02 02 03 02 04 FE 00 08 00 08 04 3A 01 08 E0 04 120
396:01 03 04 00 FE 00 08 06 04 03 32 88 03 00 00 01 0F8
397:06 01 09 04 03 32 8C 03 C0 00 01 06 01 09 FE 04 6ED
398:03 02 01 05 46 02 FE FE FE 10 14 01 29 FE 03 00 63A
399:00 10 09 5E 01 0C 00 10 FF 02 01 00 FE 02 00 00 9C6
400:07 01 4E 01 06 D5 06 FF 00 08 00 FE 01 01 02 03 B17
401:08 02 0A 11 01 C0 08 01 03 FF 09 FE 00 02 03 02 505
402:02 0A 11 02 80 40 01 04 FF 09 FE 00 00 00 08 02 16A
403:0A 13 01 C0 08 02 03 01 09 08 03 32 13 FF C0 08 9B1
404:02 03 01 02 FE 00 00 00 01 03 32 02 FC FC 04 04 D5D
405:08 FF 31 08 02 0A 10 01 C0 08 01 03 FF 00 FE 00 348
406:80 04 02 02 0A 11 01 F8 08 01 02 FF 00 02 02 0A 018
407:11 FE F8 08 01 02 FF 00 FE 00 00 00 02 02 0A 11 04F
408:01 FC 04 01 04 FF 09 02 02 0A 11 FE FC 04 01 04 B3D
409:FF 00 FE 00 08 06 04 04 3A 88 04 C0 10 01 06 01 3A2
410:02 04 04 3A 88 04 B0 F0 01 06 01 02 04 04 3A 8C DDD
411:04 41 10 01 06 01 02 04 04 3A 8C 04 31 F0 01 06 E8C
412:01 02 04 03 32 18 02 F0 04 01 02 FF 00 FE 00 20 494
413:05 04 02 0A 16 02 F0 04 01 04 FF 01 04 02 0A 17 806
414:02 B0 04 01 04 FF 02 04 02 0A 16 FE 70 04 01 04 AB4
415:FF 09 04 02 0A 17 FE 30 04 01 04 FF 09 FE 00 20 157
416:01 04 04 3A 80 04 00 00 01 03 01 09 04 04 3A 80 DB0
417:04 80 00 01 04 01 04 04 04 3A 80 04 01 00 01 04 D05
418:01 11 04 03 32 10 01 E8 06 02 02 FF 00 FE 00 08 C55
419:02 04 04 3A 84 02 C0 00 01 03 01 02 08 02 0A 18 4E2
420:01 E0 04 01 03 FF 09 FE 00 20 04 01 06 32 10 01 BCE
421:00 00 08 14 FF 00 08 04 3A 10 01 F0 02 FF 00 FF C24
422:09 FE 00 20 00 08 04 3A 80 04 E0 04 01 03 01 09 E2C
423:08 04 3A 80 04 60 04 01 03 01 02 10 03 32 14 01 809
424:00 08 01 02 01 00 FE 00 04 03 10 02 0A 18 02 04 4FA
425:06 02 03 02 00 FE 00 02 06 08 02 0A 18 01 A0 OC D01
426:01 02 02 01 10 04 3A 88 02 00 10 01 03 01 09 FE B0B
427:04 00 00 01 05 46 02 FE 03 03 12 28 01 0B FE 03 62A
428:00 00 10 09 5E 01 0C 00 10 FF 03 01 00 FE 02 04 350
429:00 07 01 4E 01 06 D6 06 FF 00 0A 00 04 03 32 13 EC5
430:01 D8 0A 01 02 01 09 04 04 3A 13 01 58 0A 01 02 E3E
431:01 09 04 02 0A 80 03 01 00 01 02 01 09 FE 01 02 721
432:00 00 0C 04 3A 98 02 40 10 01 03 01 09 06 08 66 8EB
433:16 02 E8 04 01 04 03 04 06 08 66 16 FE A8 04 01 D5A
434:04 03 02 FE 00 02 00 10 01 4E 88 01 01 10 FF 00 BA9
435:01 06 FE 00 20 03 08 02 0A 14 02 01 23 01 03 03 6D5
436:09 FE 00 00 00 08 01 4E 18 01 C0 08 FF 00 02 09 C50
437:08 04 3A 18 01 C2 08 02 04 03 00 FE 00 08 01 04 127
438:01 4E 80 02 C0 00 FF 00 01 02 04 01 4E 84 02 04 278
439:00 FF 00 01 02 10 04 3A 15 01 00 10 01 03 01 00 D4E
440:FE 00 00 00 08 03 32 10 01 C0 08 02 04 01 09 08 55F
441:08 66 1B 02 80 04 01 03 02 02 FE 00 03 05 01 06 459
442:32 10 01 80 00 08 10 03 0B 08 04 3A 10 01 68 03 DEF
443:FF 00 03 00 FE 03 00 00 10 09 5E 01 04 00 10 FF 780
444:05 01 00 08 02 0A 11 01 E0 04 01 04 02 09 FE 00 A92
445:00 00 08 01 4E 1B 01 80 08 FF 00 01 02 08 01 4E E6C
446:1C 01 80 08 FF 00 01 04 FE 00 00 00 04 04 3A 16 C0F
447:04 00 41 02 04 FF 09 04 08 66 A4 01 00 00 03 06 0F8
448:01 09 FE 00 08 02 06 03 32 18 01 E8 04 02 04 FF 401
449:09 06 04 3A 18 FF A8 04 02 04 FF 02 FE 00 00 00 707
450:08 03 3A 1B 02 C0 08 01 03 01 02 08 03 3A 1C 02 1E9
451:C0 08 01 03 01 02 04 04 32 A8 02 E0 00 01 03 01 FA8
452:00 04 04 32 AC 02 C0 00 01 03 01 00 FE 02 00 00 E52
453:0A 01 4E 01 05 CE 05 FF 00 08 00 04 02 0A 01 04 907
454:C0 10 01 04 01 09 08 03 32 19 02 01 05 01 04 01 173
455:00 FE 00 00 00 0A 01 4E 01 05 CE 05 FF 00 08 00 C48
456:0C 04 3A 98 02 80 00 04 05 01 00 FE 01 01 20 01 50D
457:06 03 32 18 02 80 10 02 05 FF 09 06 04 3A 18 02 46B
458:08 10 02 05 FF 09 FE 00 20 02 0C 04 3A 98 04 00 C45
459:00 01 03 01 02 0C 04 3A 98 04 80 00 01 04 01 31 C15
460:0C 04 3A 98 04 01 00 01 03 01 0D 04 08 66 A4 08 A08
461:80 00 01 08 01 09 FE 00 10 04 0C 02 0A 15 01 01 446
462:15 01 04 01 09 0C 07 22 98 06 0B 15 FF 00 01 00 DDO
463:0C 07 22 98 06 F2 EB FF 00 01 00 0C 04 3A 01 06 DC6
464:06 15 01 02 01 00 FE 00 08 00 01 06 32 11 01 00 800
465:00 08 0C FF 31 08 04 3A 11 01 F0 02 FF 00 FF 00 940
466:04 04 3A 90 03 C0 00 02 05 01 02 04 04 3A 94 03 DFA
467:80 00 02 05 01 02 08 02 0A 15 01 00 20 01 03 01 EA8
468:11 FE 00 00 00 0C 02 0A 13 02 D0 04 01 03 01 13 431
469:FE 00 10 05 01 06 32 11 01 00 00 08 0C FF 11 08 FF6
470:04 3A 11 01 F0 02 FF 00 FF 11 01 06 32 11 01 08 BC3
471:00 08 0C FF 11 08 04 3A 11 01 70 02 FF 00 FF 11 9A0
472:08 07 26 01 04 E0 04 FF 00 06 00 FE 00 00 08 6DE
473:08 66 19 01 A0 0C 01 03 01 09 08 08 66 1A 01 A0 D6A
474:0C 01 03 01 09 FE 00 04 03 0C 08 66 1B 02 40 10 FA1
475:01 03 01 0B 0C 08 66 1C FE 40 10 01 03 01 09 04 734
476:04 3A A4 06 0C 0C 01 05 01 11 FE 00 00 00 06 08 C11

```

```

477:66 16 02 D0 08 02 04 FF 11 06 04 3A 16 02 4C 08 OBC
478:02 04 FF 02 06 03 32 17 02 90 08 02 04 FF 11 06 174
479:04 3A 17 02 0C 08 02 04 FF 02 FE 00 10 00 01 06 784
480:32 19 01 00 00 04 0F 01 23 08 04 3A 19 01 E0 04 92C
481:FF 00 01 00 01 06 66 1A 01 00 00 04 0F 01 25 08 DAA
482:01 4E 1A 01 E0 04 FF 00 01 00 FE 04 08 02 01 06 EF9
483:66 11 01 00 00 04 0A FF 11 08 01 4E 11 01 E8 03 69A
484:FF 00 FF 00 01 05 46 02 FE 02 02 0E 19 01 13 FE 4DB
485:00 00 00 07 07 26 01 06 D4 06 FF 00 06 00 04 08 20F
486:66 16 02 F0 04 01 03 FF 02 04 08 66 16 FE B0 04 B5D
487:01 03 FF 02 08 04 3A 18 01 D0 06 01 04 FF 11 FE 723
488:00 08 04 08 02 0A 15 01 C0 08 01 02 01 00 08 04 41A
489:3A 14 01 E0 04 01 03 01 11 04 08 66 80 06 F0 00 370
490:02 05 01 02 04 08 66 84 06 A0 00 02 05 01 02 04 A39
491:08 66 80 06 50 00 02 05 01 02 FE 04 20 06 01 05 CD9
492:46 02 FD 03 03 14 28 01 37 FE 02 00 00 06 01 4E 780
493:01 07 DB 06 FF 00 0A 00 20 08 66 01 04 00 07 01 C33
494:03 01 11 FE 01 00 00 00 20 04 3A 01 0C 04 06 01 1DD
495:04 01 00 FE 00 10 04 10 08 28 01 0A 04 0C FF 00 B08
496:01 00 10 08 26 01 08 08 46 0C 00 01 00 08 02 0A 187
497:01 06 40 18 02 04 01 09 FE 00 08 05 08 04 3A 19 C01
498:02 82 10 02 04 01 00 08 03 32 19 02 80 10 01 02 2E4
499:01 11 FE 00 00 00 08 04 3A 17 04 01 21 04 04 FF AEC
500:11 01 06 32 1C 01 00 00 08 0A 01 02 08 04 3A 1C BBA
501:01 F0 02 FF 00 01 00 FE 00 10 06 08 04 3A 17 FC D9F
502:01 41 02 04 FF 11 08 04 3A 16 FC 20 41 02 04 FF 816
503:02 FE 00 00 00 04 01 4E 88 04 00 00 FF 00 01 11 090
504:04 01 4E A8 04 C0 00 FF 00 01 11 04 01 4E AC 04 568
505:80 00 FF 00 01 11 04 01 4E 8C 04 40 00 FE 00 01 D17
506:11 08 04 3A 10 01 E0 04 02 04 02 02 FE 00 20 03 344
507:0C 04 3A 98 01 00 00 03 06 01 11 06 08 66 12 01 3EC
508:40 06 02 04 FF 06 FE 00 04 01 01 06 66 1B 01 00 4E1
509:00 08 0A 03 02 08 01 4E 1B 01 F0 02 FF 00 03 00 3F2
510:01 06 66 1C 01 80 00 08 0A 03 04 08 01 4E 1C 01 944
511:70 02 FF 00 03 00 08 08 66 18 01 88 0C 01 03 FF 888
512:11 FE 00 08 06 01 06 32 10 01 00 00 0A 14 FF 21 191
513:08 04 3A 10 01 F0 02 FF 00 FF 00 01 06 66 10 01 DEF
514:80 00 0A 14 FF 06 08 01 4E 10 01 70 02 FF 00 FF CFA
515:00 FE 00 10 05 04 02 0A 16 FE E0 08 02 04 FF 21 384
516:04 02 0A 16 02 DE 08 02 04 FF 21 08 04 3A 18 01 26B
517:C0 08 03 08 FF 02 FE 00 00 00 01 06 66 11 01 00 716
518:00 0A 14 FF 02 08 04 3A 11 01 F0 02 FF 00 FF 11 4A2
519:08 07 27 01 04 C0 10 FF 00 01 04 00 FE 04 00 08 16A
520:08 66 01 06 00 20 01 06 01 11 01 05 46 02 FD 03 3D0
521:03 10 19 01 17 FE 00 10 02 01 06 66 18 01 00 00 ABC
522:0A 14 FF 02 08 04 3A 18 01 E8 03 FF 00 FF 11 01 E43
523:06 66 18 FF 00 00 0A 14 FF 02 08 04 3A 18 FF E8 20A
524:03 FF 00 FF 11 FE 04 00 00 01 05 46 02 FC 04 04 76A
525:18 64 01 37 FE 05 00 00 10 09 5E 98 02 00 00 FF 86C
526:0A 01 00 FF 01 00 00 00 01 03 32 11 01 00 00 01 2CF
527:00 FF 00 08 04 3A 11 01 F0 02 01 00 FF 00 FF 88 32F
528:00 78 00 00 A0 00 60 88 A0 78 A0 88 60 78 60 88 BF7
529:A0 D8 A0 28 A0 78 A0 88 60 78 60 28 60 78 60 88 E14
530:A0 88 E0 88 20 88 60 78 A0 78 E0 78 20 78 60 88 881
531:A0 88 E0 88 20 88 60 D8 60 28 60 78 60 78 20 78 603
532:E0 78 A0 28 A0 D8 A0 D8 E0 28 60 78 20 28 20 88 A24
533:E0 D8 E0 28 E0 78 E0 88 20 D8 20 28 20 78 20 0F 601
534:FF 0B BB 08 88 0F 00 0F A0 0F 06 6F 00 04 00 BF4
535:01 00 0F 00 10 00 FF 0F FF 0B BB 08 88 00 80 03 E2D
536:A0 0F 06 6F 00 08 01 00 0F 08 00 01 0F 0F 0F 70E
537:FF 0B BB 08 88 06 68 08 8B 0A AF 06 6F 00 40 01 DC6
538:00 0F 40 00 01 0F 4F .. 00 04 03 F2 00 00 03 EB 5EE
539:00 00 04 E2 00 00 03 F2 00 00 00 00 00 00 00 00 AAB

```

```

*****
*                               *
*      AMIGA DOS INFO          *
*      Programmname : DSGRAPH   *
*      Funktion     : Erzeugt Grafik für DoubleStrike *
*      Sprache       : HEXDUMP   *
*      Besonderheiten: Muß mit HEXTYPE eingegeben werden! *
*****

```

```

000:00 40 00 1C 7E .. 00 06 7E 3F 80 00 01 80 00 01 86A
001:FC 3F E0 01 FE 7F 80 07 FC 1D F0 03 00 00 C0 0F 366
002:B8 0E F8 .. 00 04 1F 70 0E 7C .. 00 04 3E 70 07 F17
003:3E 07 FE 7F E0 7C E0 07 FF 18 00 00 18 FF E0 03 86A
004:6F 27 FF FF E4 F6 C0 03 6F 28 00 00 14 F6 C0 01 A96
005:EF FF F0 0F FF F7 80 01 EE 20 1C 38 04 77 80 01 C86
006:EC 4F CE 73 F2 37 80 01 EF D0 2E 74 0B F7 80 01 663
007:EF D0 2E 74 0B F7 80 01 EC 4F CE 73 F2 37 80 01 574
008:EE 20 1C 38 04 77 80 01 EF FF F0 0F FF F7 80 03 925
009:6F 28 00 00 14 F6 C0 03 6F 27 FD 0F E4 F6 C0 07 B5E
010:FF 18 00 00 38 FF E0 07 3E 07 FF FF E0 7C E0 0E 7C4
011:7C .. 00 04 3E 70 0E F8 .. 00 04 1F 70 1D F0 .. 62D
012:00 04 0F B8 3F E0 .. 00 04 07 FC 3F 80 .. 00 04 954
013:01 FC 7E .. 00 06 7E 7E 00 00 01 80 00 00 7E 21 21B
014:80 01 FE 7F 80 01 84 2E E0 03 FF FF C0 07 74 17 968
015:70 07 01 80 E0 0E E8 0B 88 00 01 80 00 10 D0 08 EE4
016:DC 00 01 80 00 3B D0 05 EE 07 FF FF E0 77 A0 05 C47
017:EF 1F FF FF F8 F7 A0 02 F7 38 00 00 1C EF 40 02 976
018:F7 37 FF FF EC EF 40 01 77 E0 1F F8 07 EE 80 01 645
019:F7 C0 0F F0 03 EE 80 01 77 80 07 E0 01 EE 80 01 96B
020:74 0F C7 E3 F0 2E 80 01 74 0F C7 E3 F0 2E 80 01 F30

```


Listings

```

021:77 80 07 E0 01 EE 80 01 77 C0 OF F0 03 EE 80 01 A7B
022:77 E0 1F F8 07 EE 80 02 F7 37 FF FF EC EF A0 02 0D8
023:F7 38 06 30 1C EF 40 05 EF 1F FF FF F8 F7 A0 05 793
024:EE 07 FF FF E0 77 A0 0B DC .. 00 04 3B 0D 0B B8 7DD
025:.. 00 04 1D D0 17 70 .. 00 04 0E E8 2E E0 .. 00 E72
026:00 07 74 21 80 .. 00 04 01 84 7E .. 00 06 7E .. ABD
027:00 10 0E .. 00 06 70 07 .. 00 06 E0 03 80 .. 00 4FD
028:04 01 C0 03 C0 .. 00 04 03 C0 01 E0 .. 00 04 07 FBA
029:80 01 E0 .. 00 04 07 80 00 60 .. 00 04 06 00 00 ED9
030:60 .. 00 04 06 00 00 60 .. 00 04 06 00 00 60 1F AB4
031:E0 07 F8 06 00 00 60 3F F0 OF FC 06 00 00 60 3F 6BB
032:F0 OF FC 06 00 00 60 3F F0 OF FC 06 00 00 60 3F 773
033:F0 OF FC 06 00 00 60 1F E0 07 F8 06 00 00 60 .. DC1
034:00 04 06 00 00 60 .. 00 04 06 00 00 60 .. 00 04 CF8
035:06 00 01 E0 .. 00 04 07 80 01 E0 .. 00 04 07 80 FA7
036:03 C0 .. 00 04 03 C0 03 80 .. 00 04 01 C0 07 .. AD1
037:00 06 E0 0E .. 00 06 70 .. 00 11 40 00 26 .. 00 97E
038:18 02 00 00 40 02 00 01 40 01 00 00 40 02 00 00 D92
039:80 00 80 00 40 02 00 01 00 00 60 00 40 06 00 06 EAF
040:00 00 70 00 74 46 00 0E 00 00 38 00 7F D8 00 1C 821
041:00 00 1C 00 7E E8 00 08 00 00 0E 02 79 2F 80 70 319
042:00 00 07 06 3B 04 80 E0 00 00 03 8F BF 36 61 C0 C93
043:00 00 01 EE BB 7F 73 80 00 00 00 9C BB 4F 3F 00 F86
044:00 00 01 69 3F 47 9E 00 00 00 04 DA 7F 67 87 80 8B3
045:00 00 0B B6 FF 73 5C 80 00 00 26 6C FE 31 E6 30 67D
046:00 00 EC 4C E6 39 BE 1C 00 01 91 F9 C4 18 C0 CF C60
047:00 03 C3 63 CD 1E 41 F1 E0 01 1E 43 CE 4E 61 18 924
048:00 07 1E C5 CE 72 61 8C F8 10 19 E5 EE 7B F8 E6 861
049:7C 01 1B 81 CE 58 D8 31 BC 01 33 03 CC 68 CF D8 893
050:98 0F 27 07 CC 28 87 CC A8 00 3F 0F CC 30 40 67 7DC
051:B8 00 00 0F 87 01 00 C0 00 00 00 07 83 81 91 C0 DBD
052:00 00 00 01 C0 C0 3F 80 00 00 00 01 FF 80 0E .. D05
053:00 05 1F .. 00 55 20 00 00 04 00 04 00 00 50 00 A9B
054:50 22 00 0A 00 00 28 00 50 20 00 14 00 00 14 00 7E2
055:51 10 00 28 00 00 0A 00 56 10 00 50 00 00 05 00 E6E
056:2C 03 00 A0 00 00 02 80 28 01 11 40 00 00 01 51 449
057:2C 00 8E 80 00 00 00 63 2C 30 C5 .. 00 04 86 14 567
058:38 6B 80 00 00 03 04 14 7C 60 00 00 04 08 14 746
059:0C 20 30 00 00 18 10 14 0E 18 0C 00 00 10 30 18 832
060:06 00 03 00 00 60 00 38 07 00 00 80 00 00 80 30 8C2
061:01 80 00 00 00 01 80 30 01 80 E0 00 00 01 00 30 6A3
062:01 80 70 18 00 06 00 10 00 00 18 2C 02 04 00 30 4F3
063:20 20 0E 3C 06 0C 00 30 10 30 07 38 00 18 06 30 ECE
064:10 18 03 10 00 00 0B 30 00 0F 80 00 00 00 0F 78 F51
065:00 0F .. 00 04 06 7C 00 0E .. 00 05 3F .. 00 5F 40B
066:03 C0 .. 00 04 20 00 2F 04 00 00 04 00 50 00 5F 46C
067:FA 00 0A 00 00 28 00 5F FE 00 14 00 00 14 00 DF DBB
068:FF 00 28 00 00 0A 03 DF FC 00 50 00 00 05 07 EF 436
069:FF E0 A0 00 00 02 8F EF FF F1 40 00 00 01 7F EF BE7
070:FF FE 80 00 00 00 FF EF FF FD 00 00 00 01 FF F7 045
071:FF FB 80 00 00 07 FF FF FF FE E0 00 00 0F FF F7 105
072:FF FF F0 00 00 3F FF FF FF FF FC 00 00 .. FF 06 866
073:00 01 .. FF 06 80 07 .. FF 06 E0 0F .. FF 06 F0 0BA
074:07 .. FF 06 F8 03 .. FF 06 EC 43 .. FF 06 FE E7 7E2
075:.. FF 0C 9F .. FF 04 FF FF FF 0F .. FF 07 0F .. B93
076:FF 07 9F FF FF 7F .. FF 06 FE 7F .. FF 06 FE 3F F92
077:.. FF 06 FC 03 .. FF 06 C0 00 .. FF 06 00 00 00 E5F
078:07 FF FF E0 .. 00 0B 16 00 16 .. 00 05 78 00 00 000
079:00 84 00 00 01 32 00 00 02 79 00 00 04 CC 80 00 A9F
080:09 86 40 00 13 33 20 00 26 79 90 00 4C FC C8 00 71F
081:59 FE 68 00 59 FE 68 00 4C FC C8 00 26 79 90 00 F55
082:13 33 20 00 09 86 40 00 04 CC 80 00 02 79 00 00 DEA
083:01 32 00 00 00 84 00 00 00 78 .. 00 0B 78 00 00 65D
084:00 FC 00 00 01 CE 00 00 03 87 00 00 07 33 80 00 EAA
085:0E 79 C0 00 1C FC E0 00 39 FE 70 00 73 FF 38 00 898
086:67 FF 98 00 67 FF 98 00 39 FE 38 00 39 FE 70 00 933
087:1C FC E0 00 0E 79 C0 00 07 33 80 00 03 87 00 00 2B4
088:01 CE 00 00 00 FC 00 00 00 78 .. 00 2B 30 00 00 91B
089:00 78 00 00 00 78 00 00 00 30 .. 00 27 16 00 16 537
090:.. 00 09 54 00 00 01 7F 00 00 05 93 00 00 06 25 910
091:40 00 14 A8 C0 00 1A F5 70 00 39 FA E0 00 15 FD 80F
092:30 00 32 FE 60 00 15 FD 30 00 38 FE E0 00 1B 7C 67A
093:30 00 14 A9 60 00 1A 80 E0 00 07 29 80 00 06 B6 496
094:80 00 01 FE 00 00 01 54 .. 00 17 80 00 00 02 6C C3E
095:00 00 01 DA 80 00 0B 57 00 00 05 0A 80 00 06 05 964
096:00 00 0A 02 C0 00 0D 01 80 00 0A 02 C0 00 07 01 8EB
097:00 00 04 83 C0 00 0B 56 80 00 05 7F 00 00 00 06 C6C
098:00 00 01 49 .. 00 16 01 FE 00 00 07 FF 80 00 0D D6D
099:FF C0 00 1F FF E0 00 3F FF F0 00 3F 57 F0 00 7F F76
100:0B F8 00 7E 05 F8 00 7E 02 F8 00 7D 01 F8 00 7E C79
101:02 F8 00 7F 01 F8 00 7E 83 F8 00 7F 57 F8 00 3F F55
102:FF F0 00 3F FF F0 00 1F FF E0 00 0F FF C0 00 07 D90
103:FF 80 00 01 FE .. 00 07 2E 00 2C .. 00 18 06 .. 618
104:00 05 07 .. 00 05 03 83 00 00 C2 00 02 E7 00 80 11A
105:C6 00 03 3F E1 CF FE 00 03 4F E3 FF EE 00 03 0C 6E0
106:27 FD CC 00 01 00 1C 20 44 00 01 18 18 00 CC 00 365
107:01 B3 F0 5C 8C 00 03 37 F7 9F 8E 00 02 5F F7 9F 22F
108:CE 00 03 3F 1F 8F C7 80 03 1E FF E7 E0 C0 07 1D 149
109:FF E3 FC 70 06 11 FF F3 F8 70 18 00 FF F1 FE E0 396
110:30 79 FF F9 F6 E0 30 01 FF FD E4 E0 30 04 7F FF 8D0
111:C3 80 38 06 7F FF 87 00 0C 13 3F FF CC 00 0E 0F B2C
112:9F FE 66 00 07 1F CF AC 43 00 03 1F E3 60 C3 80 D5D
113:00 1F B8 D1 87 80 02 1F C8 77 1F C8 06 33 C7 EC 652
114:70 00 06 03 C7 E0 70 00 06 01 C3 E2 70 00 04 77 11B
115:83 C0 30 00 00 F3 07 C0 38 00 0F D1 01 CF 18 00 759
116:1E 18 10 D8 E8 00 00 1C F8 38 3C 00 00 0F FC 30 9EB
117:0F 00 00 0F EE F0 00 00 00 07 07 E0 .. 00 38 01 19D

```

```

118:.. 00 06 C0 .. 00 05 B0 00 00 10 00 00 B3 C0 02 107
119:30 00 00 80 63 DF 88 00 00 87 E7 FF 30 00 00 0C E11
120:0F A3 70 00 00 08 08 60 70 00 00 20 08 60 30 00 791
121:00 C0 00 70 38 00 00 E0 00 18 1F 00 00 E2 00 1C 8AF
122:03 80 01 EE 00 0C 07 80 07 FF 00 0E 01 00 0F 86 D84
123:00 06 09 00 0F FE 00 02 1B 00 0F FB 80 00 3C 00 DAB
124:07 F9 80 00 78 00 03 EC C0 00 30 00 01 F0 60 01 9B7
125:98 00 00 E0 30 53 9C 00 00 E0 1C 9F 1C 00 01 E0 8E7
126:47 2E 38 00 01 E0 37 88 20 00 01 CC 38 10 00 00 86A
127:01 FC 38 19 80 00 01 FE 3C 19 80 00 03 88 7C 3F 474
128:C0 00 07 0C F8 3F C0 00 00 0E FE 30 E0 00 07 808
129:EF 20 10 00 00 03 07 C0 .. 00 04 03 C0 .. 00 04 4F3
130:01 .. 00 15 18 .. 00 05 1C .. 00 04 0E .. 00 407
131:04 C0 0F 80 00 00 01 C0 0F CF C0 00 C3 80 0F FF A75
132:C0 13 F3 80 0F FF E3 F7 FF 80 0F FF E7 FF FF 00 0F4
133:07 .. FF 04 00 07 BF FF FF FF 00 07 .. FF 04 00 375
134:0F FF FF FF BF 00 0F FF FF FF FF 80 0F CF 0F A3 999
135:7F 80 0F E8 08 70 7F 80 0F E0 F8 78 3F 80 1F C1 8A4
136:F0 7E 3F F0 1F E1 C0 1E 1F FC 7F E3 00 1F 03 FC BDD
137:7F EF 00 0F 07 FC FF FF 00 0F 01 FC FF 86 00 07 37F
138:09 F8 FF FE 00 03 1B E0 FF FF 80 00 3F C0 7F FF 819
139:80 00 7F 80 3F EF C0 00 3F 00 3F F3 E0 01 9F 80 1F8
140:1F E1 F0 73 FF C0 0F E0 FC FF FF E0 07 E0 7F EF 7A3
141:FF E0 0F E0 3F 8F FF E0 1F CC 38 1F FF C0 1F FC F5F
142:38 1F FC 00 1F FE 3C 1F FC 00 1F F8 7C 3F F8 00 CA2
143:1F FC 8F 3F C0 1F FE FE 3F FE 00 3F 7F FF 3F 156
144:FF 00 3C 7F FF FC 0F 00 7F 7F FF FC 7F 80 00 3F A6E
145:FF FC 1F 80 00 1F FF F8 03 C0 00 1F CF F8 00 00 5BA
146:00 0E 07 E0 00 00 00 20 00 20 40 0F 02 A0 70 472
147:0E 05 51 E8 17 8A 2B D0 0B D4 17 A8 15 E8 0F 50 E19
148:0A F0 1E A8 15 78 3D 50 0A BC 3E A0 05 7C 7F 50 994
149:0A FE 7F A0 05 FE 79 F0 0F 9E F8 78 1E 1F F4 1C A8C
150:38 2F FA 04 20 5F F5 00 00 BF FA A1 85 4F F5 52 4D1
151:4A AF FA A2 45 5F FD 51 8A BF 7A A0 05 5E 7D 5F 174
152:FA BE 7E FF FF 7E 3D F3 CF BC 3E A0 05 7C 1F 50 EEA
153:0A F8 0F A8 15 F0 17 D4 2B E8 2B E8 17 D4 51 F0 1DE
154:0F 8A A0 78 1E 05 40 0F F0 02 80 00 30 01 40 08 BA6
155:1E 02 20 17 FF 84 10 28 1F C8 08 57 FF F0 00 AF 001
156:FF F0 01 57 FF F8 02 AF FF FC 01 50 0F FC 00 AF A5E
157:FF 8E 00 50 0E 1E 06 0F F8 7E 07 87 E1 FF 0B E3 839
158:07 FF 05 9B D9 FF 0A FE 7F FF 05 5C 3F FF 0A A9 7D1
159:9F FF 05 59 9F FF 02 AC 3F FF 05 5E 7F FE 02 A0 63D
160:07 FE 01 00 01 FE 02 33 CC FC 01 53 CF FC 00 AB AE4
161:DF F8 00 57 FF F0 08 2B FF F0 10 10 1F C8 20 0F BF2
162:FF 84 40 00 1E 02 80 00 30 01 C0 03 C0 03 F0 07 52F
163:E0 F0 78 07 E0 1E 7C 07 E0 3E 38 00 00 1C 10 00 104
164:00 08 .. 00 09 0F F0 00 0E 0F F0 70 07 8F F1 E0 580
165:07 E0 07 E0 07 F8 1F E0 03 FC 3F C0 01 FC 3F 80 EA5
166:00 7D BE 00 00 03 C0 00 00 07 E0 00 00 07 E0 00 987
167:00 03 C0 00 00 01 80 00 00 1F F8 00 00 7F FE 00 AC7
168:00 FF FF 00 00 0C 30 00 00 04 20 00 10 00 00 08 60C
169:38 00 00 1C 7C 07 E0 3E 78 07 E0 1E F0 07 E0 0F BDD
170:C0 03 C0 03 00 20 00 20 00 01 00 00 00 2C 80 00 42A
171:00 7F 50 00 01 DF F4 00 03 6A 8A 00 0E 81 DB 00 EDO
172:0C 0D 7D 00 2A E2 BC 00 39 71 2F 40 35 F2 FD 08 6CA
173:78 E8 BE 40 32 43 7D 80 78 04 D7 00 4E 2B FD 80 7AA
174:35 7F F6 40 7E D7 7C 80 DD FD FE 50 3F FF F2 00 FA2
175:2F BF 58 80 1D F7 FA 00 15 BA C2 00 23 6D 80 40 EB1
176:15 D9 44 00 00 0D 20 00 02 24 00 00 00 09 10 .. EEO
177:00 06 10 .. 00 1F 20 00 00 00 95 00 00 01 7E 20 C15
178:00 03 F2 80 00 05 1D 40 00 06 8E D0 00 0A D0 00 C1D
179:00 07 17 40 00 0D BC 80 00 07 FB 20 00 11 04 00 CEB
180:00 0A 80 00 00 01 28 80 00 02 02 .. 00 07 40 .. 2D3
181:00 37 0F F0 00 00 7F FE 00 01 FF FF 80 03 FF FF 077
182:C0 07 FF FF E0 0F FF FF F0 1F F7 FF F8 3F 1F FF 6B2
183:FC 3E 8F FF FC 7A 0F FF FE 7F 17 FF FE 7D BF FF A11
184:FE .. FF 20 7F FF FF FE 7F FF FF FE 7F FF FF FE 43A
185:3F FF FF FC 3F FF FF FC 1F FF FF FF 8F FF FF F0 918
186:07 FF FF E0 03 FF FF C0 01 FF FF 80 00 7F FE 00 13F
187:00 0F F0 00 00 18 00 18 00 E7 00 00 03 A5 C0 00 3C4
188:0F 52 F0 00 1E A5 78 00 3F 42 FC 00 3A B1 5C 00 879
189:77 FF AE 00 7B C1 DE 00 ED D4 B7 00 FB E1 5F 00 BF1
190:FD D4 BF 00 EB FC 57 00 FD E4 BF 00 FB C1 5F 00 A5E
191:ED D0 B7 00 FB C5 DF 00 7F FF BE 00 7A 81 5E 00 06C
192:3F 52 FC 00 3D A5 BC 00 1F 42 F8 00 0F E7 F0 00 A90
193:03 D3 C0 00 00 E7 00 00 00 1F 00 00 00 5F C0 00 7EB
194:00 AF F0 00 01 5F F8 00 00 BF FC 00 05 4F FC 00 795
195:0B FF FE 00 05 FF FE 00 13 EF FF 00 05 0F FF 00 DAB
196:03 EF FF 00 15 FF FF 00 03 DF FF 00 05 0F FF 00 3C5
197:13 EF FF 00 05 FF FF 00 03 FF FE 00 05 7F FE 00 A2B
198:00 AF FC 00 02 5F FC 00 00 BF F8 00 00 1F F0 00 66E
199:00 2F C0 00 00 1F .. 00 1A 03 FF 80 00 01 C1 80 27E
200:00 01 C0 80 00 01 C0 00 00 01 C4 00 00 01 FC 00 D5A
201:00 01 C4 00 00 01 C0 00 00 01 C0 80 00 01 C1 80 47D
202:00 03 FF 80 .. 00 1E 30 00 00 2F 00 00 0C 30 00 892
203:00 01 F8 9F 80 00 00 06 D4 2F 60 00 00 0B A8 95 950
204:00 00 00 11 68 2E 88 00 00 7E D0 17 7E 00 00 F7 B7D
205:0D 0B EF 00 1D FE AF D5 7F B8 33 BD 4C 62 AF CC E96
206:2F EA AB A5 5B F4 27 FD 4A B2 BF E4 2F 6D 8F E1 5F2
207:6D F4 3F FD 17 E8 BF FC 1F BA 87 C1 5D 8F 3F ED B3A
208:43 82 87 FC 3D FA 14 49 5D BC 3F D5 40 02 AF FC C08
209:3F 6A 91 09 56 FC 7D FD 44 22 BF BE 7F DA A8 89 0E4
210:6D FE 00 16 38 00 00 41 FC 00 00 3F 82 01 FC 91E
211:00 00 3F 80 01 FC 00 00 3F 80 41 FC 00 00 3F 82 ED3
212:01 FC 00 00 3F 80 00 00 16 38 00 00 7F DA A9 89 BB5
213:6D FE 7F FF 45 22 FF FE 3F 6A 91 09 56 FC 3F D7 7B3
214:41 02 EF FC 3D FA 15 49 5D BC 3F EF 43 82 F7 FC 111

```



```

215:1F BB 87 C1 DD F8 1F FD 17 E8 BF F8 2F D7 8F E1 A9E
216:ED F4 27 FD 4A B2 BF E4 2F EA AB A5 5B F4 33 BD OFE
217:CC 63 AF CC 1D FE EF D7 7F B8 00 FF D1 0B FF 00 34B
218:00 7E A3 45 7E 00 00 3F F5 2F FC 00 00 1F B9 9D 8BC
219:F8 00 00 07 05 2F E0 00 00 01 FD BF 80 00 00 9D0
220:3F FC .. 00 04 13 F8 .. 00 04 07 FF 80 00 00 01 7B1
221:2B FF E0 00 00 08 57 FF 90 00 00 10 17 00 08 00 FOE
222:00 00 2F E8 3E 00 00 08 2F FF FF 00 1C 01 57 FF OFF
223:FF B8 32 42 BC 7F FF CC 23 15 58 3F FF C4 2B 82 1AD
224:B9 3F FF D4 27 A9 7C 7F FF E4 3F 82 EF FF FF FC 1E8
225:07 C5 7F FF FF F8 00 12 BF FF FF FC 02 05 EB FF FFE
226:FF FC 00 2A BF FF FF FC 00 95 6E FF FF FC 02 02 7E5
227:BB FF FF FE 00 25 57 FF FF FE 00 00 19 F8 00 00 FD9
228:40 01 FF FF 80 02 00 01 FF FF 80 00 00 01 FF FF EDO
229:80 00 40 01 FF FF 80 02 00 01 FF FF 80 00 00 00 DBA
230:19 F8 00 00 22 27 57 FF FF FE 22 02 BB FF FF FE C7B
231:22 97 6F FF FC 12 2A BF FF FF FC 13 07 EB FF CF3
232:FF FC 11 12 BF FF FF FC 0F C5 7F FF FF F8 1F 83 E9B
233:EF FF FF F8 27 A9 7C 7F FF E4 2B C3 B9 3F FF D4 044
234:23 55 D8 3F FF C4 33 62 BC 7F FF CC 1C 91 57 2F5
235:FF B8 00 48 6F FF FF 00 00 25 7D FF FE 00 00 12 D3D
236:2B FF FC 00 00 00 57 FF F8 00 00 03 FB FF E0 00 0F8
237:00 00 EF FF 80 00 00 00 3F FC .. 00 15 16 00 00 176
238:68 00 00 2F FF FF F4 00 00 03 FF FF C0 .. 00 07 9C8
239:1C 00 07 C0 00 38 3E 00 0F E0 00 7C 3F 00 0F E0 6EF
240:00 FC 3F 80 0F E0 01 FC 3F 80 0F E0 01 FC 3F 80 E99
241:07 C0 01 FC 07 80 07 C0 01 E0 00 00 03 80 .. 00 613
242:26 1D FD FE FF BF B8 5D FD FE FF BF BA 5D FD FE 9A4
243:FF BF BA 1D FD FE FF BF B8 5D FD FE FF BF BA .. 9D3
244:00 06 22 02 01 00 40 4A 22 02 01 00 40 40 22 02 08F
245:01 00 00 04 12 02 01 00 40 40 11 02 01 00 40 08 C39
246:11 02 03 80 00 80 0F 81 07 C0 81 F0 1F 81 07 C0 610
247:81 F8 3F 81 0F E0 01 FC 3F C1 0F E0 81 FC 3F 40 704
248:8F E1 02 FC 3F 20 8F E0 00 FC 1C 90 47 C2 08 38 FD1
249:00 48 41 02 02 00 00 24 21 00 20 00 00 12 21 04 A47
250:08 00 00 0D 11 08 A0 00 00 03 D1 02 40 00 00 00 50E
251:ED 35 .. 00 04 3F FC 00 00 00 20 20 00 00 70 38F
252:00 00 C2 44 00 07 FB 6A 70 07 FE 14 70 1D 7F 73 DDB
253:A0 1D DF 1F 70 1D 05 B5 D8 1E 10 02 78 7E 33 1E 651
254:FC 7F F3 6E 76 7F DD 1C 76 FF 2C 1E BC FF AE 6E C58
255:FF FF 5F DE 68 FE 1F ED 78 FF 9F 57 1F FF 9F 6F C8A
256:BF FD BE 57 89 FE BE 3D 3F 7C 8A FB 1F FC 17 728
257:EE 1E D5 A1 1A 3B 59 0F 7E 7A 15 C5 4E 78 B4 43 9DE
258:DE 3F AA E0 EE 3F CF 1C BC 3D 8F 2C 40 06 DE 5F 6FF
259:C0 01 DE 0F C0 00 98 00 C0 00 30 04 00 00 00 10 EC9
260:00 00 03 BC 00 00 03 97 F0 00 03 EB F0 02 BF CF EFA
261:E0 02 7F EB F0 02 FB EF F8 01 EF FF F8 01 CC FD B5D
262:FC 00 0C FD FE 01 22 FF FE 19 D3 FD FC 1B 7D FD AAB
263:F8 1A BC FD F8 19 FC 3E FF 10 7C FF FF 10 7C FF 15E
264:7F 02 71 FF 7F 01 71 F6 FF 00 F3 7F FF 00 63 FF A57
265:FE 01 6A 7F FE 00 A6 FF FE 05 EE 3F FE 07 4F BF OE1
266:FE 00 57 1F FE 00 3F FF FC 02 FF D3 E0 01 3F A9 2BA
267:C0 00 3E 73 C0 00 7C 7F C0 00 10 3C .. 00 82 20 E02
268:00 20 00 0F 00 00 00 74 D0 00 01 89 EE 00 02 14 D96
269:97 80 04 2F 49 C0 08 EF DE E0 10 D2 B4 70 23 41 238
270:D4 38 22 B2 BA 18 4E B9 3E 0C 45 BA FA 2C 4E B0 EDF
271:B6 0C 8D 03 78 06 93 05 FD 06 8D 3B FB 06 9F 6F 237
272:6D 06 97 3D FB 46 8F 7F A5 06 8B 57 51 06 85 5E 274
273:C5 06 49 B5 02 0C 45 8A 4A 0C 40 A1 22 0C 20 94 8D5
274:02 18 20 40 04 38 10 04 04 70 08 20 08 E0 04 20 711
275:09 C0 02 10 17 80 01 89 EE 00 00 74 D0 00 00 0F C98
276:.. 00 04 F0 00 00 0B FE 00 00 77 FF 80 01 E8 1F 310
277:C0 03 90 0D E0 07 10 0C F0 0E 2D 06 78 1C BE 26 859
278:3C 19 4D 43 1C 31 46 C3 0E 32 45 03 0E 31 4F 43 618
279:0E 62 FC 81 87 64 FA 01 87 62 C4 01 87 60 90 81 FAE
280:87 60 C2 01 87 60 80 01 87 60 A0 01 87 60 80 01 87A
281:87 30 40 03 0E 30 40 03 0E 30 40 03 0E 18 40 03 6AC
282:1C 1C 20 06 3C 0E 20 06 78 07 10 0C F0 03 90 0D A9A
283:E0 01 E8 1F C0 00 77 FF 80 00 0B FE 00 00 0F A58
284:.. 00 0E 07 E0 00 00 4F F2 00 00 CF F3 00 01 9F 78B
285:FC 80 03 9F F9 C0 07 0F FC E0 0F 07 FC F0 0E 07 60C
286:F9 F0 0F 0F F9 C0 1E 7F FE 78 1E 7F FE 78 1E 7F CBA
287:FE 78 1E 7F FE 78 1E 7F FE 78 1E 7F FE 78 1E 7F B6B
288:FE 78 1E 7F FE 78 0F 3F FC F0 0F 3F FC F0 0F 3F 609
289:FC F0 07 3F FC E0 03 9F F9 C0 01 9F F9 80 00 CF 4F2
290:F3 00 00 4F F2 00 00 07 E0 .. 00 0E 18 00 18 00 35D
291:F8 00 00 03 A5 80 00 0F 52 C0 00 1E A5 70 00 3F 375
292:42 F8 00 3A 2C 58 00 77 C2 0C 00 76 95 4C 00 FD 1DB
293:00 B2 00 FA A1 5C 00 FD 00 3C 00 FF E5 00 00 FD 2A2
294:04 BE 00 FA 81 5C 00 ED 10 34 00 FE 81 72 00 7D 709
295:42 AC 00 7B 89 1C 00 3F 30 F8 00 3D A5 B8 00 0F 0E9
296:52 F0 00 0F E7 C0 00 02 CB 00 00 00 C6 .. 00 07 342
297:5F 00 00 00 AF C0 00 01 5F F0 00 00 BF F8 00 05 475
298:80 F8 00 08 3F 1C 00 01 6F EC 00 02 FF F2 00 05 21D
299:5F FC 00 02 FF FC .. 00 05 02 FF FE 00 05 7F FC 74B
300:00 12 EF FC 00 01 7F F2 00 02 BF EC 00 04 7F 1C 24F
301:00 00 80 F8 00 02 5F F8 00 00 AF F0 00 00 1F C0 8C5
302:00 00 3F .. 00 07 FF 00 00 03 00 C0 00 0C 00 30 BA1
303:00 10 00 08 00 20 04 00 20 7F 04 00 43 80 E2 251
304:00 4C 00 12 00 90 00 0D 00 A0 00 03 00 C0 00 03 C9D
305:00 FF FF FF 00 80 00 01 00 C0 00 03 00 A0 00 03 DE0
306:00 90 00 00 00 4C 00 12 00 43 80 E2 00 20 7F 04 7E7
307:00 20 00 04 00 10 08 00 0C 00 30 00 03 00 C0 286
308:00 00 FF 00 00 00 31 00 21 01 .. 00 04 40 00 00 928
309:03 00 05 70 00 60 00 00 06 00 3E 5E 00 30 00 00 55C

```

```

310:0C 40 ED 2D 81 18 00 00 18 C1 F6 5F C1 8C 00 00 83D
311:31 C1 DA 0B C1 C6 00 00 67 84 34 26 10 F3 00 00 C00
312:CA CF 58 0D 79 A9 80 00 07 7F F4 14 BF 75 80 00 E18
313:DA FB 6C 53 FF AD 80 00 07 7F A9 05 6F 75 80 00 33A
314:CA FD 74 0A 7F A9 80 00 CF FF EB E5 BF F9 80 00 92B
315:CC FF 9F FB 57 99 80 00 CF ED 7E 39 BF F9 80 00 DA0
316:C0 7F DC 1E FF 01 80 00 DF FF 6C 1B B7 FD 80 00 34C
317:C0 7B BC 1D 7F 01 80 00 CF FF 6E 3A EF F9 80 00 EAA
318:CC FF 9F FB BF 99 80 00 CF FE EB E8 FF F9 80 00 CD6
319:CA FB BA 05 DF A9 80 00 D7 7F D5 53 FF 75 80 00 238
320:DA FF EA 07 FF AD 80 00 D7 7F FC 4B FF 75 80 00 7AB
321:CA CF F9 17 F9 A9 80 00 67 87 F4 1F F0 F3 00 00 D61
322:31 C3 FA 2E E1 C6 00 00 18 C1 EE 47 C1 8C 00 00 228
323:0C 40 FD 0F 81 18 00 00 06 00 3E 3E 00 30 00 00 F41
324:03 00 07 30 00 60 00 00 01 .. 00 04 40 .. 00 0B B7E
325:80 02 F0 00 80 00 00 01 80 01 FE 00 C0 00 00 03 183
326:C0 12 FF 81 E0 00 00 07 C0 09 FF C1 F0 00 00 0F A61
327:C0 25 FF C1 F8 00 00 01 F8 0B FC 10 FC 00 00 0F 8A1
328:C0 07 F0 79 8E 00 00 30 60 0B EB FF 06 00 00 35 DBB
329:64 93 FF FF 56 00 00 30 40 56 FF FF 06 00 00 38 2E7
330:C2 8B FF FF 8E 00 00 3F C0 17 9F FF FE 00 00 3C 0EB
331:80 67 8F FF 9E 00 00 3F 92 8F 47 FF FE 00 00 3F 985
332:80 2F E7 FF FE 00 00 20 00 9D C7 FF 02 00 00 3F D2F
333:84 4F E7 FF FE 00 00 3F 80 9F 47 FF FE 00 00 3C 542
334:80 67 8F FF 9E 00 00 3F C1 17 9F FF FE 00 00 38 E2E
335:C4 45 FF FF 8E 00 00 30 43 2A BF FF 06 00 00 35 3A7
336:67 95 FF FF 56 00 00 30 63 E3 FF FF 06 00 00 38 83A
337:C1 F6 FF F9 8E 00 00 1F 80 FB FF F0 FC 00 00 0F 6CC
338:C0 25 FF E1 F8 00 00 07 C0 11 FF C1 F0 00 00 03 DCD
339:C0 02 FF 81 E0 00 00 01 80 01 FE 00 C0 00 00 00 52F
340:80 00 F0 00 80 .. 00 33 02 00 00 20 00 00 00 07 1AC
341:03 E0 03 E0 70 00 00 0F C0 FF FF 81 F8 00 00 1D 8A2
342:E0 1F FC 03 DC 00 00 1D E0 00 00 03 DC 00 00 1D 37D
343:C0 00 00 01 DC 00 00 0F C0 00 00 01 F8 00 00 07 2BF
344:C0 00 00 01 F0 00 00 00 80 00 00 00 80 .. 00 2B 42A
345:80 00 00 00 80 00 00 07 C0 00 00 01 F0 00 00 0F 877
346:C0 00 00 01 F8 00 00 1D C3 00 00 61 DC 00 00 1D 7B4
347:E7 80 00 F3 DC 00 00 1D E3 E0 03 E3 DC 00 00 0F C57
348:C1 F0 07 C1 F8 00 00 07 00 F0 07 80 70 .. 00 04 43E
349:20 02 .. 00 2C 00 00 00 00 00 00 00 00 00 00 00 8EF

```

```

*****
*                               AMIGA DOS INFO                               *
*   Programmname   : DSCALC                                                *
*   Funktion       : Wandler für DSGRAPH                                    *
*   Sprache        : AmigaBASIC                                             *
*   Besonderheiten : Mit CheckBas überprüfbar!                             *
*****

000:543 REM
001:271 REM DScalc
002:543 REM
003:728 REM Grafik- und Sounddaten fuer "Double Strike "bere
      chen
004:543 REM
005:000
006:171 DEFINT b-z:DEFLNG a,o
007:580 DIM SHARED x(127),y(127),ofs1(127),ofs2(127),adr,pt
008:813 DIM SHARED gr(127),gr2(1),gr3(1),size
009:217 pi#=3.1415927#
010:045 SCREEN 2,320,200,3,1
011:746 WINDOW 2,"DScalc",,0,2
012:590 LOCATE 10,12:PRINT "Bitte warten..."
013:000
014:883 OPEN "ram:DSdata" FOR OUTPUT AS #1
015:492 GOSUB Font
016:988 GOSUB JGameOver
017:903 GOSUB Bewegung
018:980 OPEN "DSgraph" FOR INPUT AS #2
019:115 CALL calc(1,1)
020:105 CALL calc(0,0)
021:255 CALL calc(0,-1)
022:129 CALL calc(1,8)
023:129 CALL calc(1,8)
024:129 CALL calc(1,8)
025:297 CALL calc(1,16)
026:129 CALL calc(1,8)
027:129 CALL calc(1,8)
028:129 CALL calc(1,8)
029:121 CALL calc(1,4)
030:129 CALL calc(1,8)
031:129 CALL calc(1,8)
032:129 CALL calc(1,8)
033:129 CALL calc(1,8)
034:044 CLOSE #2
035:155 FOR n=0 TO pt-1
036:294 IF ofs1(n)>32767 THEN ofs1(n)=ofs1(n)-65536&
037:830 IF ofs2(n)>32767 THEN ofs2(n)=ofs2(n)-65536&
038:089 PRINT #1,MKI$(x(n));MKI$(y(n));MKI$(ofs1(n));MKI$(
      ofs2(n));
039:598 adr=adr+8
040:398 NEXT n

```


Listings

```

041:635 GOSUB Sounds
042:043 CLOSE #1
043:464 WINDOW CLOSE 2
044:576 SCREEN CLOSE 2
045:500 END
046:000
047:664 REM Verkleinerung der Objekte berechnen
048:645 SUB calc(lflag,num) STATIC
049:503 CLS
050:264 IF lflag=1 THEN
051:083 x1=CVI(INPUT$(2,2))
052:227 y1=CVI(INPUT$(2,2))
053:783 size=INT((x1+15)/16)*y1*3
054:129 ERASE gr:DIM gr(size+2)
055:889 gr(0)=x1:gr(1)=y1:gr(2)=3
056:119 FOR n=0 TO size-1
057:723 gr(n+3)=CVI(INPUT$(2,2))
058:398 NEXT n
059:216 END IF
060:000
061:040 PUT (0,0),gr
062:264 IF num=0 THEN
063:011 FOR y=0 TO y1-1
064:747 FOR x=0 TO x1-1
065:744 IF POINT(x,y)=7 THEN PSET(x,y),4
066:809 NEXT x,y
067:215 GET(0,0)-(x1-1,y1-1),gr:num=1
068:216 END IF
069:040 IF num=-1 THEN
070:011 FOR y=0 TO y1-1
071:747 FOR x=0 TO x1-1
072:907 IF POINT(x,y)>0 THEN PSET(x,y),7
073:809 NEXT x,y
074:222 GET(0,0)-(x1-1,y1-1),gr:num=8
075:216 END IF
076:740 xp=0:yp=64
077:095 FOR nu=1 TO num
078:319 ERASE gr2,gr3:DIM SHARED gr2(1023),gr3(1023)
079:034 LINE (xp,yp)-(xp+191,yp+63),0,bf
080:632 IF nu=1 THEN
081:554 PUT (xp,yp),gr,PSET:br=x1:ho=y1
082:091 ELSE
083:118 vk1=1+(nu-1)/4
084:229 br=x1/vk1:ho=y1/vk1
085:535 FOR h=0 TO ho-1
086:963 FOR b=0 TO br-1
087:139 PSET(xp+b,yp+h),POINT(b*vk1!,h*vk1!)
088:704 NEXT b,h
089:216 END IF
090:000
091:795 GET(xp,yp)-(xp+br+15,yp+ho-1),gr3
092:746 PUT(xp+128,yp),gr3,PSET
093:999 FOR y=yp TO yp+ho-1
094:018 FOR x=xp+128 TO xp+127+br
095:901 IF POINT(x,y)>0 THEN PSET(x,y),1
096:809 NEXT x,y
097:602 GET (xp+128,yp)-(xp+143+br,yp+ho-1),gr2
098:247 size2=INT((br+31)/16)*ho
099:799 size3=size2*3
100:667 x(pt)=br:y(pt)=ho
101:438 ofs1(pt)=adr:adr=adr+size2*2
102:658 ofs2(pt)=adr:adr=adr+size3*2
103:895 pt=pt+1
104:299 FOR n=0 TO size2-1
105:185 PRINT #1,MKI$(gr2(3+n));
106:398 NEXT n
107:303 FOR n=0 TO size3-1
108:313 PRINT #1,MKI$(gr3(3+n));
109:398 NEXT n
110:881 NEXT nu
111:568 END SUB
112:000
113:832 Bewegung:
114:879 FOR n=0 TO 255
115:938 b=INT((n+21)/43):c=((n+21) MOD 43)*5.714
116:293 IF b=0 OR b=2 OR b=4 OR b=6 THEN x=c-120 ELSE x=12
0-c
117:672 y=96-SIN((n MOD 64)/64*pi#)*192
118:999 z=128+SIN((n+64)/128*pi#)*127
119:584 GOSUB WB
120:398 NEXT n
121:879 FOR n=0 TO 255
122:032 x=SIN(n/64*pi#)*120
123:536 y=SIN(n/32*pi#)*96
124:717 z=ABS(255-n*2)
125:584 GOSUB WB
126:398 NEXT n
127:879 FOR n=0 TO 255
128:936 x=SIN(n/32*pi#)*120
129:112 y=SIN(n/25.5*pi#)*56
130:309 z=32:IF n<64 THEN z=255-n*3.5 ELSE IF n>192 THEN z
=32+(n-192)*3.5
131:584 GOSUB WB
132:398 NEXT n
133:879 FOR n=0 TO 255
134:849 x=(n-128)*.937
135:584 y=SIN(n/48*pi#)*96
136:160 z=191+cos(n/48*pi#)*64
137:584 GOSUB WB
138:398 NEXT n
139:879 FOR n=0 TO 255
140:193 x=(n MOD 63)*1.875*SGN(n MOD 128-64)
141:617 y=(n MOD 63)*1.5*SGN(n-128)
142:964 z=255-(n MOD 63)*4

```

```

143:584 GOSUB WB
144:398 NEXT n
145:879 FOR n=0 TO 255
146:870 x=SIN(n/64*pi#)*90+SIN(n/12*pi#)*30
147:160 y=COS(n/64*pi#)*72+cos(n/12*pi#)*24
148:316 z=255-n
149:584 GOSUB WB
150:398 NEXT n
151:879 FOR n=0 TO 255
152:328 x=SIN(n/128*pi#)*120
153:304 y=SIN(n/128*pi#)*96
154:736 z=COS(n/128*pi#)*127+128
155:584 GOSUB WB
156:398 NEXT n
157:879 FOR n=0 TO 255
158:936 x=-SIN(n/128*pi#)*120
159:304 y=SIN(n/128*pi#)*96
160:736 z=COS(n/128*pi#)*127+128
161:584 GOSUB WB
162:398 NEXT n
163:879 FOR n=0 TO 255
164:984 x=COS(n/64*pi#)*120
165:584 y=SIN(n/64*pi#)*96
166:111 z=ABS(128-n)*2:IF z=256 THEN z=255
167:584 GOSUB WB
168:398 NEXT n
169:055 FOR n2=0 TO 1
170:879 FOR n=0 TO 255
171:976 x=SIN((n2+n/64)*pi#)*120
172:520 y=COS((n2+n/64)*pi#)*96
173:316 z=255-n
174:584 GOSUB WB
175:566 NEXT n,n2
176:055 FOR n2=0 TO 1
177:879 FOR n=0 TO 255
178:976 x=SIN((n2+n/56)*pi#)*120
179:024 y=-96+n2*192
180:316 z=255-n
181:584 GOSUB WB
182:566 NEXT n,n2
183:982 RETURN
184:000
185:538 WB:
186:128 IF x<0 THEN x=x+256
187:400 IF y<0 THEN y=y+256
188:869 PRINT #1,CHR$(x);CHR$(y);CHR$(z);
189:593 adr=adr+3
190:982 RETURN
191:000
192:290 Font:
193:767 FOR x=32 TO 127
194:721 LOCATE 1,1:PRINT CHR$(x)
195:024 GET(0,0)-(7,7),gr
196:685 FOR n=0 TO 7
197:600 y=INT(gr(3+n)/256):IF y<0 THEN y=y+256
198:857 PRINT #1,CHR$(y);
199:398 NEXT n
200:408 NEXT x
201:552 adr=adr+768
202:982 RETURN
203:000
204:970 JGameOver:
205:310 FOR n=0 TO 15:PRINT #1,CHR$(16);:NEXT n
206:427 FOR n=0 TO 95
207:929 x=144-SIN(((n+19.2) MOD 38.4)*pi#/38.4)*(128-n)
208:853 PRINT #1,CHR$(x);
209:398 NEXT n
210:630 FOR n=0 TO 15:PRINT #1,CHR$(144);:NEXT n
211:520 adr=adr+128
212:982 RETURN
213:000
214:320 Sounds:
215:288 RANDOMIZE TIMER
216:693 PRINT #1,CHR$(0);
217:814 FOR n=0 TO 4094
218:040 x=(x+32768&/((1024+n)+RND*16-8) MOD 256
219:481 CALL Writ(x)
220:398 NEXT n
221:693 PRINT #1,CHR$(0);
222:011 FOR n=4094 TO 0 STEP -1
223:496 w=w+n:IF w>4093 THEN w=w-4094:w2=RND*256-128
224:928 x=w2*(n+2048)/6142+128
225:481 CALL Writ(x)
226:398 NEXT n
227:445 PRINT #1,CHR$(0);CHR$(0);
228:723 FOR n=2048 TO 0 STEP -1
229:544 x=(RND*256-128)*n/2048+128
230:481 CALL Writ(x)
231:481 CALL Writ(x)
232:398 NEXT n
233:311 w2=0:w=127
234:693 PRINT #1,CHR$(0);
235:395 FOR n=8190 TO 0 STEP -1
236:324 w2=w2+SIN(n/256)+64
237:336 IF w2>1023 THEN w=-w:w2=w2-1024
238:480 x=INT(w*n/8190)+128
239:481 CALL Writ(x)
240:398 NEXT n
241:817 PRINT #1,CHR$(0)
242:982 RETURN
243:000
244:301 SUB Writ(x) STATIC
245:544 b=x-128:IF b<0 THEN b=b+256
246:647 PRINT #1,CHR$(b);:adr=adr+1
247:568 END SUB

```

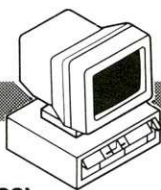



CPS Computertechnik GmbH
Hamburger Str. 283 • 3300 Braunschweig
Fax (05 31) 33 06 61 • BTX *20088 1490#

Ladenöffnungszeiten: Mo-Fr 9-18 Uhr
Sa 9-13 Uhr
Studentenrabatte geg. Nachweis a. Anfrage !

Tel. (05 31) 33 06 63

CPS AT



CPS AT I (siehe Test DOS 1/89)

80286-12 CPU 8/12 MHz. 1 MB RAM,
bis 4MB on board. 0-Wait-State,
Sockel für Co-Prozessor, EMS, Phoenix Bios
64 K ROM mit Setup-Funktion, Uhr u. Kalender,
1 ser./2 par./Game-Port, Herc. komp. Grafikkarte
20 MB Festplatte 38 ms Zugriffszeit,
1 Lw 5,25" 1,2 MB und 1 Lw 3,5" 720KB/1,44MB,
Cherry Tastatur mit sep. Cursorblock,
14" Flat Screen Monitor s/w oder amber,
MS-DOS 4.01 dt. und GW-Basic 3540,-

CPS AT I Mini-Tower

80286-12 CPU 8/12 MHz. 1 MB RAM,
bis 4 MB on board. EMS, 0-Wait-State,
sonst wie AT I 3660,-

CPS AT I Tower

80286-12 CPU 8/12 MHz. 1 MB RAM,
bis 4 MB on board, 0-Wait-State, EMS,
Sockel für Co-Prozessor 80287,
AMI Bios 64 K ROM mit Setup-Funktion,
Uhr u. Kalender, 1 ser./ 2 par./ Game-Port,
Herc. komp. Grafikkarte, 20 MB Festpl. 38 ms,
1 Lw 5,25" 1,2 MB und 1 Lw 3,5" 720KB/1,44 MB
Cherry Tastatur mit sep. Cursorblock,
14" Flat Screen Monitor s/w oder amber,
MS-Dos 4.01dt. und GW-Basic 4175,-

CPS 386 SX

80386 SX CPU 16 MHz, 0-Wait-State,
2 MB RAM, bis 4 MB on board, Sockel für 80387,
Herc. komp. Grafikkarte mit Printerport,
1 ser./ 1 par./ Game-Port,
1Lw 5,25" 1,2 MB und 1 Lw 3,5" 720KB/1,44 MB,
40 MB Festplatte 28 ms Zugriffszeit,
Cherry Tastatur mit sep. Cursorblock,
14" Flat Screen Monitor s/w oder amber,
MS-DOS 4.01 dt. und GW-Basic 4175,-

CPS AT III/ Cache

80386-25 CPU 25 MHz, 2 MB RAM,
bis 16 MB on board, 0-Wait, Sockel für 80387,
2 par./2 ser./1 Gameport, Memory Expansion
Card, EMS, 32 KB Cache, 1Lw 5,25" 1,2 MB,
1Lw 3,5" 1,44 MB, 40 MB Festplatte 28 ms,
Cherry Tastatur, Tower-Gehäuse,
14" Flat Screen Monitor, Mouse,
MS-DOS 4.01 dt. und GW-Basic 7990,-

Technische Änderungen vorbehalten.
Gerätekonfigurationen nach Ihren Wünschen
veränderbar.

COMMODORE

PC 10 III, 1 Laufwerk 360 K, Festplatte 20 MB,
EGA Grafikkarte, EGA Colormonitor 14" Quadram,
MS-DOS 3.3 / GW-Basic,
Drucker Panasonic KX-P 1081 2999,-

COMMODORE

AMIGA 500 795,-
AMIGA 500 + 1084 1399,-
AMIGA 2000 1985,-
AMIGA 2000 + 20 MB Harddisk Autoboot
inkl. Contr. 2090 A 2990,-
AMIGA 2000 + Monitor 1084 2480,-
68020 Prozessorkarte Preis a. Anfr.
68030 Prozessorkarte Preis a. Anfr.

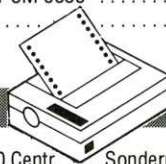
ERWEITERUNGEN

20 MB Harddisk A 2000
inkl. 2090 A Contr. 1220,-
2 MB Erw. int. (A 2000) orig. CBM 1180,-
PC/XT Karte inkl. 5,25" Lw
+ MS-DOS + GW-Basic 899,-
PC/AT Karte inkl. 5,25" Lw
+ MS-DOS + GW-Basic 2400,-
512 KB Erw. A 500 330,-

AMIGA-ZUBEHÖR

LW ext. 3,5" ohne Display 290,-
LW ext. 5,25" ohne Display 419,-
LW intern. 3,5" inkl. Einbausatz 220,-
Commodore 1084 630,-
Philips RGB Color CM 8833 630,-
EGA Standard 815,-

DRUCKER

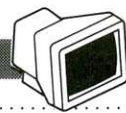


OKI Microline 390 Centr. Sonderpreis a. Anfr.
OKI Microline 393 Centr. Sonderpreis a. Anfr.
OKI Microline 320 Sonderpreis a. Anfr.
STAR LC 10 Centr. 499,-
Epson LX 400 499,-
NEC P 6 Plus Centr. 1499,-
NEC P 7 Plus Centr. 1985,-
Centronics Drucker kabel 18,80

Weitere Drucker auf Anfrage
Wir liefern nur mit dt. Handbuch, Seriennummer und
Herstellergarantie !!! Drucker-Grauimporte mit engl.
Handbuch, ohne Seriennummer, ohne Herstellergarantie
sind bei uns ausgeschlossen.

Nutzen Sie unseren Bequem-Kauf-Kredit!

MONITORE



EGA Standard 815,-
EIZO 9060s 1855,-
NEC MULTISYNC 2 A 1550,-
TTL 14" Flat Screen m. Fuß sw/amber 235,-
TTL 12" Flat Screen m. Fuß sw/amber 180,-
Mitsubishi 1481 1398,-
weitere Monitore auf Anfrage

DISKETTEN



NN 2DD 3,5" 10 Stck. 15,-
NN 2DD 5,25" 10 Stck. 6,90
Magix MF 2DD 3,5" 24,50
Select MF 2DD 3,5" 33,50
Select MD 2DD 5,25" 19,-
Fuji MD 2DD 5,25" 27,-

FARBEBÄNDER



STAR NL/NG/ND/NR-10, Stück 12,-
EPSON LX-800/LQ-500, Stück 8,-
PANASONIC KX-P, Stück 13,-
OKI ML 320 14,50
OKI ML 390 16,85
NEC 2200 13,50
NEC P6+/P7+ 15,30
Star LC 10 10,-

Sämtliche Angebote freibleibend, Zwischenverkauf
vorbehalten. Wir liefern an Nicht-Kaufleute nur per
UPS-Nachnahme. Ins Ausland nur per Vorkasse.
Fordern Sie unseren Gesamtkatalog gegen 3,-DM
in Briefmarken. Es gelten unsere allgemeinen
Geschäftsbedingungen.

Versand nur über Braunschweig

"Wer pfiffig rechnet und jetzt
eine EDV-Anschaffung plant,
kann bei CPS eine Menge
Geld sparen und hat
3 weitere dicke Vorteile."

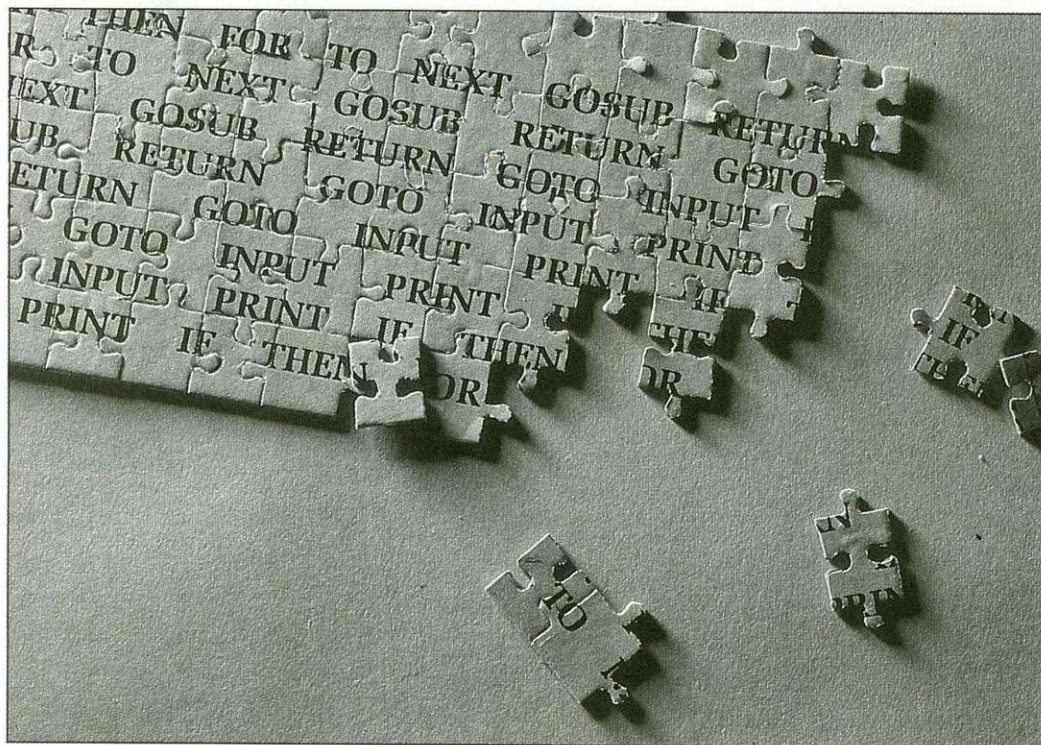


CPS Computertechnik GmbH
Braunhirschstr. 29 • 3100 Celle
Tel. 0 51 41/ 3 20 04 • Fax 0 51 41/ 38 14 22

CPS Computertechnik GmbH
Großbeerenstr. 5 • D-1000 Berlin 42
Telefon 0 30/ 7 05 30 55

**CPS weil Preis &
Leistung stimmen !**

✕ 24 Monate Garantie auf CPS-Rechner
✕ 10 Tage Rückgaberecht auf Hardware
✕ Eigener Reparatur-Service



AmigaBASIC

Programmieren mit Erfolg

Wenn Sie gerade einen Amiga erstanden haben und nun auf die Programmierung gespannt sind, dann sind Sie hier richtig. Wenn Sie Ihren Amiga bis jetzt nur für Anwendungen und Spiele genutzt haben und jetzt mal ein bißchen in das BASIC hineinschnuppern wollen, dann sind Sie hier ebenfalls richtig. Und wenn Sie wissen wollen, wie man sich unter AmigaBASIC eine Telefondatei mit Grafik und Sprachausgabe programmiert, dann sind Sie erst recht hier richtig. Denn dies ist das Ziel unseres BASIC-Kurses für Einsteiger.

Kursfahrplan:



Teil 1: BASIC zum Kennenlernen

Teil 2: Dem Spaghetti-Code keine Chance

Teil 3: Vom Strich zum Bild

Teil 4: Dateien über Dateien

Teil 5: Unter dem Programm: ein Unterprogramm

Beim Auspacken seines neuen Amiga findet der stolze Computerbesitzer außer der äußerst wichtigen Workbench noch eine zweite Diskette, die leider allzuoft und allzuschnell in der Versenkung verschwindet: die Amiga-EXTRAS. Auf dieser Diskette befindet sich auch der BASIC-Interpreter, der für Einsteiger wohl den ersten Kontakt mit

dem 'programmierbaren' Computer Amiga darstellt. Leider führt das Amiga-BASIC ein Mauerblümchendasein, da die meisten Programme in den effektiveren Sprachen C, Modula oder Assembler geschrieben werden. Als Einsteiger steht man allerdings diesen Sprachen ähnlich gegenüber wie der berühmte Ochse dem Klavier (was keine Beleidigung sein

soll, sondern ein Ausdruck der Hilflosigkeit, die einen als Einsteiger angesichts der unbekannten Sprachelemente eines dieser Programmiersysteme beschleicht).

Zu allem Überfluß wird dem Anfänger, der gerade mal in das BASIC reingeschnuppert hat, alles vermiebt, wenn vermeintliche Fachleute beim Nennen des Begriffes BASIC ein hämisches Lächeln sehen lassen und mit Pseudo-Fachbegriffen um sich schmeißen, ohne zu bemerken, wie verwirrend das alles auf den wirken muß, der gerade erst lernt.

An dieser Stelle wollen wir Ihnen weiterhelfen. Unser Kursus wird Sie zwar nicht zu einem ausgefuchtesten Systemkennner machen, dafür aber werden Sie die Grundlagen der BASIC-Programmierung kennenlernen und die wichtigsten Funktionen des Inter-

preters und des Editors spielend anwenden. Was wir voraussetzen, ist, daß Sie sich im BASIC schon einmal umgeschaut haben und daß Ihnen die gebräuchlichsten Befehle wie PRINT, INPUT, FOR/NEXT, IF/THEN bekannt sind. Sollten Sie hier noch Schwierigkeiten haben, dann ziehen Sie auf jeden Fall zu unserem Kurs das BASIC-Handbuch Ihres Amiga hinzu, da es für erste Gehversuche vollkommen ausreicht. Wir werden in diesem Kurs vor allem die Amiga-spezifischen Grafik, Sound- und Dateibefehle besprechen und Ihnen zeigen, wie man unter AmigaBASIC effektiv programmieren kann. Und am Ende unseres Einsteigerkurses wartet als kleine Belohnung eine gebrauchsfertige Telefondatei mit einigen Extras auf Sie.

Doch jetzt genug der Vorrede,
fangen wir lieber an.

BASIC heißt...

... 'Beginners All Purpose Symbolic Instruction Code' – ein mächtiges Wortgebilde für ein eigentlich ganz vernünftiges System: Um dem Besitzer eines Computers das Programmieren zu erleichtern, wurde eine Programmiersprache entwickelt, die leicht erlern- und anwendbar sein sollte, nämlich BASIC. Grundlage für BASIC sind die vielfältigen Befehlswörter, die wichtige Routinen mit einem kurzen, aber einprägsamen, an die englische Sprache angelehnten Ausdruck aufrufen.

Bei BASIC und auch beim AmigaBASIC handelt es sich um eine Interpreter-Sprache. Dies bedeutet, daß die Befehlswörter zwar für uns Anwender verständlich sind und zusammen auch ein Programm bilden können, für den Prozessor allerdings, der ja bekanntlich ausführendes Organ innerhalb eines Computers ist, sind unsere BASIC-Worte unverständliches Kauderwelsch. Er benötigt zum Arbeiten nur zwei Informationen, genannt '0' oder '1'. Unser BASIC-Programm muß also zuerst umgewandelt werden, zum besseren Verständnis für den Prozessor. Dazu dient der Interpreter. Die Befehle des BASIC werden einzeln vom Interpreter eingelesen, auf richtige Schreibweise überprüft und anhand einer Tabelle mit einer Kennzahl

der Fachmann spricht von 'Token', ausgetauscht. Diese Token gibt wiederum eine Stelle im Speicher an, an der eine Maschinensprache-Routine steht, die genau das ausführt, was wir mit unserem BASIC-Befehl erreichen wollten.

Damit haben wir gleich außer dem Interpreter auch den Hauptgrund für die Langsamkeit des BASIC kennengelernt. Denn jedes Mal, wenn Sie Ihr Programm starten, muß es erst Zeile für Zeile durch den Interpreter, bevor ein Ergebnis zu sehen ist. Bevor Sie jetzt skeptisch in die Gegend schauen, gibt es erst mal Entwarnung: So schlimm, wie es auf den ersten Blick aussieht, ist es zum Glück nicht. Zwar ist der Interpreter des AmigaBASIC nicht einer der schnellsten, jedoch läßt sich durch geschicktes Programmieren so manche Zeitersparnis erzielen. Und dies vor allem deshalb, weil AmigaBASIC einen Vorteil gegenüber so manchem anderen BASIC-Dialekt hat, der diese Schwäche wieder wettmacht: Es ist strukturierbar.

Einrücken, zweirücken...

Wissen Sie, was ein 'Spaghetti-Code' ist? Bevor Sie jetzt an italienische Nahrungsmittel denken, lesen Sie lieber erst mal weiter. Sie, als Programmierer, wollen sich eine Datei für Ihre Briefmarkensammlung schreiben. Gut und schön, Sie setzen sich hin und programmieren munter drauf los – bis Sie merken, daß sich ein Programmteil mehrmals wiederholt. Diesen, so meinen Sie, kann man als Unterprogramm deklarieren und von jeder beliebigen Stelle aus aufrufen, wozu gibt es den GOTO (Sprünge zur Zeile-) Befehl. Und dann fällt Ihnen auf, daß Sie einen wichtigen Programmteil vergessen haben, und dann noch einen, und dann muß noch dies geändert, das verbessert werden, und schon ist die Übersichtlichkeit futsch, Sie sind genervt, und das Programm wandert als Sondermüll in die Entsorgungs-Schublade, ohne jemals in die engere Auswahl zum Hit des Monats zu kommen. AmigaBASIC stellt uns da ein Werkzeug zur Verfügung, daß es in sich hat: Durch eine aus-

geklügelte Programmstruktur mit vielen Entscheidungs- und Abfragemöglichkeiten kann man weitestgehend auf Sprünge innerhalb eines Programms verzichten. Das erhöht die Lesbarkeit und den Wirkungsgrad und reduziert das vorhin benannte Interpreterproblem auf ein Minimum. Durch Einrücken von Zeilen läßt sich auch innerhalb von Schleifen oder Abfragen eine gute Lesbarkeit erzeugen.

Machen wir doch die Probe aufs Exempel und schauen uns ruhig einmal Listing 1 an: Na, wie fanden Sie das? Garantiert nicht gut, denn es war das beste Beispiel dafür, wie man es nicht machen sollte. Kommen wir jetzt zu Listing 2:

Das ist ja gleich ein Hammer. Ein so kurzes Programm, und doch sieht man auf den ersten Blick, daß hier mehr Übersicht herrscht. Und noch etwas sieht man hier: AmigaBASIC benötigt keine Zeilennummern.

"Ja, aber...", vernehmen wir des Lesers Stimme, "...woher soll ich denn dann wissen, wo ein Programmteil anfängt, den ich aufrufen will?" Genau das soll das Ziel unseres Kurses sein!

Willkommen im BASIC-Editor...

Wie schön wäre es, wenn wir mit dieser Zeile im AmigaBASIC begrüßt würden, dann wüßten wir wenigstens, wo wir sind. Statt dessen...

Um in das BASIC zu gelangen, müssen wir unseren Amiga zuerst mit der Workbench 'hochfahren'. Besitzer eines Zweitlaufwerkes sind mal wieder im Vorteil, sie brauchen nur die 'EXTRAS'-Diskette in DF1: einzulegen, während die 'Only-One-User' die nervige Diskette-rein-Diskette-raus-Prozedur über sich ergehen lassen müssen. In der Extras-Diskette befinden sich außer dem BASIC-Interpreter noch Demo-Programme sowie Utilities, darunter ein Texteditor namens MEMacs (Micro-Emacs). Was wir mit diesem Texteditor anstellen können, das erfahren wir noch an anderer Stelle. Wir öffnen das BASIC durch Anklicken des Icons. Kurz darauf meldet sich der Amiga mit zwei Fenstern wieder, das eine dient zum Eingeben der Programme, während das andere auf Direkteingaben reagiert.

```

1 REM Programm SO_NICHT
2 REM -----
100 farbe=1:x1=1:y1=1:x2=640:y2=256:flagx=0:flagy=0:abst=4
110 SCREEN 1,640,256,4,2:WINDOW 2,"SO NICHT!!!",3,1
120 LINE (x1,y1)-(x2,y2),farbe:LINE (x1,y2)-(x2,y1),farbe
130 IF flagx=0 THEN x1=x1+abst:x2=x2-abst:IF x1>640 THEN flagx=1
140 IF flagx=1 THEN x1=x1-abst:x2=x2+abst:IF x1<2 THEN flagx=0
150 IF flagy=0 THEN y1=y1+abst:y2=y2-abst:IF y1>255 THEN flagy=1
160 IF flagy=1 THEN y1=y1-abst:y2=y2+abst:IF y1<2 THEN flagy=0
170 farbe=farbe+1:IF farbe=4 THEN farbe=1
180 w$=INKEY$:IF w$="" THEN 120

```

Listing 1: Ein Beispiel dafür, wie man es nicht machen sollte! AmigaBASIC beherrscht zwar das Hantieren mit Zeilennummern, besser ist es jedoch, sich vor dem Programmieren Gedanken über das Projekt zu machen

Nach dem Start des BASIC erscheinen auch zwei Meldungen über den freien Speicher des Computers, einmal der Gesamtspeicher und zweitens der Speicherbereich, der für unsere Programme zur Verfügung steht. Neuankömmlinge werden aber jetzt ihren Augen nicht trauen; ganze 25000 Bytes stellt uns der Amiga dazu zur Verfügung. Da hatte ja der gute alte C-64 noch mehr frei, ganz zu schweigen von den anderen 'Oldies'.

Hier müssen sich die Newcomer ein bißchen umstellen, da einiges nicht mehr dem gewohnten Bild entspricht. Nicht nur, daß man sich nicht mehr direkt nach dem Einschalten im BASIC-Editor befindet, sondern ihn erst nach einer Reihe von Klimmzügen, sprich Icon-Klicks, erreicht, dann hat man noch weniger Speicherplatz zur Verfügung als auf dem vorherigen 'Achtbiter', und daß auf einem Computer, der 512 kByte beziehungsweise 1 MByte an Speicher besitzt (angeblich!!). Des Rätsels Lösung ist recht einfach: Wir selbst können unsere Programme für größeren Speicherbedarf einrichten. Dazu gibt es den Befehl CLEAR. Er richtet uns den Gesamt-BASIC-Speicher nach Bedarf ein. Doch dazu auch später mehr.

Jetzt soll uns erst einmal interessieren, wie wir Programme eingeben, ablaufen lassen und abspeichern.

Ohne Maus geht nix!

Der BASIC-Editor des Amiga ist, wie soll es auch anders sein, mausunterstützt. Wir können dies ganz einfach feststellen, indem wir die rechte Maustaste gedrückt halten, sofort erscheint eine Menüleiste, in der eine Menge Funktionen stehen. Schauen wir uns die einzelnen Funktionen, also die herunterge-

lassenen Boxen mit den Funktionen, einmal in Ruhe von links nach rechts an:

Das erste Menü hat den Namen PROJEKT, in ihm sind alle Funktionen für den Umgang mit BASIC-Dateien zu finden, an erster Stelle die Funktion NEW. Mit ihr können wir den BASIC-Speicher für ein neues Programm freigeben. NEW ist aber auch ein BASIC-Befehl, daher können wir NEW in ein Programm einbinden oder in Direkteingabe im BASIC-Fenster zur Wirkung kommen lassen. Doch Vorsicht!! NEW löscht ein Programm gänzlich im Speicher. Das Amiga-Betriebssystem ist jedoch in dieser Beziehung sehr anwenderfreundlich, der Rechner fragt vorher nach, ob wir das erstellte Programm nicht doch lieber vorher sichern, also auf Diskette speichern wollen. Bei einem Nein unsererseits besteht dann aber keine Hoffnung mehr auf ein Wiedersehen, also vor dem Ausschalten des Computers erst einmal das Denkkzentrum einschalten.

Als nächstes steht die Funktion OPEN auf dem Programm. Mit OPEN holen wir uns abgespeicherte Programme zurück in den Editor. Nach Anwählen der OPEN-Funktion fragt unser Amiga nach dem Namen des Programms. In die Dialogbox können wir zum Beispiel "TEST.BAS" oder "DF1:TEST.BAS" eintragen, Zweitlaufwerke werden also auch anerkannt.

Zum Starten eines Programms können wir aber auch zwei BASIC-Befehle einsetzen: LOAD und RUN. Mit LOAD wird ein Programm nur geladen, während es mit RUN direkt nach dem Laden gestartet wird. 'RUN"TEST.BAS"' lädt und startet also das Programm TEST.BAS in einem Arbeitsgang.


```

REM Programm: So ist's besser!
Zuweisungen:
Farbe = 1:x1 = 1:y1 = 1:x2 = 640:y2 = 256
FlagX = 0:FlagY = 0:Abstand = 4
SCREEN 1,640,256,4,2
WINDOW 2,"SO ist's besser !!!",,3,1
Schleife:
WHILE INKEY$=""
  LINE (x1,y1)-(x2,y2),Farbe
  LINE (x1,y2)-(x2,y1),Farbe
  IF FlagX=0 THEN
    x1=x1+Abstand:IF x1>640 THEN FlagX=1
    x2=x2+Abstand
  ELSE
    x1=x1-Abstand:IF x1<2 THEN FlagX=0
    x2=x2+Abstand
  END IF
  IF FlagY=0 THEN
    y1=y1+Abstand:IF y1>255 THEN FlagY=1
    y2=y2-Abstand
  ELSE
    y1=y1-Abstand:IF y1<2 THEN FlagY=0
    y2=y2+Abstand
  END IF
  Farbe=Farbe+1:IF Farbe=4 THEN Farbe=1
WEND

```

Listing 2: So ist es schon besser: Ein strukturiertes BASIC-Programm zeigt, wo es langgeht. Versuchen Sie, anhand der beiden Beispiele die Aufgabe zu lösen

Die nächsten beiden Funktionen heißen SAVE und SAVE AS. Bei SAVE wird ein Programm unter dem Namen abgespeichert, unter dem es sich schon einmal auf Diskette befand, dies erspart einem die immer wiederkehrende Namensangabe. Nur bei SAVE AS wird nach dem Namen gefragt, hier kann man also auch ein und dasselbe Programm unter verschiedenen Namen abspeichern.

QUIT schließlich beendet AmigaBASIC und geht zurück ins DOS (oder in die Workbench, je nachdem, von wo Sie BASIC aus gestartet haben).

Das nächste Pulldown-Menü namens EDIT beschäftigt sich mit dem Editor selbst. Hier finden wir als erstes die Funktion CUT. Mit CUT können wir einen selektierten Bereich des Programms entfernen und in einen separaten Speicherbereich ablegen. Selektieren können wir, indem wir mit der Maus den Cursor an die Anfangsposition bringen, die linke Maustaste aber gedrückt halten und den Endpunkt ansteuern. Der selektierte Bereich wird jetzt in der Grundfarbe unterlegt. Mit PASTE wird der zwischengespeicherte Text an anderer Stelle wieder geschrieben. COPY funktioniert wie CUT, nur daß hier der selektierte Bereich nicht vom Bildschirm

und aus dem Programm gelöscht wird. Das Arbeiten mit COPY, CUT und PASTE noch einmal in Kürze:

1.) Programmtext an andere Stelle bringen:

- Selektieren,
- mit CUT entfernen (wird zwischengespeichert),
- mit PASTE an die neue Stelle bringen.

2.) Programmtext kopieren:

- Selektieren,
- mit COPY zwischenspeichern, ohne zu entfernen,
- an anderer Stelle mit PASTE übernehmen.

Selektierte Bereiche lassen sich mit der BACKSPACE-Taste übrigens auch ganz entfernen. Ist doch eigentlich eine einfache Sache, oder?

In der nächsten Menübox mit dem Namen RUN befinden sich Funktionen für den Programmablauf. START und STOP sind wohl auf den ersten Blick klar, sie starten und stoppen (was denn sonst!) ein im Editor befindliches Programm. CONTINUE setzt an der Stelle im Programm wieder an, an der mit STOP geendet wurde – vorausgesetzt, es wurde nicht mit einer Fehlermeldung abgebrochen, denn dann weigert sich der Amiga, auf CONTINUE einzugehen. SUSPEND 'friert'

ein ablaufendes Programm ein, ohne in den Direktmodus überzuwechseln. Sollte Ihr selbstgeschriebenes Programm einen Fehler aufweisen, so läßt sich dieser mit SUSPEND genau verfolgen, wenn er auch keine Fehlermeldung erzeugt. TRACE ON/TRACE OFF ist ein Einzelschritt-Ablauf, indem jede Programmzeile einzeln abgearbeitet wird. TRACE kann ein- (ON) bzw. abgeschaltet (OFF) werden, man erhält damit eine gute Programm-Ablaufkontrolle. STEP schließlich ist der Einzelschritt-Ablauf im Listing-Interpreter. Jede Zeile, die gerade vom Interpreter abgearbeitet wird, erhält einen roten Rahmen zur Anzeige. Durch wiederholtes Anklicken von STEP wird also jede Zeile durch den Interpreter geschickt. Mit TRACE, SUSPEND und STEP hat man eine komfortable Programmkontrolle zur Hand, die eventuelle Fehler sofort an der richtigen Stelle anzeigt. Das letzte Menü-Fenster heißt WINDOWS und enthält die Funktionen SHOW OUTPUT und SHOW LIST. Mit Ihnen wird zwischen dem List-Fenster, in dem das Programm steht und dem Anzeige-Fenster, in dem alle Ergebnisse des Programms stehen, hin und hergeschaltet. So, damit hätten wir die Funktionen des BASIC-Editors durch, schauen wir uns jetzt einmal an, wie wir Programme erstellen.

Auch Programmieren will gelernt sein!

Das Eingeben der Programme gestaltet sich im AmigaBASIC eigentlich recht bequem. Für direkte Eingriffe steht uns das BASIC-Fenster, für Listings das LIST-Fenster zur Verfügung. Jedoch muß sich der Einsteiger, der sich gerade mühsam ein bißchen in das AmigaDOS eingearbeitet hat, wieder einmal in puncto Dateibehandlung umgewöhnen. Deshalb wollen wir anhand eines Miniprogramms die Unterschiede kennenlernen.

Zuerst klicken wir mit unserer Maus das LIST-Fenster an, sofort erscheint der Cursor in der ersten Zeile. Sollten Sie den Cursor nicht oder kaum sehen, erscheint es ratsam, mittels Preferences sofort bessere Bildschirmfarben einzustellen.

Geben wir nun die erste Zeile unseres Kurzlistings ein:

REM Kurzes Listing

REM markiert eine Kommentarzeile, alles was nach REM kommt, wird vom Interpreter nicht beachtet, deshalb lassen sich Programme mit REM kommentieren, ohne daß es zu Fehlermeldungen kommt. Einen Nachteil hat REM allerdings, die Kommentare brauchen Speicherplatz. Bei einer REM-Zeile geht's ja noch, aber wenn alle Programmteile kommentiert werden, dann... Kommen wir aber zum Rest unseres kleinen Programms:

```

FOR n = 1 to 20
LOCATE n,1:PRINT "HALLO
WELT"
NEXT n

```

Hier die Erklärung zu den Zeilen:

Die FOR-NEXT-Schleife wird benötigt, um Anweisungen beliebig lange zu wiederholen, in unserem Fall wird durch den Wert 'n' der Cursor 20mal in den Zeilen 1 bis 20 an den Anfang der Zeile positioniert, um dort den Satz "Hallo Welt" hinzuschreiben. Wir nehmen also eine Zählvariable, nämlich 'n', und nutzen sie zweimal, einmal als Zähler für die FOR-NEXT-Schleife und als Zeilenposition für den Cursor. So weit, so gut. Unser kleines Programm druckt nun den Satz "Hallo Welt" in den Zeilen 1 bis 20. (Deshalb nennt man es unter Profis "Hallo-Welt-Programm" und lächelt bei dessen Anblick satanisch über alles, was wie ein Einsteiger aussieht. Daß die 'Hallo-Welt-Programme' gerade den Einsteigern beim Einstieg helfen sollen, bedenken diese 'Fachleute' nie!). Starten Sie es ruhig einmal, indem Sie in der Menüleiste die Option START mit der Maus anklicken. Kurz darauf hat sich der OUTPUT-Bildschirm mit 20 'Hallo Welt' angereichert. Nun ist allerdings das List-Fenster weg.

Jedoch ist dies kein Problem, mit der Menüleisten-Funktion SHOW LIST erscheint es sofort wieder. Probieren Sie jetzt ruhig einmal die Funktionen des BASIC-Editors aus, als erstes sollten Sie jedoch die Funktion SAVE AS anklicken. Hier öffnet sich ein sogenannter Requester, eine Entscheidungshilfe. Der Amiga fragt nach, wie das Programm heißen soll, das er in seine Diskettenobhut übernimmt. Den Namen geben wir

In den reichsten
Fischgründen Westafrikas:

Leere Netze

Vor der Küste Senegals hatten traditionsgemäß viele Kleinfischer ein sicheres Auskommen durch den Fang von Thunfisch, Sardinen und Krustentieren. Heute bevölkern sie die Slums von Dakar, der Hauptstadt des Senegal, oder arbeiten als schlechtbezahlte Tagelöhner auf den staatlich geförderten Hochseefangflotten, die ihre Fischgründe leerfischen. Eine kleine Gruppe junger Leute geht einen anderen Weg: Sie schlossen sich zu einer Fischereigenossenschaft zusammen. Durch Kredite aus BROT FÜR DIE WELT-Spenden können sie ihre Einbäume so ausstatten, daß sie trotz erdrückender Konkurrenz das Überleben ihrer Familien sichern.

BROT FÜR DIE WELT-Spenden helfen bei der Finanzierung.

Postf. 101142 · 7000 Stuttgart 10

Brot

für die Welt

Postgiro Köln 500500-500

Sie besitzen einen AMIGA?

Sie wollen ihn erweitern?
Sie wissen nicht wie?

Aber wir !!!!

Wir bieten für AMIGA:

- Speichererweiterungen
- Festplatten
- Wechselseitplatten
- Tapestreamer
- Turboprozessorkarten
- AMIGA-Reparatur in eigener Werkstatt

Auch für Ihren DOS-Rechner oder Bridgecard bieten wir Zubehör oder Komplettlösungen.

Rufen Sie an oder
kommen Sie vorbei:

ABAG

Andreas Bergmann &
Adalbert Geyer GdBR

Schöner Str. 17b
Tel. 06 21/78 74 01

6800 Mannheim 31
Fax: 06 21/78 74 09

JINTECK-VERSAND, Auwiesenweg 3, D-8049 Unterbrück/Fahrenzhausen, Tel. 0 81 33/801-3



JINTECK

S P A B am Home-Computer S P A B am Home-Computer

Modell	Kaufpreis/DM		Versand/DM	
Diskettenlaufwerke				
3 A-Senator	233,--			
Master 5 A-1	296,--			
Master 3S	252,--			
Master 5S	355,--		6,--	
Beschleuniger				
Oceanic Turbo Dos	45,--		5,--	
Mouse				
Oceanic Mouse Amiga	75,--			
Oceanic Mouse Atari	75,--			
Oceanic Mouse C64	45,--		5,--	
Komplettes Spar-Paket	P1	P2	P3	P4
KY-1000H	x	x	x	x
Turbo Dos	x	x	x	x
Joystick		x		x
Mouse			x	
DM	300,--	304,--	339,--	343,--

* Aufpreis für ein Original C64 zu jedem Paket: DM 282,--
+ Versand DM 7,--

Modellbeschreibung:

KY-1000 H: 5,25"-Diskettenlaufwerke für C64 Commodore-Computer. Speicherkapazität 174 K (GCR-formatiert), 36 Spulen, 48 TPI-Dichte, Erweiterungsstecker für DOS
 Senator: 3,5"-Diskettenlaufwerke für Amiga-Computer, Speicherkapazität 1 MByte (MFM-formatiert), 160 Spulen, 135 TPI-Dichte
 Master 3A II: 3,5"-Diskettenlaufwerke intern für Amiga 2000-Computer (PDD)
 Master 5A-1: 5,25"-Diskettenlaufwerke für Amiga-Computer, Speicherkapazität 1 MByte (MFM-formatiert), 160 Spulen, 96 TPI-Dichte, 40/80-Schalter
 Master 3S: 3,5"-Diskettenlaufwerke für Atari ST-Computer, Speicherkapazität 1 MByte (MFM-formatiert), 160 Spulen, 96 TPI-Dichte, 40/80-Schalter
 Master 5S: 5,25"-Diskettenlaufwerke für Atari ST-Computer, Speicherkapazität 1 MByte (MFM-formatiert), 160 Spulen, 96 TPI-Dichte, 40/80-Schalter
 Turbo DOS-Cartridge: voll kompatibel zu OC-118N und KY-1000 H
 3 x schneller als 1541, kein Laden nötig
 voll kompatibel zu 64/128
 Ersatz für Atari ST-Maus
 Ersatz für Amiga-Maus

- * Ein Anruf bei uns genügt !
- * Das Paket wird Ihnen innerhalb 3 Tagen zugestellt !
- * Sie zahlen erst bei Paketzustellung !
- * 1/2 Jahr Garantie !
- * Reparatur dauert nur 3 Tage !

Der Joker für Ihren Einkauf

FunHouse

zuverlässig * preiswert * schnell

DISKETTEN:

weiße Ware 3.5" 2DD 15,98 DM
 (Staffelpreise erfragen!)
 Nashua 5.25" HD 10 Stk. inkl. Aufkleber 33,80 DM

*MODEMS:

Discovery 2400C+ 388,00 DM
 Discovery 2400C+ inkl. Kabel und Software 398,00 DM
 300,1200,2400 Baud, CITT V.21, V.22, V.22 bis
 Discovery 1200C+ 298,00 DM
 Discovery 1200C+ inkl. Kabel und Software 315,00 DM
 300,1200,1200 \ 75,75 \ 1200 Baud, CITT V.21, V.22, V.23

SPEICHERERWEITERUNGEN:

512 KB für A500 mit Akku-Uhr alle ICs ges. 179,00 DM
 512 KB für A500 145,00 DM

LAUFWERKE:

3.5" Floppy 199,00 DM
 durchgef. Bus, abschaltbar, slimline
 5.25" Floppy 265,00 DM
 40 \ 80 Tracks, abschaltbar, durchgef. Bus

SOFTWARE:

Shadow of the Beast 85,00 DM
 Barbarian II 65,00 DM
 Space Quest III 98,00 DM
 World Games 14,89 DM
 Road Runner 16,98 DM
 BTX/VTX Manager V2.2 FTZ 248,00 DM

ZUBEHÖR:

Diskettenbox für 80 Stk. 3.5" abschließbar 14,80 DM

PD-SERVICE

Fred Fish 1-2588 je 3.5" Disk 2,80 DM

- Wir kopieren nur auf WEISSER WARE 3.5" 2DD
- 100 % ERROR-free und virusfrei
- auch Pakete erhältlich: Games, Anwender usw.
- Wir stellen auch Disketten nach Ihren Wünschen zusammen

* Der Anschluß dieser Modems an das öffentl. Postnetz in der BRD und in West-Berlin ist bei Strafe verboten!!!!
 Der Versand erfolgt per Vorratszahlung (Überweisungformat \ ec-Scheck = Skonto) oder per NN zuzüglich Versandkosten.

24 Stunden Bestellservice · Ruf 0 21 51/63 12 78
 FunHouse · Inh. F. Ricken, Mariannenstr. 34 · 4150 Krefeld 1

Hard- und Software · Zubehör

Hard- und Software · Zubehör

im oberen Kästchen ein, danach brauchen wir nur noch OK anzuklicken. Den Namen können Sie sich selbst aussuchen, achten Sie aber darauf, daß Sie nur erlaubte Zeichen nehmen, sonst macht der Amiga Ärger. Beispiele sind "HALLO_WELT.BAS" oder "HALLOWELT.BAS". Die Endung ".BAS" ist nicht zwingend, aber auf diese Art unterscheiden sich ihre BASIC-Programme immer von anderen Dateien.

So, abgespeichert haben wir, jetzt wollen wir auch wissen wo. Also, wie wir vom AmigaDOS (hoffentlich) wissen, flugs DIR eingegeben und schon...

Blink! Piep! Undefined subprogram!

"Nanu??", denkt sich der Amiga-Besitzer. "Was meint er damit?" Nun, 'kleines dummes Amiga' hat das DIR als BASIC-Befehl oder Name einer Subroutine (was das ist, werden wir Ihnen noch zeigen) aufgenommen und nicht als DOS-Befehl.

Versuchen wir es mit dem zweiten DOS-Befehl LIST. Er wird eingegeben, RETURN gedrückt und... listet unser Programm auf, was noch im Listfenster steht. Stimmt also auch nicht.

Des Rätsels Lösung ist auch hier einfach: DOS-Befehle sind im AmigaBASIC nicht direkt nutzbar. Statt dessen stellt uns AmigaBASIC eigene Befehle zur Verfügung. Um die Dateien auf beliebigen Laufwerken anzeigen zu lassen, muß man den BASIC-Befehl FILES benutzen. Versuchen wir es:

Klicken Sie das OUTPUT-WINDOW an und geben Sie im Direktmodus ein: FILES. Danach zeigt der Amiga den Inhalt des gerade aktuellen Laufwerks, in unserem Fall von DF0: oder DF1:, je nachdem, wo Sie Ihre EXTRAS-Diskette eingelegt haben. Weitere Beispiele sind 'FILES"DF0:"', 'FILES"RAM:"', der logische oder physikalische Laufwerksname muß in Anführungszeichen eingegeben werden. Mit FILES lassen sich allerdings keine Jokerzeichen benutzen, 'FILES DF0: #?.BAS' geht also nicht. Was machen wir denn, wenn unser BASIC-Programm in einem Unterverzeichnis steht? 'FILES"DF0:BASIC/ BASIC_Programme' geht problemlos, jedoch wird die dauernde Eintipperei der Unterverzeichnisse lästig, wenn man

dauernd laden oder abspeichern muß. Im DOS gibt es den Befehl CD (Change Directory), in BASIC gibt es ein Äquivalent dazu, den Befehl CHDIR, er hat die gleiche Wirkung. Nach 'CHDIR"BASIC/BASICPROGRAMME"' werden mit FILES die Dateien des Unterverzeichnisses BASIC\BASICPROGRAMME angezeigt.

So, das soll fürs erste reichen, immerhin ist eine Menge Wissen jetzt schon durch Ihren Kopf gegangen, Sie können jetzt schon mit dem Editor arbeiten und die einfachen BASIC-Befehle nutzen, die Ihnen bekannt sind. Im nächsten Teil werden wir uns eingehend mit Grafik, Sound und Sprachausgabe beschäftigen, denn unser Amiga würde nicht Amiga heißen, wenn dies nicht auch im BASIC möglich wäre.

Zuvor wollen wir jedoch noch eines machen, nämlich eine startfähige Diskette erstellen.

BASIC automatisch

Wie schon am Anfang erwähnt, finden wir BASIC auf der EXTRAS-Diskette. Nun ist es aber nicht gerade schön, dauernd durch sämtliche Workbench-Icons zu müssen, nur um endlich programmieren zu können. Deshalb ist es sinnvoll, sich eine startfähige BASIC-Diskette zu erstellen. Doch was muß sich auf dieser Diskette alles befinden?

Zuerst formatieren wir eine neue Diskette, von der Workbench mit INITIALIZE, vom DOS aus mit dem FORMAT-Befehl. Sinnvoller ist es vom AmigaDOS aus, da wir hier beim Formatieren den 'Mülleimer' (Trashcan) durch die Option 'NOICONS' direkt weglassen können, da er a) Speicher verbraucht und b) sowieso nicht benötigt wird. Nun müssen wir die Diskette 'bootfähig', also startfähig machen, dazu dient der DOS-Befehl INSTALL. Auf die Diskette kopieren wir aus dem C:-Verzeichnis der Workbench-Diskette die am meisten gebrauchten Befehle des DOS, wie COPY, FORMAT, CD, DIR, INFO und des weiteren in das Stammverzeichnis (ROOT) der neuen BASIC-Diskette. Um mit dem deutschen Zeichensatz arbeiten zu können, brauchen wir aus dem SYSTEM-Verzeichnis das SETMAP und aus dem DEVS/KEYMAPS-Unterverzeichnis die D-Datei. Jetzt

fehlt nur noch das BASIC von der Extras-Diskette und das Unterverzeichnis S: mit der Startup-Sequenz, oder?

Nicht ganz, wir haben etwas ganz Wichtiges vergessen. AmigaBASIC benötigt zum Kommunizieren mit dem Betriebssystem des Amiga eine Reihe von System-Bibliotheken, die sogenannten Libraries. In ihnen sind Routinen zum Arbeiten mit Windows und Screens, das Dateihandling für Sound, Grafik und noch einiges mehr enthalten.

Danach befinden wir uns in vertrauter Umgebung.

So, das war unser erster Kurs-Teil, im nächsten geht es endlich 'ans Eingemachte', nämlich ans Programmieren. Sie werden allerdings nicht entlassen, ohne daß Sie eine kleine Aufgabe zu lösen haben:

Schreiben Sie ein Programm, das mittels des LINE-Befehls waagerechte Linien in Farbe 1 vom oberen zum unteren Rand des Bildschirms zieht. Wenn der Bildschirm voller

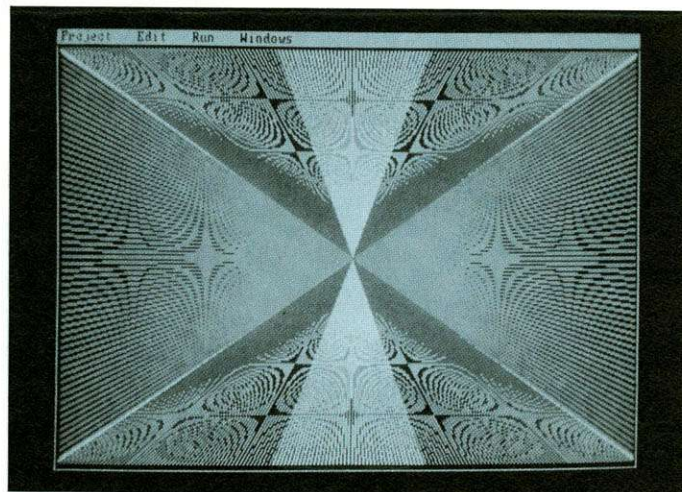


Bild 1: Übrigens: Das soll unser Beispielprogramm hervorbringen. Funktionieren wird es mit Listing 1 und Listing 2, jedoch wird mit Listing 2 besser ersichtlich, warum

Als BASIC-Einsteiger hat man am Anfang keinen Kontakt mit ihnen, wenn Sie jedoch später BASIC in- und auswendiglernen, werden Sie schnell wieder auf sie stoßen. Für unseren Kurs sind sie im Augenblick nur deshalb interessant, weil AmigaBASIC sie benötigt. Zu finden sind sie im FD1.2- (oder FD1.3-) Verzeichnis. Dieses Verzeichnis kopieren wir uns ebenfalls vollständig auf die BASIC-Diskette. Jetzt fehlen aber immer noch wichtige Dateien, die AmigaBASIC befähigen, bei Bedarf auf das System zurückzugreifen. Diese Dateien finden wir im LIBS:-Verzeichnis. Auch dieses wollen wir komplett auf die BASIC-Diskette kopieren, auch wenn nicht alle Libraries (schon wieder!) benötigt werden. Jetzt benötigen wir noch die DATEI NARRATOR.DEVICE aus dem Unterverzeichnis DEVS, damit unser Amiga auch sprechen kann. Was jetzt noch fehlt, ist eine Startup-Sequenz, die eigentlich nur aus zwei Zeilen besteht:

```
SETMAP D
AMIGABASIC
```

Linien ist, soll das Programm mit der nächsthöheren Farbe wieder anfangen, bis die Farben 1 bis 3 genutzt wurden. Benutzen Sie dazu die Funktionen des EDITORS.

Die benötigten Befehle sind:
CLS: Löscht das OUTPUT-Window
FOR/NEXT für Schleifen
LINE(x-Koordinate1,y-Koordinate1)(x-Koordinate2,y-Koordinate2),
Farbe
LINE(1,1)-(619,189),1 zieht eine Linie von den Koordinaten 1,1 (links oben) zu den Koordinaten 619,189 (rechts unten) in der Farbe 1 (kann je nach Farbgebung durch Preferences variieren).

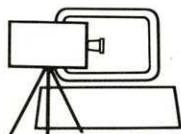
Als Variablen können dienen:

Die ANZAHL der Durchläufe: 3 (ist gleichzeitig Anzahl der Farben)

Die Zeilen: 189

Mit diesen Angaben sollte es Ihnen möglich sein, das Programm zu erstellen, und denken Sie daran: Erst speichern, dann starten.

(jb)



VD 2000 AMIGA

Echtzeit Color Video Digitizer

- eingebauter RGB-Splitter mit FBAS/S-VHS-Eingang
- REAL-TIME S/W und REAL-TIME COLOR (20 ms)!!!
- Steckkarte für AMIGA 2000
- **65536 Farben gleichzeitig darstellbar!!!**
- Alle Videoauflösungen
- alle Funktionen sind softwaregesteuert
- umfangreiche Software (IFF-kompatibel), Help usw.
- Bildmanipulation, Verfremdung, Verbesserung usw.
- **eigener Bildspeicher**
- RGB-Ausgang für zweiten RGB-Monitor
- komplett in moderner SMD-Technik

Preis 2.898,-
Genlocks: MiniGen, MaxiGen, AG5

MERKENS EDV

Computer-Video-Systeme

VD 4 AMIGA

Color Video Digitizer

- eingebauter RGB-Splitter mit FBAS/S-VHS-Eingang
- REAL-TIME S/W, COLOR in 60 ms
- für alle AMIGAs, Videokameras und Videorekorder
- bis 6 Bit/Farbauszug, Lo/HiRes, Overscan und Interlace
- alle Funktionen sind softwaregesteuert
- umfangreiche Software (IFF-kompatibel), Help usw.
- umschaltbarer Druckerport!!!
- schnelle Outline- und Differenzbilderzeugung
- komplett in moderner SMD-Technik und Flachgehäuse
- mit externem Netzteil

Preis 1.298,-

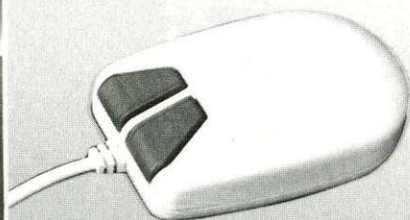
Fuchstanzstraße 6 a · 6231 Schwalbach · Telefon (0 61 96) 30 26 · Telefax (0 61 96) 8 27 49

reis-mouse

Maus für AMIGA 500 / 1000 / 2000
Die Maus, die Ihren Computer schneller macht

- sofort anschließbar
- hohe Qualität
- ergonomische Form

1 Jahr Garantie



extrem preiswert -
vergleichen Sie!!

Verkauf über den Fachhandel - Händleranfragen erwünscht

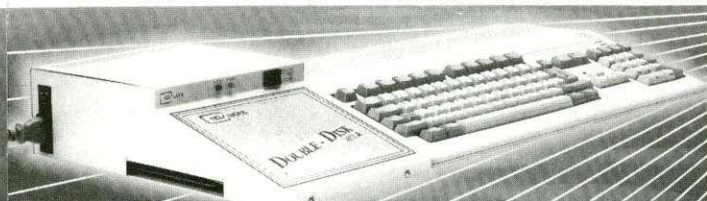
reis-ware
Computer-Produkte GmbH

Postfach 36
D-5584 Bullay
Telefon 06542/2086-2087
BTX * Reisware #
Fax 06542/21017

Neue Dimensionen

DOUBLE-DISK
2012 A

für
AMIGA 500



Hervorragende
Testergebnisse in

AMIGA-MAGAZIN 8/1989
und AMIGA-POWER 7/89!

AMIGA-Test

gut

8,1
von 12

GESAMT-
URTEIL
AUSGABE 8/89

- 20 MB/40 MB Hard-Disk + 2 MB RAM-Erweiterung
- einfache Bedienung durch Bildschirmführung
- durchgeführter Datenbus
- ansprechendes, formschönes Design

Edgar Meyzis

Aufsteigerwissen Assembler

In diesem Heft starten wir einen Assemblerkurs für Amiga-Freaks, die ihre Maschine schon ein wenig kennen und auch bereits über Erfahrung in der Programmierung verfügen. Wir wollen gemeinsam mit Ihnen ein kleines Projekt in Assembler mit einfachen Mitteln sicher realisieren. Steigen Sie ein, der Zug fährt ab.

Man kann eine Wüste auf verschiedene Weise überwinden, zu Fuß, mit einem Jeep oder per Flugzeug. Jede Art der Überquerung hinterläßt ihre Eindrücke, verlangt unterschiedliche Anstrengungen und unterscheidet sich im Zeitaufwand.

Zur Konzeption des Kurses

Jeder, der die Wüste wirklich kennenlernen will, muß das Flugzeug als Mittel ablehnen. Vernünftigerweise würde man sich für einen Jeep entscheiden. Ähnlich ist es auch mit der Programmierung des Amiga, die man mit einer Wüste vergleichen könnte, obwohl sie nicht wüst erfolgen sollte. Mit einer Hochsprache, wie z.B. BASIC, fliegt man darüber hinweg. Zu Fuß zu programmieren, das heißt, einzelne Bits zu setzen, scheidet aus. Der Einsatz der Maschinensprache mit Hilfe eines Assemblers könnte den Vergleich mit dem Jeep standhalten.

Ziel des Kurses ist es, Ihnen Sicherheit im Umgang mit Assembler zu vermitteln, die Sie dazu befähigt, die gewonnenen Erfahrungen in Ihren Programmieralltag einzubringen. Um das Ziel auch wirklich in fünf Folgen zu erreichen, legen wir den Schwerpunkt auf die Programmierung von Grafik und Dateihandling einschließlich der Programmierung einer Benutzeroberfläche. Am Ende einer jeden Etappe wartet eine Oase auf uns, ein lauffähiges Programm als Belohnung für die erlittenen Strapazen. Mit Abschluß der vierten Folge werden Sie über ein Malprogramm verfügen, das wir

"Paint Box" genannt haben. Die "Paint Box" wird vornehmlich von Ihnen mit Malwerkzeugen gefüllt werden müssen. Wir werden nur einige wesentliche exemplarisch behandeln, Ihnen aber weiterführende Wege aufzeigen. Der Kursfahrplan informiert Sie grob über den Inhalt der Folgen. Ziel des Kurses ist es nicht, Sie im einzelnen in die Programmierung des Prozessors 68000 und des Amiga einzuführen. Wir wollen Assembler auf die besprochene Aufgabenstellung konkret anwenden und Sie motivieren, eventuell vorhandene Wissenslücken im Bereich der Grundlagen anhand von Literatur (siehe beigefügtes Verzeichnis) selbst zu schließen.

Unsere Tools – Assembler und Linker

Wer schaut noch die "Fish-Disks" 186 oder gar 110 an, wo wir uns auf die laufende Nummer 300 zubewegen? Wir haben es getan und uns entschlossen, den dort enthaltenen Assembler "A68k" (Fish 186) und den Linker "BLink" (Fish 110) in unserem Kurs einzusetzen. Entscheidend für die Auswahl war, daß beide Tools so ohne nennenswerten finanziellen Aufwand zur Verfügung stehen.

Bei dem A68K handelt es sich um einen zweiphasigen Makroassembler, der auch mit Include-Dateien umzugehen weiß und den üblichen Konventionen genügt. Uns sind natürlich verschiedene kommerziell vertriebene Assembler mit ihren Raffinessen bekannt. Da nicht davon auszugehen ist, daß die bei unseren Lesern vorhandenen Assem-

bler von einem Typ sind, galt es, einen gemeinsamen Nenner zu finden, den wir im A68K sahen.

Es wird Ihnen nicht schwerfallen, unsere Programme mit dem von Ihnen gewohnten Assembler nachzuvollziehen, da der A68K keine einmaligen Besonderheiten aufweist. Um mit dem Assembler Bekanntschaft zu schließen, sollten Sie zunächst die ihm beigegebene Datei "A68k.doc" ausdrucken. Sie werden entdecken, daß im wesentlichen der Aufruf des Assemblers (Seite 6) und einige Eigenschaften dargestellt sind, mehr nicht. Wir werden den Assembler schrittweise bei der praktischen Arbeit kennenlernen. Listing 1 demonstriert einige Möglichkeiten des Assemblers und ist als Orientierungshilfe gedacht.

Sollten Sie feststellen, daß Sie bereits jetzt schon sehr viel mehr über die Arbeitsweise eines Makroassemblers und die Instruktionen für den 68000 wissen wollen, dann stöbern Sie einmal in einer Buchhandlung herum. Die Literaturhinweise können die zielgerichtete Suche erleichtern.

Probieren wir unseren A68K doch gleich einmal von einer Sicherheitskopie aus. Bevor wir unser erstes Programm schreiben, sollten wir jetzt schon an den sicheren Weg denken, den wir einschlagen wollen. Ordnung kann zur Sicherheit beitragen. Es bietet sich deshalb an, ein Verzeichnis mit der Bezeichnung "Assembler" anzulegen und dazu Unterverzeichnisse mit den Namen "txt", "obj", "bin" und "incl". Das Verzeichnis "txt" soll die Quelltexte aufnehmen, und "obj" ist für den

Output des Assemblers gedacht. Die fertigen Programme finden in "bin" ihren Platz und "incl" wird Textdateien aufnehmen, die wir noch zu erstellen haben. Rufen Sie Ihren Lieblingseditor auf, und schreiben Sie einfach die Assemblerbefehle

```
move.l d0,d1  
rts
```

auf den Bildschirm. Speichern Sie den Text als D0D1.asm im Unterverzeichnis "txt" ab. Der Suffix "asm", der nicht zwingend vorgeschrieben ist, kennzeichnet Quelltexte als Assemblerdateien.

Rufen Sie nun den Assembler "A68K", der im aktuellen Pfad liegen muß, wie folgt auf:

```
A68K txt/D0D1.asm obj/D0D1.o
```

Dabei wird vorausgesetzt, daß das aktuelle Verzeichnis (cd) "Assembler" lautet, wie vorhin vorgeschlagen. Mit dem Aufruf weisen Sie den Assembler an, die Textdatei zu assemblieren und das Ergebnis, den erzeugten Code, im Unterverzeichnis "obj" abzulegen. Der Suffix "o" kennzeichnet so erzeugte Objektdateien. Bei der Arbeit des Assemblers können Sie erkennen, daß zwei Phasen durchlaufen werden. Leider wird Ihre erste Bekanntschaft gleich mit einer Fehlermeldung belohnt. Jeder Quelltext muß nämlich mit einem "END" (oder "end") abgeschlossen sein. Fügen Sie also dem Quelltext eine weitere Zeile mit "end" hinzu. Ihr erstes Programm müßte jetzt Listing 2 entsprechen. Sie wissen nun, daß der "Fish", das heißt der A68K, nicht verdorben ist.

Sind Ihnen die Kommentare aufgefallen, die wir in Listing 2

Kursfahrplan

Teil 1. Einführung und Grundlagen

Teil 2. Vertiefung der Programmiertechnik

Teil 3. Praktische Arbeit mit Graphics

Teil 4. Praktische Arbeit mit DOS

Teil 5. Hardwarenahe Programmierung von Grafik

hinter Semikolons versteckt haben? Gut so, denn Kommentare erleichtern das Lesen von Programmen. Der in der ersten Zeile vermerkte ist jedoch recht überflüssig, da die Assemblersprache selbst deutlich macht, daß der Inhalt eines Registers in ein anderes kopiert werden soll. Hier hätte vermerkt werden müssen, warum das Kopieren erfolgt. Also ein Negativbeispiel!

Sollten Sie noch tieferen Einblick in die Arbeit des Assemblers nehmen wollen, dann fügen Sie bitte dem Aufruf `“l*”` an. Der Assembler protokolliert nun seine Arbeit in Form eines Listings auf den Bildschirm, das Sie z.B. auch mit `“ram:Temp”` anstelle des Sterns in eine editierbare Datei umleiten und mit `“prt:”` sogar an den Drucker schicken können. Objektdateien enthalten jedoch noch keine ablauffähigen Programme. Wir müssen noch einen Linker ansetzen, der uns den Objektcode mit einem Aufstartcode zu einem lauffähigen Programm bindet. Diese Arbeit kann `“BLink”` verrichten. Das Programm finden Sie auf der Fish Disk 110 im Unterverzeichnis `“PDC/Examples”` versteckt. Der Linker sollte, wie der Assembler, im aktuellen Pfad liegen, z.B. im C-Verzeichnis. Also lassen wir BLink für uns arbeiten mit dem Aufruf:

```
BLink obj/DOD1.o TO bin/DOD1
```

Das erste Programmchen ist fertig. Starten Sie es mit

```
bin/DOD1
```

und nichts Sichtbares läuft ab. Das ist auch gut so, denn wer will schon ständig Gurus zu Besuch haben. Übrigens, seit wann leben Gurus in der Wüste?

Damit sind uns unsere Werkzeuge schon ein wenig vertraut. Wir werden sie noch genauer kennenlernen und bald auch beherrschen. Bleiben Sie ruhig bei Ihrem Lieblingseditor. Sollte dieser aber auf den Namen `“ed”` hören, dann versuchen Sie sich doch mit `“MEMACS”` anzufreunden, der zum Lieferumfang der Workbench 1.3 gehört und die Arbeit wesentlich erleichtern kann, u.a. weil zwei Texte gleichzeitig darstellbar sind, z.B. ein Quelltext und das korrespondierende Assemblerprotokoll.

Für alle, die bereits in Maschinensprache programmiert haben, folgt nun ein langweiliger Abschnitt, der zudem an der Oberfläche bleibt. Wir geben nur wesentliche Informationen zur Programmierung der CPU 68000, um eine Ausgangsbasis zu schaffen, die im Verlauf der Folgen erweitert werden wird.

Wie sagen wir es der CPU?

Die CPU MC 68000 bietet 16 Register und einen Programmzähler, jeweils mit einer Breite von 32 Bits. Die ersten acht Register mit den Bezeichnungen D0 bis D7 (bzw. d0 bis d7) sind Datenregister, die sowohl in ihrer gesamten Breite (Bits 31 bis 0) als auch wortweise (Bits 15 bis 0) oder byteweise (Bits 7 bis 0) eingesetzt werden können.

Weiterhin gibt es sieben Adreßregister mit den Bezeichnungen A0 bis A6 (a0 bis a6) und den Stapelzeiger (stack pointer, sp), der auch ein Adreßregister (a7) ist, das auf die nächste freie Adresse des Stapels zeigt. Der Stapel wächst wie ein Eiszapfen (und das in der Wüste) von oben nach unten, das heißt in einem reservierten Speicherbereich von seiner höchst möglichen Adresse auf die niedrigste zu. (Was ist eine Adresse? – Die Bytes des Amiga sind durchnummeriert. Eine Adresse ist die Nummer, unter der ein Byte, genauer der Speicherinhalt eines Bytes, angesprochen werden kann.)

Neben den genannten 16 Registern gibt es noch ein Statusregister. Die CPU kennt zwei Betriebsarten, den `“User Mode”` und den `“Supervisor Mode”`. Den `“User Mode”`, als Betriebsart voreingestellt, wollen wir nicht verlassen. Im `“Supervisor Mode”` kann auf einen weiteren Stapelzeiger und auf die zweite Hälfte des Statusregisters (sr) zugegriffen werden. Auch die CPU kennt Datentypen, nicht nur Hochsprachen. Die Typen sind:

- Bit (null oder eins)
- BCD (binär kodierte Dezimalziffern, vier Bits)
- Byte (acht Bits breit, mit `“.b”` markiert)
- Word (16 Bits breit, mit `“.w”` markiert und ohne diesen Zusatz so angenommen)

– LongWord (32 Bits breit, mit `“.l”` bezeichnet)

Kommen wir zurück zu unserem ersten Programm. Durch den Zusatz `“.l”` hinter dem Befehl `“move”` haben wir die CPU aufgefordert, ein Langwort zu übertragen.

Was macht eine CPU eigentlich? – Im wesentlichen sind es drei Aufgaben, nämlich Daten holen, diese gegebenenfalls manipulieren (z.B. zu einem Registerinhalt adressieren) und diese abspeichern. Wie kommt man an die Daten heran? – Nun, dazu gibt es vierzehn raffinierte Adressierungstechniken, die sich in sechs Arten gliedern. Tabelle 1 gibt einen Überblick.

Rückblickend auf unser erstes Beispiel können wir spätestens jetzt feststellen, die Adressierungsart `“Datenregister direkt”` verwendet zu haben. Zum Grundverständnis fehlt uns noch der Befehlssatz, den die CPU versteht, und dann können wir loslegen. Tabelle 2 stellt häufig benötigte Instruktionen dar, also einen Ausschnitt aus dem Befehlssatz, der unseren Ansprüchen genügen sollte. Der gesamte Umfang ist in jedem Assemblerbuch dargestellt, z.B. in [1] bis [4]. Tabelle 2 ist in Verbindung mit Tabelle 3 zu sehen, wenn es um die bedingten Verzweigungen (Bcc) geht.

Es gibt 16 verschiedene Möglichkeiten, die wir für unsere Arbeit jedoch nicht ausschöpfen werden. Schließen wir das Kapitel mit einem Hinweis ab. Assemblerprogramme sind schnell. Sollten Sie Spaß daran haben, die Programmausführung zeitlich voll in den Griff zu nehmen, dann empfehlen wir Ihnen das Studium der Druckschrift von Motorola [1] zur CPU. Sie können dort u.a. ab Seite 7-5 erfahren, daß Adressierungsarten sich auf den Zeitbedarf für die Ausführung von Instruktionen auswirken, und daß Divisionen etwa doppelt soviel Zeit in Anspruch nehmen wie Multiplikationen. Mit den dargestellten Grundlagen wollen wir uns nun an den Amiga heranwagen.

Zugang zu den Systemroutinen

Müßten wir bei der Assemblerprogrammierung jedes Bit

selbst setzen, dann hätten wir uns jetzt mit der Hardware auseinanderzusetzen, die unseren Bildschirm bedient oder von der Tastatur liest. Der Amiga stellt wie jeder andere Computer Systemroutinen zur Verfügung, die uns viele nützliche Funktionen als Bausteine für eigene Programme bieten. Ein Teil der Systemroutinen befindet sich in Festwertspeichern oder ist als Kickstart (Amiga 1000) zu laden. Andere verstecken sich im Verzeichnis `“LIBS”` der Workbenchdiskette. Die Systemroutinen sind in Bibliotheken (engl. libraries) zusammengefaßt. Der Amiga lädt die benötigten und nicht integrierten Systemroutinen selbstständig von einem externen Speichermedium, vorausgesetzt sie befinden sich im Verzeichnis `“LIBS”`. Bei Speichermangel werden die ladbaren Routinen selbsttätig aus dem Speicher entfernt, falls nicht mehr benötigt.

Wie findet man sich nun in den Bibliotheken zurecht, und wie benutzt man die darin enthaltenen Systemroutinen? – Einen recht guten Einblick in die Systemroutinen vermittelt z.B. [6]. Wir werden im wesentlichen einzelne Routinen der Bibliotheken DOS, Exec, Graphics und Intuition einsetzen und diese bei der praktischen Anwendung besprechen. Sollten Sie über die `“EXTRAS”`-Diskette zur Workbench 1.3 verfügen, dann werfen Sie doch einmal einen Blick auf die FD-Files (function description), die den Inhalt der einzelnen Bibliotheken darstellen. Ein typischer Eintrag der Bibliothek FD1.3/exec_lib.fd lautet:

```
CloseLibrary(library)(A1)
```

Der Eintrag bedeutet: Schließen Sie die Bibliothek, deren Adresse im Register A1 enthalten ist. Wir probieren das gleich aus.

In der Folge vier werden wir aus den FD-Files Dateien (Include Files) erzeugen, die den Zugriff auf Systemroutinen erheblich erleichtern. Es wäre deshalb von Vorteil, wenn Sie sich in den nächsten Wochen die `“EXTRAS”`-Diskette beschaffen würden. Zu jeder öffentlichen Bibliothek muß man sich Zutritt verschaffen. So ist es im übertragenen Sinne auch beim Amiga. Jede Bibliothek ist zu öffnen, bevor daraus eine Routine einge-

Adressierungsarten	Syntaxbeispiel
Register direkt - Datenregister - Adressregister	d0 a5
absolut - kurz - lang	\$6F0
Prog. zähler relativ - mit Offset - m. Offset u. Index	12(a7) 12(a7,d0)
Adressreg. indirekt - Register - Reg. postincrement - Reg. predecrement - mit Offset - mit Offs. + Index	(a1) (sp)+ -(sp) d0(a5) d0(a1,d1)
Konstanten - immediate - quick immediate	#16 #1 bis #8
implizit	pc, sr, sp

Tabelle 1 Adressierungsarten der CPU 68000

Befehl	Bedeutung	Beispiel
add	addiere	add.b #27,d0
and	logisches UND	and.l d0,d1
asl/asr	arithmet. shift	asl.l #3,d0
Bcc (*)	verzweige bedingt	bcc ende
bra	verzweige immer	bra ende
clr	lösche Operand	clr.w d1
cmp	vergleiche	cmp.l d0,d1
eor	exklusives ODER	eor.w d1,d3
jmp	springe	jmp ende
jsr	spr. zur Subrout.	jsr ZickZack
lea	lade effekt. Adr.	lea.l Intui,a6
lsl/lsr	logisches shift	lsr.b #3,d0
move	lade	move.l (a0),d1
nop	tue nichts	(am Sonntag)
not	Einerkomplement	not d0(a1)
or	logisches ODER	or.b (a3)
rts	verlasse Subrout.	(wie in Basic)
sub	subtrahiere	sub.l d0,(a3)
tst	teste null/neg.	tst.l d1

Tabelle 2 Häufig benötigte 68000-Instruktionen
(*) Bedingungen gem. Tabelle 3

setzt werden kann. Eine Ausnahme bildet lediglich die Bibliothek Exec. Sie ist stets geöffnet, weil ohne sie nichts laufen würde, und weil sie den Schlüssel zu den anderen Bibliotheken enthält.

So wie in jeder ordentlichen Bibliothek die Bücher durchnummeriert sind, verfährt man auch für die Systemroutinen. Es werden negative Zahlen benutzt, die einen Versatz (engl. offset) zur Anfangsadresse der jeweiligen Bibliothek darstellen. Die negativen Zahlen bilden eine Sprungtabelle, die bei minus 30 beginnt. Die Einträge sind jeweils um sechs Bytes versetzt. Gültige Sprungvektoren sind z.B. -36 und -102.

Mit dem Aufruf einer Systemroutine wird bestimmt, was zu tun ist und womit. Es wird nämlich erwartet, daß an die Routine Parameter in Registern übergeben werden. Im obigen Beispiel "CloseLibrary" ist ein Parameter in A1 zu übergeben. Systemroutinen liefern zumeist Ergebnisse, die dem Register D0 und bei Bedarf zusätzlich noch D1 zu entnehmen sind.

Fassen wir zusammen: Die Systemroutinen des Amiga bilden wesentliche Bausteine unserer Programme. Die Routinen werden mit Hilfe von Sprungvektoren aufgerufen. Dabei sind Parameter in Registern zu übergeben. Systemroutinen liefern Ergebnisse in

D0 und D1. Genug der Theorie! Wir stürmen jetzt eine Bibliothek, ohne darin etwas anzurühren, und verlassen sie wieder gekonnt. Listing 3 zeigt den Plan dazu.

Wir definieren den Beginn der Bibliothek Exec an der Adresse vier und die Sprungvektoren der Routinen für das Öffnen und Schließen von Bibliotheken. Die Adresse vier ist die einzige feststehende im Amiga für Software. Weiterhin definieren wir im Datenbereich den Namen der zu öffnenden Bibliothek (IntuiLib) und bestimmen mit der Version 33, daß wir nicht weiter als bis zum Entwicklungsstand 1.2, (bezogen auf das Betriebssystem) zurückgehen wollen. Mit anderen Worten, jede angetroffene Version der Bibliothek "Intuition" mit einer Versionsnummer von 33 oder höher soll recht sein. Im eigentlichen Programm, ab Main, werden zunächst die Parameter für "OpenLibrary" in die Register A1 und D0 geladen. Da "OpenLibrary" eine Routine aus "Exec" ist, erhält das Register A6 die Adresse vier. Erinnern Sie sich bitte, daß die Routinen über ihren Versatz zur Basisadresse der jeweiligen Bibliothek angesprochen werden.

Durch die Adresse im Register A6 wird dem Betriebssystem bekanntgemacht, welche Bibliothek gemeint ist. Zurück zu Listing 3. Um Intuition zu öffnen, benutzen

wir mit der Instruktion "jsr OpenLibrary(a6)" die Adressierungsart "Register indirekt mit Offset". Die Routine "OpenLibrary" liefert die Basisadresse von Intuition oder null, falls uns kein Erfolg beschieden war. Dieses testen wir mit "tst.l d0". Die Fortsetzung eines Programms hätte keinen Sinn, wenn uns die zu öffnende Bibliothek nicht zur Verfügung stünde.

Ist D0 gleich null (zero flag im Statusregister gesetzt), dann ist zum Programmende zu verzweigen, sonst sichern wir die Adresse übungshalber. Dann schließen wir die Bibliothek wieder. Geöffnete, und nur erfolgreich geöffnete, Bibliotheken sind bei Programmende stets zu schließen. Das eigentliche Pro-

gramm ist zwischen dem Öffnen und Schließen einzufügen. Wer hätte das gedacht?

Fassen wir zusammen: Bibliotheken sind vor dem erstmaligen Gebrauch zu öffnen und spätestens bei Programmende zu schließen. Die Bibliothek Exec ist stets geöffnet. Für das Öffnen und Schließen stehen die Routinen "OpenLibrary" und "CloseLibrary" aus Exec zur Verfügung. Bei der Benutzung einer Bibliothek weist das Register A6 auf die Basis. Register D0 ist nach dem Öffnen auf null zu testen, um sicher zu sein, daß die benötigten Routinen verfügbar sind.

Da auch unser zweites Programm ohne Aufsehen zu ereignen abläuft, wäre es doch schön, einen Beweis für die

Bedingung	Bedeutung
cc	carry clear
cs	carry set
eq	equal
ge	greater equal
gt	greater than
hi	high
le	less
ls	low or same
lt	less than
mi	minus
ne	not equal
pl	plus
vc	no overflow
vs	overflow

Tabelle 3 : Condition Codes im Überblick, auswertbar durch Interpretation des Statusregisters

Funktionsfähigkeit zu erhalten. Und nun sind Sie gefordert: Rufen Sie die Routine "DisplayBeep" vor dem Schließen von Intuition auf. Dazu müssen Sie lediglich das Register D0 mit Null sowie das Register A6 mit der Basisadresse von Intuition laden und dann "DisplayBeep" entsprechend "OpenLibrary" aufrufen. Hat es geklappt? – Die Lösung finden Sie in Listing 3a auf Seite

Ein Programmiermodell

Bevor wir so richtig loslegen, wollen wir noch wichtige Rahmenbedingungen für unsere Arbeit abklären:

Registerkonventionen

Wie bereits bekannt, bildet die Bibliothek "Exec" mit der festen Adresse "vier" die Basis des Betriebssystems. Alle anderen Bibliotheken sind explizit zu öffnen. Die enthaltenen Systemroutinen oder Daten sind über Sprungvektoren zu erreichen, mit der Basisadresse der Bibliothek in A6. Folgen Sie den Konventionen, und verwenden Sie kein anderes Adressregister für diesen Zweck. Die Register A0, A1 sowie D0, D1 können von Systemroutinen verändert werden. Sie eignen sich also nicht, um Werte zu speichern, die nach zeitlichem Aufruf einer Systemroutine weiterverarbeitet werden sollen. Die beiden Datenregister (hauptsächlich d0) enthalten Funktionsergebnisse von Systemroutinen.

Die Gültigkeit von Funktionsergebnissen ist explizit zu testen (z.B. mit `tst` oder `move`) mit anschließender Abfrage des Statusregisters (z.B. mit `beq`).

Verwendung von Datenstrukturen

Datenstrukturen sollten auf geraden Adressen beginnen. Für DOS-Strukturen sind in vielen Fällen sogar Langwort-

adressen (durch vier ohne Rest teilbar) vorgeschrieben. Die entsprechende Ausrichtung auf Wort- bzw. Langwortgrenzen erreicht man z.B. mit den Assembleranweisungen "cnp 0,2" bzw. "cnp 0,4". Byteweise organisierte Strukturen, z.B. Zeichenketten, dürfen an beliebigen Adressen beginnen. Es ist von Vorteil, Datenstrukturen mit Null oder anders zu initialisieren, um bei Programmstart mit einem definierten Zustand zu beginnen.

Verwendung von Adressen

Von der Verwendung absoluter Adressen ist abzuraten. Die Adressen der so angesprochenen Hardware können sich ändern. In unserer Routine zur Abfrage des Mausknopfes stoßen wir dagegen.

Zum Stack

Das Betriebssystem überwacht nicht die Einhaltung des Stackbereichs. Eine Überschreitung des Bereichs führt zu hässlichen Abstürzen. Dimensionieren Sie den Stack hinreichend groß, z.B. mit dem entsprechenden CLI-Befehl. Für unsere Experimente reichen die üblichen 4000 Bytes.

Programmaufbau

Wir werden alle Programme entsprechend Listing 3 strukturieren, nämlich zuerst Konstanten deklarieren, dann Unterprogramme, gefolgt vom Hauptprogramm, aufrufen und schließlich Speicherplatz für globale Daten reservieren und diese initialisieren. Aus den genannten Rahmenbedingungen sollte sich nun nach Wunsch des Autors (schemenhaft) ein Modell dessen ergeben, was bei der Programmierung zu beachten ist.

Paint Box 1.0

Bauen wir nun Listing 3 aus, um die erste Folge auch wirklich mit einem Grafikpro-

Lassen Sie sich bei Ihrer Arbeit von folgenden Fragen/Aufgaben leiten:

- Was bewirken Makros? Was erzeugen Sie? Generieren sie Code? Sehen Sie sich ein Assemblerprotokoll (-l nicht vergessen) dazu an.
- Entwickeln Sie Makros für den Aufruf von Routinen aus Graphics und Intuition analog zu `CallExec`.
- Welche Farbe hat der Punkt in der Mitte der Ellipse?
- Welche Farbe befindet sich in Farbregister Nr. 9?
- Was geschieht, wenn die Position der Ellipse so verändert wird, daß sie den Fensterrand schneidet?
- Was ist schneller, "RectFill" oder "Flood"?
- Warum sind die unteren Spitzen des Zickzackmusters nicht zu sehen?
- Ziehen Sie eine Linie von 30,40 nach 800,800. Was geschieht?

In der nächsten Folge können Sie Ihre Lösungen mit unseren Antworten vergleichen. Viel Spaß mit der Paint Box bis dahin!

gramm abzuschließen und Ihnen eine Bühne für eigene Experimente zu öffnen. Listing 4 stellt die erste Stufe der Paint Box dar. Wir werden in der nächsten Folge ausführlich darauf eingehen, bieten Ihnen aber jetzt schon die Möglichkeit, sich zielgerichtet einzuarbeiten. Versuchen Sie das Programm anhand der Kommentierung nachzuvollziehen. Die Bücher [3] und [6] können dabei helfen. Tippen Sie das Listing nicht einfach ab! Versuchen Sie zuerst die Makros "APTR" und "CallExec" zu verstehen. Ein Blick auf das Assemblerprotokoll kann dabei helfen. Erarbeiten Sie sich das Verständnis für die Unterprogramme "LibsOeffnen", "LibsSchliess" und "WarteTaste". Modifizieren Sie Listing 3 entsprechend durch Benutzung der Makros, und schaffen Sie Unterprogramme. Testen Sie den Ablauf, bevor Sie an die wirklich neuen Dinge herangehen. Ergänzen Sie Ihr Programm um die Routinen für das Öffnen und Schließen des Screen und des Window. Wenn das einwandfrei klappt, dann nehmen Sie sich bitte Stück für Stück des ersten Unterpro-

gramms ("Fuellei") vor. Schrittweises Vorgehen erleichtert es Ihnen, Fehler schnell zu erkennen. Außerdem sichert es Ihnen viele kleine Erfolgserlebnisse. Noch ein Hinweis: Die Definition der "NewScreenData" und der "NewWindowData" muß sehr sorgfältig erfolgen. Die Erklärung dazu liefern wir in der nächsten Folge.

Literaturverzeichnis:

- MC 68000/D, Advance Information 60seitiges Datenblatt, kostenlos bei Motorola, München
- 68000 Assembly Language Programming Berkeley, 1985, Klassiker mit vielen Beispielen
- Der 16-Bit-Mikroprozessor 68000, Befehlsvorrat und Eigenschaften, 1986
- Programmierung des M 68000, 1986
- Das Amiga GURU-Buch, 1989
- Amiga Intern Band II, 1989

```

100 * Einzubeziehende Textdatei, so man hat
101
102 INCLUDE incl/Exec.i
103
104 * Konstanten
105
106 wert2 EQU 8 ; Konstante mit Wert 8
107 wert3:=56 ; keine Zwischenspeicherplätze!

108 sedezimal equ $45
109 dezimal = 2345
110 oktal SET @10
111 binaer EQU %101101
112 text set 'Ami' ; entspricht EQU aber redefinierbar

```

```

113 text SET "neu"
114
115 * Symbole
116
117 Farbe equ d0 ; ein Register
118
119 * Makrodefinition
120
121 APTR macro ; reserviere vier Speicherstellen und initialisiere sie mit null
122 dc.l 0
123 endm
124
125

```



```

126          CODE          ; ohnehin vorein
127                      ; gestellt
128 L1: move.l d0,d1      ; mit Label
129 L2 move.l d0,d1      ; Label auch ohn
                        ; e ":"
130          move.l d0,d1 ; nicht am Zeile
                        ; nanfang beginn
                        ; en
131
132          move.l #1<<2,d0 ; shifte 1 zweim
                        ; al links, Erge
                        ; bnis nach d0
133          ; d.h. lade d0 m
                        ; it 4
134          MOVE.l 2*3,d0 ; 6 nach d0 ( /,
                        ; +,- arbeiten a
                        ; uch)
135          mOvE.L 3&2,d0 ; 3 AND 2 also 2
                        ; nach d0
136          move.l #2!1,d0 ; 2 OR 3 also 3
                        ; nach d0
137          move.l #-1,d0
138          move.l ~7,d0 ; Einerkomplemen
                        ; t
139          move.l d0,-(A7) ; d0 auf den Sta
                        ; ck
140          move.l (a7)+,d0 ; d0 vom Stack
141          move.l d0,-(sp) ; und noch einma
                        ; l in anderer S
                        ; chreibweise
142          move.l (sp)+,d0
143
144          move.w d0,d1 ; wortweise arbe
                        ; iten
145          move d0,d1 ; identisch zur
                        ; Vorzeile
146
147          move.l *,d1 ; Programmzähler
                        ; nach d1
148
149          ifeq 0 ; zur Steuerung
                        ; des Assemblers
                        ; auch:
150          ; ifne, ifgt, if
                        ; ge, iflt, ifle
151          Move.l d0,d1
152          endc
153 ende      rts
154
155          DATA Unsere Daten ; Name muß sein
156
157          cnop 0,4
158          dc.b 20,$55 ; 20 Bytes mit $
                        ; 55
159 ConstBlock
160          DC.L 45, 78, 89 ; drei Langworte
161          String DC.B 'test', 0 ; nullterminiert
162          even ; auf gerade Adr
                        ; esse gehen
163          dc.w 1 + 2 ; ein Wort, init
                        ; ialisiert mit
                        ; 3
164          Soso dc.w 34 * 7 ; Na ?
165
166          cnop 0,4 ; auf Langwortad
                        ; resse gehen
167
168          BSS uninitialisierte Daten ; verhindert Abs
                        ; peicherung
169
170          Speicher DS.l 40 ; 40 Langworte r
                        ; eservieren
171
172          end

```

Listing 1: Der Makroassembler A68k stellt sich mit wesentlichen Eigenschaften vor

```

100          move.l d0,d1 ; Register umspeichern
101          rts ; das Programm ist zuende
102
103          end ; Endmarke für den Assemb
                        ; ler

```

Listing 2 : Mit diesem Programmchen testen wir unsere Werkzeuge

```

100 * Programm DisplayBeep
101
102 * (der Code aus Listing 3a ist einzufügen, um dem Progra
mmnamen gerecht zu
103 * werden.)
104
105 * Konstanten

```

```

106
107 execBase equ 4 ; Bibliothek Exec an
                        ; fester Adresse
108 CloseLibrary equ -414 ; a1 = Adresse der B
                        ; ibliothek
109 OpenLibrary equ -552 ; a1 = Name, d0 = Ve
                        ; rsion
110
111 Version equ 33 ; Version 33 oder jü
                        ; nger, also
112 ; Betriebssystem 1.2
                        ; oder 1.3
113
114 DisplayBeep equ -96 ; a0 = ScreenAdr. od
                        ; er 0 für alle
115 ; für eine spätere L
                        ; ernkontrolle
116
117 jsr Main ; überspringe Unterp
                        ; rogramme
118
119 * Unterprogramme
120
121 * Hauptprogramm
122
123
124 Main lea.l IntuiLib,a1 ; Intuition wollen w
                        ; ir öffnen
125          move.l #Version,d0 ; Version 33 oder jü
                        ; nger bitte
126          move.l execBase,a6 ; Basis der Biblioth
                        ; ek holen
127
128          jsr OpenLibrary(a6) ; Intuition öffn
                        ; en
129          tst.l d0 ; gültige Adresse er
                        ; halten ?
130          beq End ; leider nein
131
132          move.l d0,IntuiBase ; Bibliotheksadresse
                        ; sichern
133
134          ; und hier etwas tun
135
136          move.l IntuiBase,a1 ; Intuition Adr. hol
                        ; en
137          move.l execBase,a6 ; wir wollen Exec nu
                        ; tzen
138          jsr CloseLibrary(a6) ; Intuition wird
                        ; geschlossen
139
140          End rts ; unser Programm ist
                        ; zuende
141          ; läßt Amiga nicht a
                        ; bstürzen
142
143 * Globale Daten
144
145
146 IntuiLib dc.b 'intuition.library',0 ; Ende String
                        ; s mit 0 mar
                        ; kiert
147          cnop 0,4 ; Auf Langwort ausri
                        ; chten
148          IntuiBase dc.l 0 ; Speicherplatz für
                        ; Intuition
149          ; Basisadresse
150          end ; macht den Assemble
                        ; r glücklich

```

Listing 3 : Öffnen der Bibliothek Intuition, um die darin enthaltenen Systemroutinen einzusetzen. Das Vorgehen ist zu verallgemeinern.

```

100 .....
101
102          ; und hier etwas tun
103
104          move.l d0,a6 ; Benutzung Intuitio
                        ; n vorbereiten
105          move.l #0,a0 ; alle Screens solle
                        ; n blinken
106          jsr DisplayBeep(a6) ; und jetzt wird geb
                        ; linkt
107
108          move.l IntuiBase,a1 ; Intuition Adr. hol
                        ; en
109
110 .....

```

Listing 3a : Und so bringen wir die Screens zum Blinken. Wie sieht Ihre Lösung aus? Kürzer? Gut so, aber seien Sie vorsichtig.


```

100 * Programm Paint Box 0.1
101
102 * aus Exec:
103 execBase      equ 4
104 CloseLibrary  equ -414
105 OpenLibrary   equ -552
106
107 * aus Intuition:
108 CloseScreen  equ -066
109 CloseWindow  equ -072
110 OpenWindow   equ -204
111
112 OpenScreen  equ -198
113
114 * aus Graphics:
115 Draw      = -246
116 DrawEllipse = -180
117 Flood     = -330
118 Move      = -240
119 RectFill  = -306
120
121 SetAPen   = -342
122 WritePixel = -324
123
124 * Hardware:
125 PRA:      equ $bfe001
126 LButton:  equ 6
127
128 * Register:
129 XPos      equ d6
130 Farbe     equ d5

```

```

129
130 APTR MACRO
131
132 dc.l 0
133
134 ENDM
135
136 CallExec MACRO
137
138 move.l execBase,a6
139 jsr    \1(a6)
140
141 ENDM
142
143 OpenLib MACRO
144
145 lea    \1,a1
146
147 moveq  #33,d0
148
149 CallExec OpenLibrary
150 move.l d0,\2
151
152 ENDM
153
154 CloseLib MACRO
155
156 move.l \1,d0
157
158 ENDM
159
160 CallExec CloseLibrary
161 nop

```

Amigadrives

- 1 Jahr Umtauschgarantie
- Anschlußfertig mit Kabel
- Amiga-farbenes Metallgehäuse
- Automatische Diskchangeerkennung
- Abschaltbar
- 100 % kompatibel
- 5,25" 40/80-Track schaltbar

Busdurchführung bis DF3:

NEC 1037 A

mit deutscher Seriennummer

IDS 3,5	218,-
IDS 5,25	249,-
IDS 3,5 intern	149,-

PCXT-Karte	789,-
Turboboard	675,-

Autoboot Filecards für A-2000
Autoboot Festplatten für A-500
auf Anfrage



Hard & Software GmbH
Frohnberg 23, 6921 Epfenbach

Speicher

- ICs der führenden Hersteller.
- Jedes Gerät einzeln geprüft.

für Amiga 500

- Intern
- Abschaltbar
- Schnelle Migabitchips

512 KB	199,-
1.8 MB	775,-
Minimax	389,-

für Amiga 1000

- Externes Metallgehäuse
- Abschaltbar

2 MB Golembox	825,-
----------------------	--------------

für Amiga 2000

- Intern im Amigaslot

8 MB-Karte mit 2 MB	995,-
----------------------------	--------------

07263/5693

Tages- und Händlerpreise erfragen!

07263/5693


```

154 ENDM
155
156 DoRpFunct macro           ; Führe eine Gra
                             ; fiktionsfunktion au
                             ; f einem
157                             ; Rastport (Rp)
                             ; aus.
158                             ; Param: Funktio
                             ; n
159                             move.l Window,a0    ; Window-Adresse
                             ; laden
160                             move.l 50(a0),a1     ; Rastport-Adres
                             ; se holen
161                             move.l GraphBase,a6  ; Bibliotheksbas
                             ; is
162                             jsr    \1(a6)        ; funktion ausfü
                             ; hren
163 endm
164
165 SetzePen macro            ; Param: XPos, Y
                             ; Pos
166                             move.w \1,d0        ; XPos in Parame
                             ; terreg.
167                             move.w \2,d1        ; YPos in Parame
                             ; terreg.
168                             DoRpFunct Move      ; Stiftposition
                             ; setzen
169 endm
170
171 FaerbeAPen macro         ; Farbregistern
                             ; ummer
172                             move.w \1,d0        ; Farbe ins Par
                             ; ameterregiste
                             ; r
173
174 DoRpFunct SetAPen
175
176 SetOpen macro            ; Farbnummer
177                             move.l Window,a0    ; Window-Adress
                             ; e laden
178                             move.l 50(a0),a1     ; Rastport-Adres
                             ; se holen
179                             move.b \1,27(a1)    ; Outlinefarbe
                             ; in Rp aendern
180 endm
181
182 LineTo macro              ; Param: xPos,
                             ; yPos
183                             move.w \1,d0        ; XPos laden
184                             move.w \2,d1        ; YPos laden
185                             DoRpFunct Draw      ; Linie zeichnen
186 endm
187
188 Ellipse macro             ; Param: XM, YM,
                             ; Xr, Yr
189                             move.w \1,d0        ;
190                             move.w \2,d1        ;
191                             move.w \3,d2        ;
192                             move.w \4,d3        ;
193                             DoRpFunct DrawEllipse
194 endm
195
196 Punkt macro              ; Param XPos, YP
                             ; os
197                             move.w \1,d0        ;
198                             move.w \2,d1        ;
199                             DoRpFunct WritePixel
200 endm
201
202 FlaecheFuell macro       ; Param: Mode, X
                             ; Pos, YPos
203                             move.w \1,d2        ;
204                             move.w \2,d0        ;
205                             move.w \3,d1        ;
206                             DoRpFunct Flood
207 endm
208
209 RechteckGefuell macro    ; Param: X1, Y1,
                             ; X2, Y2
210                             move.w \1,d0        ;
211                             move.w \2,d1        ;
212                             move.w \3,d2        ;
213                             move.w \4,d3        ;
214                             DoRpFunct RectFill
215 endm
216
217
218 jsr Main                  ; Sprung über Un
                             ; terprogramme z
                             ; um
219                             ; Hauptteil
220
221 * Unterprogramme
222
223 LibsOeffnen OpenLib GraphLib,GraphBase
224 beq \1
225 OpenLib IntuiLib,IntuiBase
226 rts
227
228 LibsSchliess CloseLib IntuiBase ; Intuition Libr
                             ; ary
229 CloseLib GraphBase ; Graphics Libr
                             ; ary
230 rts
231 ScreenAnlegen move.l IntuiBase,A6 ; Basisadresse I
                             ; ntuition Library

```

```

232 lea NewScreenData,A0 ; so soll Sc
                             ; reen ausse
                             ; hen
233 jsr OpenScreen(A6); Screen oeffnen
234 move.l D0,Screen ; Adresse Screen
                             ; bewahren
235
236 rts
237 ScreenSchliess move.l Screen,d0 ; Adresse des Sc
                             ; reen
238 beq \S ; falls null dan
                             ; n nichts tun
239 move.l d0,a0 ; Screen Adr ins
                             ; Parameterreg.
240 jsr CloseScreen(a6) ; Screen schli
                             ; eßen
241 \S
242 rts
243 WindowAnlegen move.l Screen,d0 ; Adresse Screen
                             ; in Window-Be-
244 move.l d0,nw_screen ; schreibung ein
                             ; setzen
245 lea NewWindowData,a0 ; so soll Wi
                             ; ndow ausse
                             ; hen
246 move.l IntuiBase,a6 ; Basisadresse I
                             ; ntuition Libr
                             ; ary
247 jsr OpenWindow(a6) ; Window oef
                             ; fnen
248 move.l d0,Window ; Adresse Window
                             ; bewahren
249
250 rts
251 WindowSchliess move.l Window,d0 ; wird zu MOVEA,
                             ; wenn direkt i
                             ; n A0
252 beq \W ; falls null dan
                             ; n nichts tun
253 move.l d0,a0 ; Window Adr. in
                             ; s Parameterreg
254
255 move.l IntuiBase,a6 ; Basisadresse I
                             ; ntuition Libr
                             ; ary
256 jsr CloseWindow(A6) ; Schliesse Wi
                             ; ndow
257 \W
258 rts
259 WarteTaste btst #LButton,PRA ; Warten auf Tas
                             ; tendruck
260 bne WarteTaste ; Warteschleifen
                             ; vergeuden
261 rts ; R e c h e n z
                             ; e i t !!!!
262
263 FaerbeAPen #14 ; Unterprog. dem
                             ; onstriert eini
                             ; ge
264 SetzePen #10,#20 ; Grafikfaehigke
                             ; iten
265 LineTo #100,#200
266 FaerbeAPen #9
267 RechteckGefuell #380,#70,#500,#200
268 FaerbeAPen #15
269 Ellipse #200,#110,#200,#100
270 SetOpen #15
271 FlaecheFuell #0,#200,#110 ; Ellipse fu
                             ; llen
272 FaerbeAPen #2 ; Geduld !!!
273 rts ;!!!!
274
275 ZickZack move.w #630,XPos ; XPos bestimmen
                             ; als Ausgangsp
                             ; unkt
276 SetzePen XPos,#350 ; Zeichenstift a
                             ; uf Ausgangspun
                             ; kt
277 move.w #15,Farbe ; höchstmögliche
                             ; Farbnummer, v
                             ; on der nur
278 ; untere 4 Bits
                             ; benötigt für 1
                             ; 6 Farben
279 FaerbeAPen Farbe ; Farbe setzen
280
281 Zick LineTo XPos,#0 ; es geht nach o
                             ; ben
282 sub.w #1,Farbe ; Farbe ändern
283 FaerbeAPen Farbe ; Farbe setzen
284 sub.w #10,XPos ; neue Position
                             ; setzen
285 beq Zack ; falls am Ende
286 LineTo XPos,#350 ; und nun bergab
287 sub.w #10,XPos ; neue Position
                             ; setzen
288 bne Zick ; noch einmal
289 rts ; endlich
290
291
292 * Hauptprogramm
293
294
295 Main: jsr LibsOeffnen ; alles oeffnen,
                             ; Reihenfolge

```



```

296      beq      endeL      ; beachten
297      jsr      ScreenAnlegen
298      beq      endeS
299      jsr      WindowAnlegen
300      beq      endeW
301
302      jsr      FuehleEi      ; unser eigentli
                           ches Programm
303
304      jsr      ZickZack
305
306      jsr      WarteTaste      ; Zeit zum Staun
                           en, nicht mult
                           itas-
                           ; king-gerecht r
                           ealisiert
307
308 endeW      jsr      WindowSchliess ; alles schlies
                           sen, in umgek
                           ehrter
309 endeS      jsr      ScreenSchliess ; Reihenfolge w
                           ie geoeffnet
310 endeL      jsr      LibsSchliess
311
312      rts              ; Programm endet
                           hier
313
314
315 * Globale Daten
316
317      cnop 0,4          ; auf Langwort a
                           usrichten
318 GraphBase:  APTR      ; Basisadresse d
                           er Grahpics-Li
                           brary
319 IntuiBase:   APTR      ; Basisadresse v
                           on Intuition
320 Screen:     APTR      ; Adresse des S
                           creen für spä
                           teren Gebrauch
321 Window:     APTR      ; Fensteradresse
                           später häufig
                           gebraucht
322
323

```

```

324 GraphLib:   dc.b      'graphics.library',0 ; Ende S
                           trings
                           mit 0
                           marki
                           ert
325 IntuiLib:   dc.b      'intuition.library',0
326
327      even      ; auf Wortgrenze
                           ausrichten
328 NewScreenData: dc.w      0,0,640,256,4 ; X, Y, Breite,
                           Hoehe, 4 Bitpl
                           anes
329
330      dc.b      0,1      ; Zeichen- u. B
                           lockstift
331      dc.w      $8000,$F ; Aufloesung 640
                           , custom scree
                           n
332      dc.l      0,ScreenTitle ; Zeichensatz, T
                           itel
333
334 ScreenTitle: dc.b      ' Paint Box 0.1 ',0
335
336      even
337 NewWindowData: dc.w      0,10,640,246 ; X, Y, Breite,
                           Hoehe
338
339      dc.b      1,0      ; Zeichen- und B
                           lockstift
340      dc.l      0,0,0    ; kein IDCMP, Ac
                           tivate + Backd
                           rop
341 nw_screen:   dc.l      0      ; nach Öffnen de
                           s Screen nach
                           ragen
342      dc.l      0      ; Bitmap
343      dc.w      0,0,0,0 ; Ausdehnung min
                           /max
344      dc.w      $F      ; custom screen
345
346      end

```

Listing 4 : Unser erstes Grafikprogramm bildet den Grundstock des Malprogrammes Paint Box

Diskettenlaufwerke

3 1/2" Laufwerk AMIGA 2000 intern komplett mit Einbaunit und Anleitung	DM 159,-
3 1/2" Laufwerk für alle AMIGAs extern abschaltbar, Busdurchführung, AMIGAFarben	DM 229,-
5 1/4" Laufwerk für alle AMIGAs extern abschaltbar, Busdurchführung, 40/80 Tracks, AMIGAFarben	DM 279,-

Disketten

3 1/2" No Name 2DD	10 St.	DM 16,95
3 1/2" Verbatim Verex 2DD	10 St.	DM 25,-
3 1/2" Sentinel COLOR-Disk	10 St.	DM 29,90
lieferbar in Grün, Gelb, Rot, Orange und Weiß		
5 1/4" No Name 2SD	10 St.	DM 5,90
5 1/4" Verbatim Verex	10 St.	DM 14,90

Staffelpreise auf Anfrage

Festplatten

Kapazität	Speed	Filecard	A200	A500
20MB/514"	65ms	-	798,-	998,-
20MB/31/2"	35ms	898,-	848,-	1048,-
30MB/514"	65ms	-	898,-	1098,-
30MB/31/2"	35ms	1148,-	1098,-	1298,-
40MB/514"	28ms	-	1198,-	1398,-
50MB/31/2"	35ms	1398,-	1348,-	1548,-
60MB/514"	28ms	-	1398,-	1598,-

Alle unsere Festplatten werden wahlweise mit ALF V1.6 oder mit Autoboot-Modul ausgeliefert.

AUTOBOOTMODUL für Amiga

Amiga 2000 Autobootmodul	DM 119,-
Amiga 500 Autobootmodul	DM 149,-
(Bitte Festplatten- und Controllertyp angeben)	
Festplatten-Interface	DM 99,-
Die Adapterplatte paßt den PC-BUS eines Festplatten-Controllers an den AMIGA-BUS an. (Bitte Rechnertyp angeben)	
ALF 1.6 Festplattentreiber	DM 98,-
ALF 2.0 Festplattentreiber	DM 198,-
MFM-Set	DM 349,-
OMTI 5520B, ALF V1.6, Festplatten-Interface, Kabelsatz	
RLL-Set	DM 369,-
OMTI 5527B, ALF V1.6, Festplatten-Interface, Kabelsatz	

Software

X-Copy II	DM 49,-
X-Copy II mit Hardwarezusatz	DM 69,-
Sherlock V2.0 AntiVirusTools	DM 49,-
Turboprint II	DM 89,-
RAM-Test für Amiga	DM 24,50
Quarterback (Festplatten-Backup)	DM 119,-

HK-Computer

F. Hansmann & Th. Küpper GbR
Bonner Straße 37 ** 5000 Köln 1

RAM-Erweiterung AMIGA 500 auf 1MB DM 198,-
mit Uhr, abschaltbar
dto. Platine mit Uhr & Schalter ohne RAMs DM 79,-
dto. Leerplatine mit Stecker DM 39,-

RAM-Erweiterung II AMIGA 500 auf 1MB DM 239,-
Megabittechnologie, mit Uhr, abschaltbar
dto. Platine mit Uhr & Schalter ohne RAMs DM 79,-
dto. Leerplatine mit Stecker DM 35,-

AMIGA-Bremse * der Highscore-Killer * DM 39,50
- regelt die Geschwindigkeit stufenlos bis zum Stillstand
- ideal für schnelle Games und Bildschirmfotografie

BOOT-Selector für Amiga DM 14,50
wahlweise Booten von DF0: oder DF1: oder DF2: oder DF3:
bei Bestellung bitte angeben

Drive-Expander DM 39,-
- für externe Laufwerke ohne Busdurchführung, einstellbare Laufwerksnummer, keine Kabellängenprobleme, abschaltbar
bei Verwendung eines beliebigen Boot-Selectors kann von jedem Laufwerk gebootet werden

Maus & Joystick-Adapter DM 44,50
- für gleichzeitigen Anschluß von Maus und Joystick, mit LED-Anzeige, alle Maussteuerleitungen elektronisch geschaltet

Modem Set Discovery 1200C + DM 298,-
- inkl. AMIGA-Anschlußkabel und DFÜ-Software

Modem Set Discovery 2400C DM 448,-
- inkl. AMIGA-Anschlußkabel und DFÜ-Software
beide Modems sind Exportmodelle ohne Postzulassung! Inbetriebnahme strafbar

BTX/VTX Decoder mit FTZ-Zulassung DM 248,-
- (neuer Power-LED Träger mit integrierten 7-Segmentanzeigen)

Trackdisplay A2000 intern DF0: & DF1: DM 98,-
(neuer Power-LED Träger mit integrierten 7-Segmentanzeigen)

Trackdisplay extern DF0: bis DF3: DM 79,-
für jedes Laufwerk einstellbar

Kick-ROM DM 49,-
- Kickstartumschaltplatte für zwei OriginalROMs
OriginalROM 1.2 oder 1.3 DM 65,-
Kick-ROM mit einem OriginalROM DM 98,-
KickstartEproms 1.2/1.3/Guardian DM 99,-
Drucker, Farbbänder, Kabel, Staubschutzhauben und Software auf Anfrage

Telefon: 02 21/31 16 06 * Telefax: 02 21/32 11 66
Mo-Fr. 10.00-13.30 u. 14.30-18.30, Sa. 10.00-14.00
Stadtsparkasse Köln, Kto 6342133, BLZ 37050198

»PowerFire« Das Superding! DM 29,50
Dauerfeuerinterface für Joystick und Maus
- optimale Impulsfolge für jedes Game einstellbar
- Dauerfeuer wird über Feuer- bzw. Maustaste aktiviert
- einfach zwischen Maus/Joystick und Rechner stecken
- abschaltbar

Neu ab Dezember • Neu ab Dezember • Neu ab Dezember

RAM-Erweiterung AMIGA 500 auf 2,3MB DM 698,-
mit Uhr, abschaltbar, intern!
dto. Platine mit Uhr&Schalter ohne RAMs DM 198,-

RAM-Erweiterung AMIGA 2000 DM 848,-
8MB mit 2MB bestückt, autokonfigurierend, einfach durch zusätzliche RAMs
und Jumper umstecken aufrüstbar, keine neuen PALs erforderlich
dto. Platine teilbestückt ohne RAMs DM 498,-
RAM-Satz für 2 Megabyte DM 498,-

Kickstartumschaltplatte 3-fach DM 59,-
- für zwei OriginalROMs und eine Epromversion
Umschaltplatte mit einem Epromsatz DM 155,-
Umschaltplatte mit einem OriginalROM DM 108,-

Genlock-Interface DM 248,-
nur Computer/nur Video/beide Bilder gemischt durchgeführter Monitorport,
Positiv/Negativ
externes Gehäuse

Professional View DM 998,-
Video-Digitizer der neuen Generation
On Board RGB-Splitter bereits integriert !!
On Board Genlock-Interface bereits integriert !!
Realtime-Digitalisierung in 16 Graustufen, 25 Bilder/s
HAM-Modus mit 4096 Farben
kurze Realtime Animationen in s/w, Übertragung 3,5MB/s !
Alle Bilder und Animationen auf Diskette speicherbar
für Animationen mind. 1MB RAM
unterstützt 1MB ChipRAM

Neu ab Dezember • Neu ab Dezember • Neu ab Dezember

Haben Sie Hard- oder Software für den Amiga entwickelt ?
Wir bieten Ihnen eine großzügige Umsatzprovision und eine ehrliche Abrechnung.

»»»Sprechen Sie uns an«««

Wir reparieren Ihren Amiga und Zubehör schnell und preisgünstig.

UPS-Versand: Nachnahme 10 DM, Vorauskasse 5 DM,
Großgeräte nach UPS-Tabelle ohne Aufschlag, Ausland nur gegen Vorauskasse + 15DM

Fordern Sie unser kostenloses Info an
Händleranfragen erwünscht



Michael Winterberg

Soundeditor in Assembler

Grundlagen der Soundprogrammierung

Wer kennt Sie nicht, diese hervorragenden Musikstücke und Soundeffekte, die uns bei Computerspielen meist begleiten? Gerade beim Amiga fallen diese Kompositionen dabei durch ihre Brillanz auf. Schlagzeuge, E-Gitarren und viele andere Instrumente sowie diverse Geräuschkulissen werden lebensecht imitiert. Doch wie kommen all diese Sachen in den Rechner? Wie man seinen Amiga dazu bringt, Sounds und Musiken auszugeben, können Sie in unserem kleinen Musikkurs schnell und leicht erlernen.

Die einfachste Art der Computermusikmacherei ist die Programmierung mittels Musikeditor. Solche Editoren sind mittlerweile auch schon für wenig Geld in der Public Domain erhältlich.

Gewiß, es gibt auch die Möglichkeit, Noten in einem Assembler-Source-Code unterzubringen. Diese Methode ist jedoch nicht allzu sinnvoll, da für jede neue Musik leider auch der Source-Code geändert werden müßte. In unserer Serie widmen wir uns der Musikeditor-Programmierung. Beginnen werden wir zu-

nächst mit den Grundlagen: Sämtliche Beispiele wurden übrigens mit dem Seka-Assembler geschrieben.

Wie arbeitet ein Editor? Da stellen wir uns einmal ganz dumm

Im Prinzip arbeitet ein Editor immer nach dem gleichen Schema:

Dateneingabe, Datenverarbeitung und Datenausgabe. In unserem Fall wäre dies die Eingabe der Noten, die Verar-

beitung und dann die Ausgabe eines entsprechenden Tones über die Audiokanäle des Amiga.

Jedes Böhnchen gibt ein Tönchen – die Tonausgabe auf dem Amiga

Es gibt zwei verschiedene Möglichkeiten zur Ausgabe von Tönen. Einmal können Töne ausgegeben werden, indem man direkt die Audio-Hardware-Register programmiert; die zweite, wesentlich einfachere Möglichkeit ist die Benutzung des sogenannten Audio-DMA's (DMA = Direct Memory Access), der einen direkten Speicherzugriff auf den Soundchip des Amiga vornimmt, ohne den 68000er zu belasten.

Diese einfache Methode ist am besten zur Datenausgabe geeignet, daher werden wir sie für unsere Beispiele später auch benutzen. Was macht ein Audio-DMA im einzelnen?

Nun, er liest die digitalen Daten der Töne in einer bestimmten Geschwindigkeit

aus dem Speicher und gibt diese in einer bestimmten Lautstärke durch den Digital/Analog-Wandler auf die Audio-Kanäle aus.

Der Chip, der für die Soundeffekte zuständig ist, liebevoll Paula genannt, enthält vier Audio-Kanäle, die als Stereoausgänge gekennzeichnet sind und über neun Oktaven verfügen.

Alle vier Kanäle können gleichzeitig vom Audio-DMA bearbeitet werden. So können Sie beispielsweise vier verschiedene Töne parallel ausgeben lassen. In den nächsten Zeilen erfahren Sie, wie man einen Ton über einen DMA ausgibt. In unserem Beispiel wählen wir einfach mal den Audio-DMA 0.

Das DMACON-Register... – "Die Macht ist in ihm!"

Das DMACON-Register (\$DFF096, Direct-Memory-Access-CONTROL-Register) hat sozusagen die "Macht" über alle Audio-DMA's. Sie können mit diesem Register unter anderem die einzelnen Audio-DMA's an- bzw. ausschalten, indem Sie die entsprechenden Bits setzen bzw. löschen. Hier die Aufteilung der einzelnen Bits des DMACON-Registers:

Bit 0	AUD0EN	Audio-DMA 0
Bit 1	AUD1EN	Audio-DMA 1
Bit 2	AUD2EN	Audio-DMA 2
Bit 3	AUD3EN	Audio-DMA 3
Bit 4	DSKEN	Disk-DMA
Bit 5	SPREN	Sprite-DMA
Bit 6	BLTEN	Blitter-DMA
Bit 7	COPEN	Copper-DMA
Bit 8	BPLEN	Bit-Plane-DMA
Bit 9	DMAEN	Gesamt-DMA
Bit 10	BLIPRI	Blitter-DMA (Priorität über Prozessor)
Bit 11	unbelegt	
Bit 12	unbelegt	
Bit 13	BZERO	Ergebnis aller Blitteroperationen
Bit 14	BBUSY	Zeigt an, ob der Blitter gerade arbeitet
Bit 15	SET/CLR	Bit setzen bzw. löschen

Von diesen 16 Bits sind für uns eigentlich nur folgende von Bedeutung: Bit 0 bis Bit 3 für die Audio-DMA's, Bit 9 für den Gesamt-DMA und Bit 15, mit dem wir die einzelnen Bits setzen oder löschen können.

Wollen wir zum Beispiel den Audio-DMA 0 anschalten, so

müssen wir Bit 0 und Bit 15 setzen. Tabelle 1 verdeutlicht die Berechnung der Hexadezimal-Werte, die die einzelnen Bits setzen.

Nach Tabelle 1 ergibt das Anschalten des Audio-DMA 0 also einen Wert von Hex\$ 8001. Um sicherzugehen, daß der DMA wirklich gestartet wird, sollten wir unbedingt auch den Gesamt-DMA anschalten, also auch Bit 9 setzen. Das ergibt dann nach Tabelle 1 einen Wert von Hex\$ 8201. Die Assembler-Zeile, die den Audio-DMA 0 anschaltet müßte also wie folgt aussehen:

```
1 MOVE.W #?$8201,$DFF096
```

Die Audio-DMA-Register: Bitte ein Bit!

Bevor wir jedoch einen Audio-DMA anschalten, müssen wir ihm auch mitteilen, wie er einen bestimmten Ton ausgeben soll. Dazu gibt es folgende vier Register für jede Stimme:

- AUD?LC für die Startadresse der auszugebenden digitalen Daten,
- AUD?LEN für die Länge der digitalen Daten in Worten,
- AUD?VOL die Lautstärke,
- AUD?PER die SamplePeriod, also die Abspielgeschwindigkeit für die digitalen Daten,

Das ? steht für einen Audio-DMA (0 bis 3).

Tabelle 2 zeigt die einzelnen Speicheradressen für den jeweiligen Audio-DMA.

So, jetzt wissen wir also, welche Register wir beispielsweise für den Audio-DMA 0 beschreiben müssen, um einen Ton auf diesem Kanal auszugeben.

Chip-RAM oder Fast-RAM, das ist hier die Frage

Nehmen wir also als Beispiel den Seka-Source-Code 1.

Ziel des Programms ist es, einen kurz anhaltenden Ton auszugeben. Falls Sie mehr

Bin	Hex	Bin	Hex	Bin	Hex	Bin	Hex
0000 = 0		0100 = 4		1000 = 8		1100 = C	
0001 = 1		0101 = 5		1001 = 9		1101 = D	
0010 = 2		0110 = 6		1010 = A		1110 = E	
0011 = 3		0111 = 7		1011 = B		1111 = F	

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Gesetzt = 1																
Ungesetzt = 0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Hex-Wert	8				0				0				1			

Tabelle 1: Berechnung der Hex-Werte

als 512 kByte Speicherplatz in Ihrem Amiga haben, also auch Fast-RAM, dann schalten Sie es bitte vorher mit NO-FASTMEM ab. Dieser Befehl befindet sich normalerweise im System-Ordner der Workbench-Diskette. Warum das Fast-RAM abschalten? Nun, die DMAs können leider nur digitale Daten verarbeiten, die im Chip-RAM liegen. Bei neueren Seka-Versionen können Sie übrigens auswählen, ob Sie mit Chip- bzw. Fast-RAM arbeiten wollen.

Ausgabe eines Tones... Jetzt geht's zur Sache

Als erstes brauchen wir die Startadresse der digitalen Daten, die der DMA ausgeben soll. Diese Startadresse schreiben wir mit

```
move.l #start,$dff0a0
```

in das AUD?LC-Register des Audio-DMA 0, also in die Adresse \$DFF0A0. Als nächstes muß der DMA wissen, wie lang denn die Daten sind, die er abspielen soll. Daher benötigt er die Anzahl der digitalen Daten, die durch 2 geteilt werden muß, denn der DMA will die Länge in Worten! Diesen Wert erhalten wir, wenn wir das Label START vom Label END abziehen und durch 2 teilen. Also:

```
move.w #endstart/2,$dff0a4
```

Als nächstes folgt die Lautstärke, die ins AUD?VOL-Register geschrieben wird. Die Lautstärke darf einen Wert zwischen 0 und 64 annehmen.

0 ist die geringste und mit 64 wird die Lautstärke voll wiedergegeben. Drehen Sie also beim Wert von 64 Ihren Lautstärkeregler am Bildschirm nicht gerade bis zum Anschlag auf. Der Befehl, um die Lautstärke zu setzen, lautet:

```
move.w #64,$dff0a8
```

Das wichtigste für das Abspielen von digitalen Daten ist die sogenannte Sample-Period, also die Abspielgeschwindigkeit. Der Wert 124 darf dabei nicht unterschritten werden, sonst bringen Sie den DMA total zum Ausrasten. Für Noten gibt es feste SamplePeriods. Eine Liste darüber gibt es im Anschluß an den Seka-Source-Code 1. Prinzipiell kann man aber sagen:

Je niedriger die Sample-Period, desto schneller werden die digitalen Daten ausgegeben und desto höher ist dann auch der Ton.

In unserem Beispiel wollen wir einfach einen recht hohen Ton ausgeben. Der Befehl dazu lautet:

```
move.w #150,$dff0a6
```

So, das wären eigentlich schon alle Befehle, um den DMAs die Daten zukommen zu lassen. Fehlt nun also nur noch, daß wir den DMA starten. In unserem Beispiel benutzen wir den Audio-DMA 0. Wir müssen also im DMA-CON-Register das Bit 0 setzen. Um sicherzugehen, daß der DMA auch wirklich gestartet wird, müssen wir auch das Bit 9 für den Gesamt-DMA setzen. Bit 15 setzen wir, damit die angegebenen Bits im DMA-CON-Register gesetzt werden. Hier die Befehlszeile:

```
move.w #?$8201,$dff096
```

Damit wird also der Ton produziert. Die Daten, die ab dem Label START stehen, werden über den Audio-DMA 0 ausgegeben. Nun soll der Ton

aber nicht gleich wieder ausgeschaltet, sondern für ein paar Sekunden gehalten werden. Damit dies geschieht, brauchen wir noch eine Warteschleife, im Beispiel "Loop" genannt.

```
move.l #400000,d0
```

Als erstes brauchen wir einen Startwert, von dem bis 0 heruntergezählt wird. Nehmen wir einfach den Wert 400000. Dieser Wert sieht zwar auf den ersten Blick mächtig gewaltig aus, ist er aber in Wirklichkeit gar nicht. Dieser Wert in der Schleife wird innerhalb von zirka 2 Sekunden auf 0 heruntergezählt.

So, nun kommt als erstes das Label "Loop". Damit das Programm weiß, wohin es springen soll, falls d0 noch nicht 0 ist.

```
loop:
```

Es folgt der Befehl, der d0 um 1 dezimiert.

```
sub.l #1,d0
```

Das soll die Schleife machen, bis d0 = 0 ist. Wir überprüfen dies wie folgt:

```
test.l d0
```

Ja, und wenn d0 nicht 0 ist, also ungleich (NOT EQUAL, = NE), dann soll er zum Label loop verzweigen (verzweigen = branch), also

```
bne loop
```

Wenn dann der Wert 0 erreicht wird, dann soll der Audio-DMA 0 wieder ausgeschaltet werden. Wir löschen also Bit 0 im DMA-CON-Register. Bit 15 muß gelöscht bleiben, damit das angegebene Bit im DMA-CON-Register gelöscht wird. Hier der dazugehörige Befehl:

```
move.w #?$0001,$dff096
```

Damit wäre der Programmcode soweit fertig, es folgt dann ein Rücksprung zum Seka-Assembler mit:

```
rts
```

Fehlen also nur noch die digitalen Daten. Zuerst das Label,

	Audio-DMA0	Audio-DMA1	Audio-DMA2	Audio-DMA3
AUD?LC	\$DFF0A0	\$DFF0B0	\$DFF0C0	\$DFF0D0
AUD?LEN	\$DFF0A4	\$DFF0B4	\$DFF0C4	\$DFF0D4
AUD?VOL	\$DFF0A8	\$DFF0B8	\$DFF0C8	\$DFF0D8
AUD?PER	\$DFF0A6	\$DFF0B6	\$DFF0C6	\$DFF0D6

Tabelle 2: Speicheradressen für den jeweiligen Audio-DMA

Note	SamplePeriod
c	427
c #	404
d	381
d #	359
e	339
f	320
f #	302
g	285
g #	269
a	254
a #	240
h	226
c	214

Tabelle 3. Noten und das dazugehörige SamplePeriod

das den Start der Daten markiert
start:

Nun folgen die digitalen Daten

dc.b -120,-100,-80,-60,
-40,-20,0,20,40,60,
80,100,120,100

dc.b 80,60,40,20,0,-20,
-40,-60,-80,-100

und nun das Label, das für das Ende der Daten steht
end:

In Listing 1 ist der gesamte Seka-Source-Code mit den Kommentaren zusammengefaßt.

Tippen Sie dieses Programm am besten einmal in den Seka-Assembler ein. Wenn Sie es starten, sollten Sie einen Beeeep-Ton hören. Wenn nicht, dann haben Sie etwas falsch gemacht. Wurde beim Abtippen kein Fehler gemacht, dann kann es nur daran liegen, daß der Seka-Programm-Speicher in das Fast-RAM gelegt wurde. Sie können diesen "Fehler" beheben, indem Sie, wie schon beschrieben, das Fast-RAM ausschalten oder bei einer neueren Seka-Version das Chip-RAM als Arbeitsspeicher angeben.

Noten und dazugehörige SamplePeriods.

Was steht wofür?

Tabelle 3 zeigt die einzelnen Noten und die zugehörigen SamplePeriods.

Wenn Sie eine Note davon eine Oktave höher ausgeben wollen, dann brauchen Sie die SamplePeriod nur zu halbieren. Umgekehrt, also wenn Sie die Note eine Oktave tiefer ausgeben wollen, dann müssen Sie die SamplePeriod verdoppeln.

Soweit also das einfache Erstellen eines Tones. Wir können aber aus dem Amiga noch viel mehr herausholen. Zum Beispiel können wir den Lautstärkeverlauf eines Tones beeinflussen. Jedes Instrument hat eine eigene ADSR-Kurve. Was ist eine ADSR-Kurve? Nun, in dieser Kurve ist festgelegt, wie die Lautstärke verläuft. Sie beinhaltet, wie hart der Anschlag ist, wie die Haltezeit ist, wie lange sie gehalten werden soll und der ob Ton schnell abklingen soll.

Nicht nur schön, sonder auch laut: Lautstärke-Manipulation

Schreiben wir nun einfach mal ein kleines Programm, das den Ton aus unserem ersten Beispiel ausgibt. Diesmal wird die Lautstärke des Tones langsam dezimiert. Listing 2 enthält den Seka-Source-Code dazu.

Sie sehen, viel hat sich bei unserem Beispiel nicht verändert. Die Warteschleife ist jetzt in einem Unterprogramm gesetzt worden. Und die Hauptschleife ist nun für die Dezimierung der Lautstärke zuständig.

Wenn Sie das Programm so abgetippt bzw. geändert haben, dann müßten Sie einen Beep-Ton hören, der recht schnell leiser wird.

Es geht auch anders: Lautstärke- und/oder Frequenzmodulation

Es gibt übrigens noch eine weitere Methode, die Lautstärke eines Tones zu beeinflussen: die Lautstärke- und/oder Frequenzmodulation. Hierbei werden jedoch für einen Ton gleich zwei Audio-DMA's benutzt. Daher ist die Methode also nicht gerade sinnvoll für einen Musikeditor. Bei der Lautstärke- und/oder Frequenzmodulation werden gleichzeitig Lautstärke und digitale Daten in die Register eines Audio-DMA's geschrieben. Lautstärke und digitale Daten werden dabei von zwei verschiedenen Audio-DMA's aus dem Speicher gelesen. Wie schon gesagt, diese Methode ist nicht gerade sinnvoll, daher wollen wir sie auch nicht weiter behandeln. Nun wissen wir also, wie man einen Ton ausgibt und wie wir dessen Lautstärke nebenbei

noch verändern. Natürlich können wir die Tonausgabe auch per Interrupt laufen lassen; wie das funktioniert, erklären wir Ihnen ein anderes Mal.

Auch Mozart hat klein angefangen: Die erste Musik...

Zum Abschluß haben wir noch einen dritten Source-Code. Dieses Programm wird Ihnen das einfache Liedbeispiel "Fuchs, Du hast die Gans gestohlen" einstimmig vorspielen. o.k., das ist zwar nichts Besonderes, aber für den Anfang ist es schon ein kleiner Meilenstein.

Wir hätten uns ebensogut den Brandenburgischen Konzerten widmen können, dies hätte aber sicherlich den Rahmen dieses Artikels gesprengt.

Wenn Sie das Programm intensiv durchgehen, werden Sie feststellen, daß es eigentlich recht einfach aufgebaut ist. Sie können auch die Daten editieren und damit ein eigenes Musikstück programmieren.

0	= Pause
1	= Ton c
2	= Ton c #
3	= Ton d
4	= Ton d #
5	= Ton e
6	= Ton f
7	= Ton f #
8	= Ton g
9	= Ton g #
10	= Ton a
11	= Ton a #
12	= Ton h
13	= Ton c
255	= Ende

Tabelle 4. Notenbelegung

ren. Welche Funktion haben die einzelnen Werte? Tabelle 4 macht's deutlich. Mit Hilfe dieser Tabelle können Sie ruhig einmal herumexperimentieren und Ihrer Kreativität freien Lauf lassen.

Beim nächsten Mal wollen wir ein Musikstück per Interrupt ablaufen lassen. Also, viel Spaß beim "Komponieren", bis zum nächsten Mal!

(br)

Listings

```

1: ; Seka-Source-Code 1
2: ;
3: move.l #start,$dff0a0 ; Startadresse
                           nach AUDOLC
4: move.w #end-start/2,$dff0a4 ; Anzahl der
                           Daten nach
                           AUDOLEN
5: move.w #64,$dff0a8 ; Lautstärke
                           nach AUDOVOL
6: move.w #150,$dff0a6 ; SamplePeriod
                           nach AUDOPER
7: move.w #$8201,$dff096 ; Audio-DMA0
                           anschalten
8: move.l #400000,d0 ; Anzahl
                           Schleifen-
                           Durchläufe
9: loop: ; Schleife
10: sub.l #1,d0 ; D0 -1
11: tst.l d0 ; Ist D0 = 0 ?
12: bne loop ; Wenn nein,
                           dann verzweige
13: move.w #$0001,$dff096 ; Audio-DMA0
                           ausschalten
14: rts ; Fertig
15: start: ; Daten, Start-
                           markierung
16: dc.b -120,-100,-80,-60,-40,-20,0,20,40,60,
        80,100,120,100
17: dc.b 80,60,40,20,0,-20,-40,-60,-80,-100
18: end: ; Endemarkierung

```

Seka-Source-Code 1

```

1: ; Seka-Source-Code 2
2: ;
3: move.l #start,$dff0a0 ; Startadresse
                           nach AUDOLC
4: move.w #end-start/2,$dff0a4 ; Anzahl der
                           Daten nach
                           AUDOLEN
5: move.w #64,$dff0a8 ; Lautstärke
                           nach AUDOVOL
6: move.w #150,$dff0a6 ; SamplePeriod
                           nach AUDOPER
7: move.w #$8201,$dff096 ; Audio-DMA0
                           anschalten

```



```

8:      move.w    #64,d0          ; Start-
                                   ; Lautstärke
9: loop: jsr      pause          ; Schleife
10:                                     ; Unterprogramm
                                   ; "Pause" aufrufen
11:      sub.w     #1,d0          ; D0 -1 (= Laut-
                                   ; stärke -1)
12:      move.w    d0,$dff0a0     ; Lautstärke
                                   ; setzen
13:      tst.w     d0             ; Ist D0 = 0 ?
14:      bne      loop           ; Wenn nein,
                                   ; dann verzweige
15:      move.w    #$0001,$dff096 ; Audio-DMA 0
                                   ; ausschalten
16:      rts                                     ; Fertig
17: pause:                                     ; Unterprogramm
                                   ; "Pause"
18:      move.l     #500,d1       ; Anzahl der
                                   ; Schleifen-
                                   ; durchläufe
19: pause_loop:                                     ; Schleife
20:      sub.l     #1,d1          ; D1 -1
21:      tst.l     d1             ; Ist D1 = 0 ?
22:      bne      pause_loop     ; Wenn nein,
                                   ; dann verzweige
23:      rts                                     ; Fertig
24: start:                                     ; Daten, Start-
                                   ; markierung
25:      dc.b      -120,-100,-80,-60,-40,-20,0,20,40,60,
                                   ; 80,100,120,100
26:      dc.b      80,60,40,20,0,-20,-40,-60,-80,-100
27: end:                                     ; Endemarkierung

```

Seka-Source-Code 2

```

1:      ; Seka-Source-Code 3
2:      ;
3:      lea      musik,a0        ; Startadresse
                                   ; der Noten nach
                                   ; A0
4: spiel_loop:                                     ; Notenschleife
5:      move.b    (a0),d0        ; Note nach D0
6:      and.l     #$ff,d0        ; D0 = Nur Byte
7:      cmp.b     #0,(a0)        ; Notenwert = 0?
8:      beq      spiel_pause     ; Wenn ja, dann
                                   ; mach 'ne Pause
9:      jsr      spiel_sound     ; Rufe
                                   ; "Spiel_Sound"
                                   ; auf
10:      jmp      spiel_weiter    ; Verzweige zum
                                   ; Weiterspielen
11: spiel_pause:                                     ; Pause machen
12:      move.w    #64,d0         ; Fast genau wie
                                   ; das Unterpro-
                                   ; gramm Pause,
                                   ; nur das die
                                   ; Lautstärke
13: spiel_pause_loop:                                     ; nicht gesetzt
                                   ; wird.
14:      jsr      pause          ;
15:      sub.w     #1,d0          ;
16:      tst.w     d0             ;
17:      bne      spiel_pause_loop ;
18: spiel_weiter:                                     ; Weiterspielen
19:      add.l     #1,a0          ; Zeiger auf
                                   ; nächste Note

```

```

20:      cmp.b     #255,(a0)      ; Ende der
                                   ; Musik?
21:      bne      spiel_loop     ; Wenn nein, zur
                                   ; Notenschleife
22:      rts                                     ; Fertig, Musik
                                   ; abgespielt.
23: spiel_sound:                                     ; Ton ausgeben
24:      lea      periods,a1     ; Zeiger,
                                   ; Periods-Liste
                                   ; nach A1
25:      asl.l     #1,d0          ; Note * 2,
                                   ; Period = Word-
                                   ; breite
26:      move.w    (a1,d0),d0     ; Periodwert aus
                                   ; Liste nach D0
27:      move.l     #start,$dff0a0 ; Startadresse
                                   ; nach AUDOLC
28:      move.w    #end-start/2,$dff0a4 ; Anzahl der
                                   ; Daten nach
                                   ; AUDOLEN
29:      move.w    #64,$dff0a8    ; Lautstärke
                                   ; nach AUDOVOL
30:      move.w    d0,$dff0a6     ; Period nach
                                   ; AUDOPER
31:      move.w    #$8201,$dff096 ; Audio-DMA 0
                                   ; anschalten
32:      move.w    #64,d0         ; Start-Laut-
                                   ; stärke
33: loop:                                     ; Schleife
34:      jsr      pause          ; Unterprogramm
                                   ; "Pause" auf-
                                   ; rufen
35:      sub.w     #1,d0          ; D0 -1 (= Laut-
                                   ; stärke -1)
36:      move.w    d0,$dff0a8     ; Lautstärke
                                   ; setzen
37:      tst.w     d0             ; Ist D0 = 0?
38:      bne      loop           ; Wenn nein,
                                   ; dann verzweige
39:      move.w    #$0001,$dff096 ; Audio-DMA 0
                                   ; ausschalten
40:      rts                                     ; Fertig
41: pause:                                     ; Unterprogramm
                                   ; "Pause"
42:      move.l     #500,d1       ; Anzahl der
                                   ; Schleifen-
                                   ; durchläufe
43: pause_loop:                                     ; Schleife
44:      sub.l     #1,d1          ; D1 -1
45:      tst.l     d1             ; Ist D1 = 0?
46:      bne      pause_loop     ; Wenn nein,
                                   ; dann verzweige
47:      rts                                     ; Fertig
48: start:                                     ; Daten, Start-
                                   ; markierung
49:      dc.b      -120,-100,-80,-60,-40,-20,0,20,40,60,
                                   ; 80,100,120,100
50:      dc.b      80,60,40,20,0,-20,-40,-60,-80,-100
51: end:                                     ; Endemarkierung
52: periods:                                     ; SamplePeriod-
                                   ; Liste
53:      dc.w      0,427,404,381,359,339,320,302,285,269,
                                   ; 254,240,226,214
54: even                                     ;
55: musik:                                     ; Noten
56:      dc.b      1,3,5,6,8,8,8,8,10,6,13,10,8,0,0,0,10,
                                   ; 6,13,10,8,0,0,0
57:      dc.b      8,6,6,6,6,5,5,5,5,3,5,3,1,5,8,0
58:      dc.b      8,6,6,6,6,5,5,5,5,3,5,3,1
59:      dc.b      255                                     ; Ende-
                                   ; Markierung

```

Seka-Source-Code 3

Eine Bitte an unsere Abonnenten

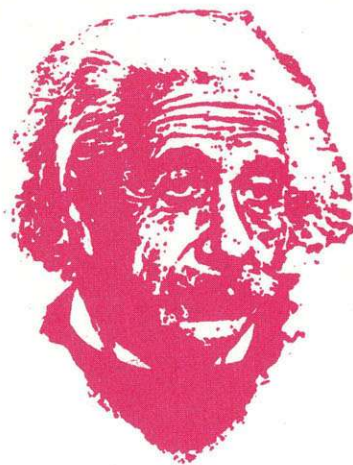
Vermerken Sie bei Schriftverkehr und Zahlungen neben der vollständigen Anschrift stets Ihre Abo-Nummer.

Sie vermeiden damit unnötige Verzögerungen bei der Bearbeitung Ihres Abonnements.

Vielen Dank
Ihre DMV-Versandabteilung

Tüftelecke

Auf dieser Seite wollen wir die kleinen grauen Zellen all jener anregen, die sich zutrauen, kleinere Probleme mittels Computer in den Griff zu bekommen. Damit die geistigen Anstrengungen der sich an unserem Wettbewerb beteiligenden Denker nicht unbelohnt bleiben, gibt es natürlich Preise zu gewinnen, und um diese dem für die Lösung notwendigen Niveau der Grübeleien anzupassen, haben wir "Ausgewählte Texte" unseres Jahrhundert-Genies, Albert Einstein, ausgesucht.



Doch bevor die Schmökereien in den schriftlichen Ausführungen des guten alten Alberts beginnen kann, ist Arbeit angesagt. Wir wollen uns diesmal mit einer der vielen Kreuzworträtselvarianten beschäftigen, dem sogenannten Kreuzwortgitter. Es handelt sich hierbei mehr um eine Wortsuche denn um ein Rätsel. Vorgegeben sind bei diesem Spielchen ein mit scheinbar willkürlich ausgesuchten Buchstaben gefülltes Gitter sowie eine Liste von Worten. All diese Worte sind nun innerhalb des Gitters versteckt. Es existiert dabei keine Vorschrift, in welcher Lage sich die Worte befinden müssen.

Im Gegensatz zu einem normalen Kreuzworträtsel, bei dem Begriffe nur von links nach rechts bzw. oben nach unten eingetragen werden, ist es in dieser Art von Suchspiel denkbar, ja sogar wünschenswert, daß die Worte diagonal angeordnet sind. Das heißt, sie verlaufen von unten nach oben oder müssen von rechts nach links gelesen werden. Der Rater hat nun alle gelisteten Worte innerhalb des Buchstabengewirrs zu finden und diese zu markieren. In unserem Buchstabengewirr wurden zehn Begriffe versteckt. Sie lauten: - Adam - Amiga - DMV-Verlag - Gage - Garage - Nahrung - Ruhe - Suppenhuhn - Sig-

ma - Uhu. Im Normalfall würde der Rater nun Begriff für Begriff aus dem Gitter mühsam herausuchen und beispielsweise mit einer durchgezogenen Linie markieren. Abbildung 1 stellt eine solche Lösungsmöglichkeit dar.

Die programmierte Lösung

Wir wollen dem Computer nun die Arbeit überlassen. Die Erstellung eines solchen Programmes soll Aufgabe dieses Wettbewerbs sein. Was soll das Programm im einzelnen tun?

Wir wollen uns auf ein regelmäßiges Gitter einigen, das heißt, ein Gitter, das entweder quadratisch oder allenfalls rechteckig ist und keinerlei eingesetzte Blindflächen oder Bilder besitzt.

Das Programm soll nun die Anzahl der waagerechten und senkrechten Felder erfragen. Danach sind aus einer Datei die Buchstaben, die im Gitter stehen, und die Begriffe, die dort versteckt sind, einzulesen. Notfalls wäre eine Eingabe der Buchstaben und Begriffe per Tastatur denkbar. Nicht in die Wertung kommen allerdings all die Lösungen, die ihre Datenmengen innerhalb des Programms abgelegt haben. Das würde dazu führen, daß das Programm stets geändert werden müßte, will man ein anderes dieser Rätsel lösen. Allgemeingültigkeit ist also Trumpf!

Des weiteren sollen die Begriffe gefunden und in einsichtiger Weise ausgegeben werden. Denkbar bei der Ausgabe ist beispielsweise die Angabe

der Position des ersten Buchstabens eines jeden Wortes und die Richtung, in der es eingetragen wird. Besonders ansehnlich wäre natürlich die Ausgabe eines Gitters, in das die Buchstaben eingetragen und die Begriffe wie in Abbildung 1 per Strich gekennzeichnet werden.

Teilnahmebedingungen

In die Wertung kommen alle Programmeinsendungen, die ein auf dem Amiga lauffähiges Programm auf Datenträger liefern. (Quellcode und kompilierte Fassung sollten auf Datenträger zu finden sein).

Unter den korrekten Einsendungen werden fünf Exemplare des Buches "Ausgewählte Texte" von Albert Einstein, erschienen im Goldmann Taschenbuchverlag, verlost.

Der Rechtsweg ist dabei wie immer ausgeschlossen.

Schicken Sie Ihre Lösung an den:

**DMV-Verlag Redaktion
AMIGA DOS
Stichwort "Einstein"
Postfach 250
3440 Eschwege**

- Der Einsendeschluß ist der **5. Januar 1990.** (Datum des Poststempels) Mitarbeiter des DMV-Verlages und deren Angehörige sind von der Teilnahme ausgeschlossen.

(jf)

N	H	U	H	N	E	P	P	U	S
A	H	H	Z	G	A	D	R	S	V
H	Q	U	A	U	M	G	R	I	M
R	L	M	P	V	A	G	I	G	A
U	G	A	L	R	E	V	V	M	D
N	V	U	A	R	U	C	H	A	A
G	A	G	E	Y	Z	H	F	Q	J
H	E	R	O	R	N	O	E	X	U

Abb.1: Die Worte sind gefunden. So könnte Ihre Lösung als grafische Ausgabe aussehen.

COMPEDO

TRANSFER FARBBÄNDER

Mit der neuen **COMPEDO-Transferfarbbandkassette**, die passend für jeden gängigen Druckertyp angeboten wird, ist es Ihnen erstmals möglich, Druckvorlagen auf Textilien **aufzubügeln**. Die mit dem Computer erstellte, digitalisierte oder gescannte Vorlage wird auf **Normalpapier** ausgedruckt und dann auf handelsübliche Textilien Ihrer Wahl durch einfaches aufbügeln übertragen. Die Qualität ist so gut, daß sogar kleinste Schriften deutlich zu lesen sind. Alle so erstellten Textilien sind wasch- und reinigungsbeständig. Die Möglichkeiten dieser Anwendung sind nahezu unbegrenzt; so könnten Sie z.B. Ihre Lieblingsgrafik oder auch Ihr Firmenlogo auf T-Shirts, Jacken, Regenschirme, Dekostoffe, Tischdecken etc. übertragen. Zum Aufbügeln ist grundsätzlich jeder Stoff geeignet, der einen Synthetikanteil (Polyester) enthält. Die Ergiebigkeit des Trans-

**IHR COMPUTERAUSDRUCK
ZUM AUFBÜGELN
AUF TEXTILIEN MIT
UNSEREM SPEZIALFARBAND**

ferfarbbandes unterscheidet sich nicht von der eines „normalen“ Farbbandes. Erhältlich sind COMPEDO-Transferfarbbänder für alle gängigen Drucker in den Farben Schwarz, Rot, Gelb und Blau. Für Color drucker (z.B. STAR-LC 10, NEC CP6/CP6+ etc.) wird ein mehrfarbiges Transferfarbband angeboten. Für weitere Informationen, Preise oder Bestellungen steht Ihnen Ihr Fachhändler gerne zur Verfügung.

6-89

**HAPPY
COMPUTER**

Die Modedrucker



Mit dem Bügeleisen kommt der Ausdruck aufs T-Shirt



Farbbänder für Textildrucke gibt es für fast alle Drucker

UND SO URTEILT DIE FACHPRESSE...

Ein heißes Eisen!

Man kann schon eine Menge hübscher Dinge mit einem Drucker und etwas Phantasie machen: Grußpostkarten, Plakate, Türschilder und vieles anderes mehr. Was wir Ihnen hier vorstellen, ist jedoch neu: Computerausdrucke, die mit Spezialfarbbändern der Firma Compedo erstellt wurden, lassen sich direkt auf Stoff aufbügeln! Schauen Sie bitte auf die Titelseite: dies ist nicht die April-Ausgabe...

„Sie brauchen ein Grafikprogramm, mit dem Sie Zeichnungen oder Grafiken erstellen können, einen Drucker, mit dem Sie diese zu Papier bringen können und für den ein Spezialfarbband erhältlich ist (laut Hersteller sind für 90 Prozent der auf dem Markt befindlichen Druckertypen Farbbänder zu haben), ein Bügeleisen und das zu verschönernde Kleidungsstück.“

– Sind die aufgebügelten Bilder waschecht?

Unserer Erfahrung nach: Ja. Ein mit einem farbigen Transfer-Farbband bedrucktes T-Shirt wurde erstmals lauwarm per Hand, dann mehrmals mit 30 Grad in der Maschine gewaschen, ohne daß der Aufdruck an Qualität litt bzw. andere Wäsche verfärbt wurde. **



Grafikausgabe auf Textilien

Was auf den ersten Blick sehr aufwendig und teuer klingt, hat in Wirklichkeit einen wädhlich preiswerten Hintergrund: Die Firma Compedo bietet die neuartigen Thermo-Transfer-Farbbänder an, mit deren Hilfe das persönliche T-Shirt für etwa 30-40 Mark möglich wird. Das Farbband wird in den Drucker eingelegt und das gewünschte Motiv spiegelsymmetrisch auf ein Blatt Papier gedruckt. Danach wird der Ausdruck auf den Stoff gelegt und mit einem Bügeleisen aufgebügelt.

Das Resultat ist ein wasch- und reinigungsfester Ausdruck mit gutem Kontrast. Für den Stoff selbst gibt es kaum Einschränkungen, er muß lediglich einen Synthetik-Anteil aufweisen. Noch interessanter ist das mehrfarbige Transfer-Band, das bei entsprechend farbtauglichen Druckern wie dem NEC-CP6 vierfarbige Motive erlaubt. Dem eigenen T-Shirt oder Sweatshirt mit Klub- oder Firmenlogo steht so nichts mehr im Wege.

1-89
DOS
INTERNATIONAL

64'er 1-89 Der C64 als Modedesigner

Manchmal gibt es Ideen, die sind so einfach wie sie gut sind.

Zum Beispiel das Bedrucken von T-Shirts mit beliebigen Mustern und Grafiken. Keine Angst, Sie müssen jetzt nicht Ihr Lieblings-T-Shirt in den Drucker zwingen – es geht einfacher.

PERNICK AND - MODERN Computerzubehör

CITIZEN 120 D	34,90	EPSON LX 80/90	31,90
EPSON FX/RX 80	33,50	EPSON LQ 500/800	35,90
FUJITSU DX/DL	38,90	NEC P2/P6 MPS 2000	37,50
NEC P3/P7/MPS 2010	40,90	NEC P6+/P7+	39,90
PRASIDENT 63xx	29,90	STAR SG-10	28,50
STAR NL/NG-10	35,90	STAR LC 24-10	36,80
STAR LC-10	33,90	NEC CP 6 4-COLOR	59,90
TALLY 81/MPS 802	36,90	STAR LC-10 C 4-COLOR	46,90
SEIKOSHA SP	35,90	OKI ML 292 4-COLOR	59,90
NEC P 2200	37,90	PANASONIC KXP 10xx	36,90
COMM. MPS 1500	39,90	OKI ML 390	36,70

Alle Farbbänder (auch zum aufbügeln) in Schwarz, Rot, Gelb oder Blau erhältlich
HÄNDLERANFRAGEN ERWÜNSCHT

Normale Farbbänder auch SUPER-preiswert! z.B.:			
STAR LC-10	9,50	STAR LC 10 COLOR	17,90
NEC P2/P6	11,90	NEC P 2/P 6 COLOR	29,90
EPSON FX/RX 80	10,20	NEC P 6+/P 7+ COLOR	29,90
NEC P6+/P7+	14,40	OKI ML 292 4-COLOR	31,50

Weitere Preise auf Anfrage

POSTFACH 1352 POSTFACH 100105
5860 ISERLOHN 4630 BOCHUM
TEL.: 02371/29785 TEL.: 0234/12664
Fax: 02371/24099

VERSANDPAUSCHALE DM 6,-
NACHNAHME O. VORKASSE (AUSLAND)



dem Aufbügeln ist die Farbe fixiert und geht auch bei normalem Waschen nicht mehr heraus. Einem dauerhaften Spaß an Ihren eigenen Motiven ist damit keine Grenze gesetzt. Richtig aufregend wird die Sache dann, wenn mit farbigen Bändern gearbeitet wird. Damit ist es sogar möglich, aus schlichter Kaufhaus-Ausverkaufsauffallende, einzigartige Modeartikel zu machen. Oder denken Sie auch an Hemden, Jacken und Hosen, die man mit netten Bildern zu ganz individuellen Exemplaren machen kann. Schön ist dabei, daß die Farbbänder sehr ergiebig sind. So ist es möglich, mit einem Farbband bis zu 20 gute Drucke herzustellen. **

512KB RAM 1.8/2.0 MB RAM intern

Card für Amiga 500

A502 - abschaltbar

- Megabit-Technologie
- leicht einzustecken
- autokonfigurierend
- Schnelle RAM's
- Uhr nachrüstbar

**SOFORT
LIEFERBAR**

178,-

A502 RAM-Karte mit
akkugepufferter
Echtzeituhr

198,-

2 MB RAM

Board für Amiga 2000

- aufrüstbar bis 8 MByte
- autoconfig, abschaltbar
- null Waitstates, Megabit-Technologie

Wahnsinn!!
- Einführungspreis
Mit 4,68 MByte
auf Anfrage

948,-

Versand per Nachnahme + DM 10,-
Händleranfragen erwünscht

für Ihren Amiga 500 **A580**

- jederzeit bis auf 1.8MB nachrüstbar
- Lieferbar mit 512KB, 1MB & 1.8MB bestückt
- Speichergröße mit DIP-Schaltern einstellbar
- abschaltbar, autokonfigurierend, incl. Uhr & Akku
- wird in den Erweiterungsschacht gesteckt
- modernste Megabit- Technologie, schnelle RAMs
- Neu! A580Plus: mit FatAgnus 8372A
1MB ChipRAM und 2.5MB GesamtRAM

mit 512KB **328,-** mit 1MB **478,-** mit 1.8MB **698,-**

RAM-Erweiterung von 512k auf 1 MByte für

A1002 Amiga 1000

- soft- & hardwaremäßig abschaltbar
- läuft mit allen Erweiterungen - intern
(z.B. Sidecar, Festplatten) auf Wunsch mit Einbau

398,-

3-STATE ..die
Megabyte-Profis!

Tel: 02361/492928
02361/ 179 79
Fax: 02361/43952
Schaumburgstr. 17
4350 Recklinghausen



Edgar Meyz

Einbinden von Maschinencode in M2Amiga-Programme

Es kann sich durchaus die Notwendigkeit ergeben, Maschinencode in Modulprogramme einzubinden. Solange es sich nur um einfache Anweisungen handelt, läßt sich die Aufgabe leicht mit den vorhandenen maschinennahen Sprachelementen REG(), SETREG() und INLINE() alleine bewältigen. Was aber, wenn Parameter zu übergeben sind? Im folgenden wird aufgezeigt, wie der Compiler des M2Amiga mit Parametern bei Prozeduraufrufen umgeht. Es gibt zwei Techniken, Parameter zu übergeben, über den Stack und in Registern:

Parameter über den Stack

Die Parameterübergabe mit Hilfe des Stack entspricht dem Vorgehen in anderen Hochsprachen. Abweichend von "C" werden die Parameter jedoch in der Reihenfolge ihrer Benennung im Prozedurkopf (von links nach rechts) auf dem Stack abgelegt. Über das Register A5, dem lokalen "stack frame pointer" (ein frame ist in diesem Zusammenhang ein Bereich des Stack, der für lokale Objekte reserviert ist), erhält man den Zugriff auf die Parameter und auf die lokalen Variablen. Beim Aufruf einer Prozedur legt das aufrufende Programm die Parameter auf den Stack und sichert dort auch die Rücksprungadresse. Es folgt der Sprung zur Prozedur. Bei Eintritt in die Prozedur wird zunächst A4 auf den Stack gerettet und mit der Basisadresse der globalen Variablen des aktuellen Moduls versehen. Danach wird A5 auf den Stack gegeben und mit dem aktuellen "top of stack" geladen. Die Parameter selbst befinden sich ja schon auf dem Stack. Anschließend wird der Stack Pointer dekrementiert, um Platz für die lokalen Variablen zu reservieren. Ein Beispiel soll das Verständnis erleichtern: angenommen sei die folgende Deklaration einer Prozedur: (Abbildung 1) Der Stack sieht bei Eintritt der Prozedur in den eigentlichen Code wie folgt aus: (Abbildung 2) Die Darstellung soll vermitteln,

daß A5 auf die höchste Adresse des lokalen Parameterbereichs (frame) zeigt. Mit positiven Offsets sind die übergebenen Parameter erreichbar und mit negativen die lokalen Variablen. Der letzte Parameter aus der Parameterliste (im Beispiel P3) liegt stets am Offset "frame - 12". Die erste lokale Variable (im Beispiel L1) hat einen negativen Offset entsprechend der Breite ihres Typs. Nach Abarbeitung des Codes sorgt die Prozedur dafür, daß der Stack abgebaut wird und A4 sowie A5 wieder die Werte enthalten, wie bei Prozeduraufruf (Abweichung zu "C").

Ergebnisübergabe

Funktionsprozeduren liefern ihr Ergebnis in D0 und bei Bedarf zusätzlich in D1, selbst wenn das Ergebnis eine Adresse ist (Ausnahmen bestehen lediglich für die Janus Library). Prozeduren können Ergebnisse in Variablenparametern zur Verfügung stellen. Bei Anwendung dieser Technik werden die Ergebnisse direkt in die Speicherstellen geschrieben, deren Adressen über die Variablenparameter bekannt sind.

Registerparameter

M2Amiga bietet auch die Möglichkeit, Parameter direkt in Registern zu übergeben, also ohne Umweg über den Stack. Ein Prozedurkopf könnte z.B. wie folgt aussehen: (Abbildung 3) Beim Prozeduraufruf werden die Register entsprechend der Parameterliste geladen. Es ist unmittelbar einsichtig, daß die Register a4, a5 und a7 für die-

sen Zweck ungeeignet sind. Durch Verwendung der Compileroption \$E- läßt sich diese Restriktion jedoch zum Teil überwinden.

Zur Implementation von Integerarithmetik

Schauen Sie sich doch einmal den Code an, den Ihr Compiler für DIV/MOD und "*" mit Zweierpotenzen für ganze Zahlen generiert. Der Compiler des M2Amiga arbeitet mit SHIFT-Operationen und dem Ausblenden (ANDI) von Bits. Ein Assemblerprogrammierer könnte es nicht besser machen. Es erübrigt sich daher bei der Programmierung mit M2Amiga, Ganzzahlarithmetik mit Zweierpotenzen besonders zu behandeln. Der Compiler optimiert von sich aus. Bei der Arbeit mit Menüs besteht der Bedarf, die Nummern der Menüs, der Menüpunkte und der Untermenüs zu ermitteln. Dazu kann man wie folgt vorgehen: (Abbildung 4) Häufig wird empfohlen, diesen umständlich erscheinenden Weg nicht zu gehen und die Menüs mit SHIFT-Operationen zu bestimmen.

SHIFT arbeitet mit Langwort

Im Gegensatz zu früher wirkt sich in der aktuellen Version 3.2 des M2Amiga-Compilers SHIFT stets auf ein Langwort aus. Der Compiler unterstützt somit nicht mehr die Möglichkeit, Bits aus der oberen Hälfte eines Wortes durch zweimaliges Schieben (erst nach links und dann nach rechts) zu löschen. Die nach links aus dem Wort herausgeschobenen Bits gelangen beim Schieben nach rechts wieder hinein. Auf Menüs angewandt folgt, daß die Nummer eines Menüitems durch einfache SHIFT-Operationen alleine nicht mehr gewonnen werden kann. Im folgenden Beispiel wird angenommen, daß Intuition als Code 0CA0H liefert: (Abbildung 5) Um die Nummer des Menüitem zu erhalten, sind die Bits 5 bis 10 auszuwerten. Ein fünffacher SHIFT nach links ergibt: (Abbildung 6) Das zu löschende Bit taucht also aus der oberen Hälfte des Langwortes wieder auf. Die korrekte Lösung lautet jedoch: (Abbildung 7) Die Lösung (065H) ist nur mit

mehrmaligem SHIFT zu erreichen und dann noch nicht einmal korrekt. Es wird dazu folgender Code generiert: (Abbildung 8) Erkennen wir die Grenzen der Implementation von SHIFT und begnügen uns mit der eleganten Implementation von DIV und MOD. Angewandt auf unser Beispiel: (Abbildung 9) Nun wird wortweise nach rechts geschoben (DIV) und die oberen 10 Bits werden mit ANDI einfach ausgenipst (MOD). Generierter Code: (Abbildung 10) Thats all!

Variablen auf Langwortadressen

Der Compiler des M2Amiga unterstützt (noch?) nicht die Möglichkeit, globale Variablen auf Langwortadressen zu legen. Bei der Arbeit mit der Dos.Library besteht mitunter die Notwendigkeit dazu. Üblicherweise wird dann Speicherplatz für diese Variablen explizit vom Betriebssystem angefordert. So reservierter Speicherplatz beginnt stets an einer Langwortadresse. Es gibt aber auch einen alternativen Weg, der besonders bei Verwendung des Debuggers von Interesse sein könnte: Deklarieren Sie eine Feldvariable in der Länge der Datenstruktur, die Sie eigentlich verwenden wollen, zusätzlich drei Bytes (*1*). Setzen Sie einen Pointer (vom Typ der benötigten Variable) auf den Anfang der Feldvariable plus drei Bytes. Ermitteln Sie zu der im Pointer enthaltenen Adresse die nächst niedrige Langwortadresse. Setzen Sie den Pointer darauf (*2*). Im praktischen Beispiel verliert der beschriebene Vorgang schnell seine Schrecken: Es soll eine Variable vom Typ Dos.FileInfoBlock angelegt und über einen Pointer angesprochen werden: (Abbildung 11) Die Aufgabe ließe sich auch bei Inkaufnahme eines höheren Aufwands mit einer Multiplikation und einer Division erledigen. SHIFT hat also durchaus seine Berechtigung in der Systemprogrammierung.

Daten für das CHIP-Memory

Bestimmte Daten (z.B. für Audio, Blitter, Copper und Sprites) sind im Chip-Memory ab-

zulegen, damit die Custom-Chips darauf zugreifen können. Der Compiler des M2Amiga bietet noch nicht die Möglichkeit, Daten bei Programmaufstart direkt ins Chip-Memory zu laden. Es sind stets ein paar Programmzeilen zu schreiben, um die Daten vom Fast-Mem in das Chip-Mem zu kopieren. Neben dem üblichen Weg, die Datenwörter abzuzählen und ihre Inhalte in das Chip-Mem zu übertragen, bestehen noch andere Verfahren, die die Aufgabe erleichtern. Lassen wir zunächst Beispiele sprechen:

(Abbildung 12)

Zu den Beispielen ist anzumerken: Die Audiodaten wurden als INLINE-Code abgelegt und nicht wie üblich in einer Feldvariablen. Dadurch erspart man sich das Füllen der Feldvariablen zur Laufzeit. Wichtig dabei ist die Compileroption \$E-, um zu verhindern, daß der Compiler für die "Tabellen-Prozedur" Eintrittscode generiert.

"Verfahren1" beinhaltet die schnellste Art, Datenmengen von weniger als 96 Bytes umzuschaukeln. Die Zuweisung "audioDaten^ := audioFast^;" ist so implementiert, daß wortweise umgespeichert wird. Nachteilig an dem Ver-

fahren ist, daß dazu die genaue Menge der Daten bekannt sein muß, um den Typ "Sound" zu deklarieren. "Verfahren2" zeichnet sich durch seine Flexibilität aus. Es ist nicht erforderlich, die Menge der Daten abzuzählen. Sie wird durch eine einfache Adreßarithmetik bestimmt. Die Systemroutine "CopyMem" speichert gleichfalls wortweise um und arbeitet schneller als "Verfahren1", wenn mehr als 96 Bytes umzuschaukeln sind.

"Verfahren3" stellt bezüglich der Flexibilität eine Alternative zu "Verfahren2" dar, wenn die Datenmenge nicht dyna-

misch ermittelbar ist. Der Bedarf an Chip-Mem ist dann auf die höchstmögliche Datenmenge auszurichten. Weiterhin muß das letzte Wort der zu übertragenden Daten exakt bekannt sein, um die Schleife zu verlassen. Abschließend bleibt noch anzumerken, daß die gewählte Implementierung einer Tabelle nicht statisch ist. Der Inhalt der Tabelle kann mit Hilfe eines Pointers durchaus verändert werden. Dabei genießt man sogar einen Bereichsschutz, wenn der Zeiger beispielsweise als "POINTER TO Sound" deklariert ist.

(mm)

<pre>(ABB.1) PROCEDURE DoIt (P1,P2:ADDRESS;P3:INTEGER); VAR L1,L2 : INTEGER; L3 : LONGREAL;</pre>			
<pre>(ABB.2)</pre>			
Adressen	Param./Reg.	Stack	Anmerkung
/\	P1	frame + 18	belegt Bytes 18 - 21
	P2	frame + 14	4 Bytes (ADDRESS)
	P3	frame + 12	2 Bytes (INTEGER)
	Rücksprung.	frame + 8	für RTS
	a4 alt	frame + 4	fester Offset
	a5 alt	frame	beginnt hier
	L1	frame - 2	2 Bytes
	L2	frame - 4	2 Bytes
	L3	frame - 12	8 Bytes (frame Ende)
	a7 -->	frame - 12	ist "top of stack"
<pre>(ABB.3) CONST d0 = 0; a0 = 10C; a1 = 11C; PROCEDURE DoIt (P1{a0},P2{a1}:ADDRESS;P3{d0}:INTEGER);</pre>			
<pre>(ABB.4) menu := code MOD 32; menuItem := Code DIV 32 MOD 64; menuSubItem := Code DIV 2048 MOD 32;</pre>			
<pre>(ABB.5) 0 C A 0 0000 1100 1010 0000 = 1 Wort = 16 Bits └────────┘ MenuItem</pre>			
<pre>(ABB.6) 0001 1001 0100 0000 0000 Ein SHIFT "rechts 10" resultiert in: 0000 0000 0110 0101 = 065H</pre>			
<pre>(ABB.7) 0000 0000 0010 0101 = 025H</pre>			

<pre>(ABB.8) ls1.l #5,d6 lsr.l #2,d6 lsr.l #8,d6 ; achtmal rechts</pre>	
<pre>(ABB.9) subItem := 0CA0H DIV 32 MOD 64;</pre>	
<pre>(ABB.10) lsr.w #5,d6 andi.w #63,d6</pre>	
<pre>(ABB.11) (*1*) VAR fileInfoBuffer : ARRAY[0..SIZE(FileInfoBlock)+ fileInfo : FileInfoBlockPtr; (* importiert *) (*2*) fileInfo:=ADDRESS(SHIFT (SHIFT (ADR (fileInfoBuffer[3]),-2) 2)); WriteString (fileInfo^.fileName); (* eine Anwendung *)</pre>	
<pre>(ABB.12) TYPE Sound = ARRAY[0..7] OF WORD; VAR audioFast, audioDaten : POINTER TO Sound; (* \$E- folgende Prozedur ist eine Tabelle, beginnend mit 031H *) PROCEDURE AudioDaten; BEGIN INLINE (00031H, 05A75H, 07F75H, 05A31H, 000CFH, 0A686H, 08186H, 0A6CFH); END AudioDaten; PROCEDURE Verfahren1; BEGIN audioDaten := AllocMem (TSIZE (Sound),MemReqSet {chip}); audioFast := ADR (AudioDaten); audioDaten^ := audioFast^; END Verfahren1; PROCEDURE Verfahren2; BEGIN laenge := CAST (LONGINT, (ADR (Verfahren1) - ADR (AudioDaten))); audioDaten := AllocMem (laenge, MemReqSet {chip}); CopyMem (ADR (AudioDaten), audioDaten, laenge); END Verfahren2; VAR datenChip, datenFast : POINTER TO INTEGER; PROCEDURE Verfahren3; BEGIN datenChip := AllocMem (TSIZE (Sound),MemReqSet {chip}); datenFast := ADR (AudioDaten); LOOP datenChip^ := datenFast^; IF datenChip^ = 0A6CFH THEN EXIT END; INC (datenChip, 2); INC (datenFast, 2); END; END Verfahren3;</pre>	

BÜCHER, MIT DENEN SIE



DIE KLEINEN UND GROSSEN HELFER.

Die besten Amiga-Utilities – der Titel verfügt es bereits: Die beliebtesten und stärksten Hilfsprogramme zum Amiga werden detailliert beschrieben und erklärt. Ein kleiner Ausschnitt aus dieser Utility-Hilfste: Diskmaster, Butcher V2, Discovery, der Editor Ed, Quarterback, Aztec C Compiler, Power Window, Creat-A-Shape, Zenon – eben alles, was Rang und Namen hat.

Die besten Amiga-Utilities
ca. 420 Seiten, DM 39,-



Jetzt in überarbeiteter Neuauflage: Das große Floppybuch mit dem Basiswissen für Einsteiger und den Insider-Informationen für Aufsteiger. Von den Floppy-Operationen unter Workbench und CLI/Shell über sequentielle und relative Dateien bis zum Diskettenzugriff ohne DOS: MFM, GCR, Aufbau von Tracks, Block-Header, Datenblock, Prüfsummen, Codierung und Decodierung, Hardware-Register, SYNC und Interrupts. Außerdem wichtige Tips zum Kopier- und Virenschutz. Und starke Programme auf Diskette: z.B. Fast-, Crunch- und DeepCopy sowie ein Diskettenmonitor.

Das große Floppybuch
Hardcover, 551 Seiten
inkl. Diskette, DM 59,-

Das gesamte Know-how rund um Amiga-DOS in einem Buch: Umlenken der Ein- und Ausgabe, sinnvoller Einsatz des Jokers, Arbeiten mit einer RAM-Disk, nützliche Batch-Dateien, Startup-Sequence, Aufbau der CLI-Befehle, neue CLI-Befehle in BASIC und C und und und.

Das große Buch zu AmigaDOS
Hardcover, 406 Seiten
inklusive Diskette, DM 59,-



KNOW-HOW ZU DEN AMIGA-PROGRAMMEN.

Ob TEXTOMAT, BECKERtext, DATAMAT oder Superbase – wir liefern Ihnen das Know-how, um aus diesen Programmen das Beste herauszuholen. In unseren Büchern finden Sie Informationen, die in den Handbüchern nicht stehen können. Beispielsweise die praktischen Anwendungsbeispiele, die Sie ohne weiteres für Ihre eigenen Bedürfnisse nutzen können. Oder auch die nützlichen Tips & Tricks, mit denen Sie die eine oder andere Unzugänglichkeit der Programme geschickt umschiffen können. Natürlich immer mit einer detaillierten Beschreibung des gesamten Leistungsumfanges.



TEXTOMAT & BECKERtext
Know-how
286 Seiten, DM 39,-



DATAMAT
Know-how
442 Seiten, DM 39,-

Das große Buch zu
Superbase
414 Seiten, DM 39,-



DIE GANZE, FARBENFROHE PALETTE.

Neben einer ausführlichen Beschreibung der dPaint-Grundfunktionen zeigt dieses Buch vor allem, was dPaint tatsächlich leisten kann: perspektivische Zeichnungen, verbogene Brushes, Animation in 3D, Erstellen von Videos und und und. Dazu zahlreiche Tips & Tricks und alles über den Datenaustausch mit anderen Systemen.

Das große dPaint-III-Buch
ca. 300 Seiten, DM 39,-



Im großen Viren-Schutzpaket zum Amiga finden Sie Programme, die Computer-Viren sofort erkennen und entfernen. Sei es auf der Festplatte oder auf der Diskette. Auch zukünftige Störenfriede, beispielsweise Link-Viren, werden dabei schon berücksichtigt. Im Buch das entsprechende Hintergrundwissen zu Virenprogrammen (Verbreitung, Aufbau u.v.a.m.).

Das gr. Viren-Schutzpaket – Amiga
172 Seiten, inkl. Diskette, DM 69,-

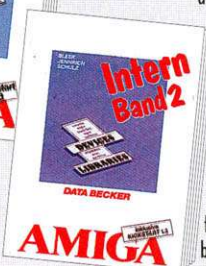
DATA BECKER

WEITERKOMMEN.

DIE KLASSIKER EINES JEDEN AMIGA-PROFIS.

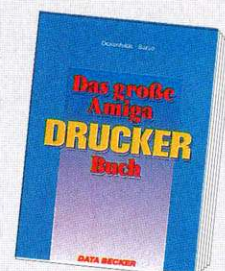


Amiga Intern
Hardcover, 716 S.
DM 69,-



Amiga Intern Band 2
Hardcover, 895 S.
DM 69,-

Für jeden engagierten Anwender, der etwas auf sich hält, gehören die Intern-Bände einfach zur Pflichtlektüre. Woher sollte er auch sonst seine Detailkenntnisse zu den internsten Vorgängen in seinem Rechner nehmen? Zum Amiga gibt es gleich zwei Intern-Bücher – damit Sie bei dieser Fülle an Informationen auch noch den Überblick behalten. So ist der erste Band für den Einsteiger genauso gedacht wie für den Profi. Sie finden hier alles, was es zum internen Aufbau des Amigas zu sagen gibt. Von der Beschreibung der einzelnen Prozessoren bis hin zur Programmierung eigener DOS-Handler. Band 2 ist die Fachliteratur für den aktiven Programmierer, der alle weiterführenden Informationen zu seiner Arbeit schnell und zuverlässig finden will.



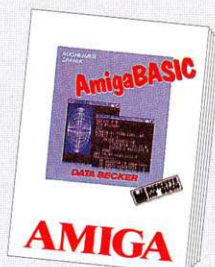
Ärgern Sie sich nicht über fehlende Umlaute oder Papierstaus beim Ausdruck Ihrer Dokumente. Schlagen Sie einfach im großen Amiga-Druckerbuch nach. Hier finden Sie nicht nur Problem-Lösungen im Text, sondern auch auf der beiliegenden Diskette: ein Utility-Programm für eine komfortable Druckersteuerung.

Das große Amiga-Druckerbuch
Hardcover, ca. 300 Seiten
inkl. Diskette, DM 59,-

TOP-PROGRAMME ZU EINEM SPOTTPREIS.

Amiga ToolBox – Software im Buch. Insgesamt über 40 Super-Programme. Für alle Amiga-Freunde. Die stärksten Werkzeuge: der CLI-Manager (Installation von Boot-Blöcken, Icons erstellen...), DISK-Manager (Bootblock-Backup, Editieren von Tracks...), Black-Copy (Kopieren von Amiga-Fast- wie diversen Fremdformaten...). Dazu weitere kleine Programme – zu nahezu allen Anwendungsbereichen. Im Buch selbst finden Sie eine detaillierte Beschreibung der einzelnen Programme sowie das nötige Hintergrundwissen zu SYNC-Markierungen, Track-Lücken...

Amiga ToolBox
Hardcover, 222 Seiten
inkl. Diskette, DM 69,-



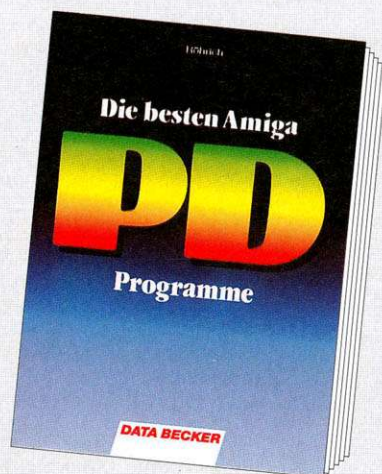
Das international erfolgreiche Buch zum AmigaBASIC. Mit allem, was BASIC-Programmierern Spaß macht: Animation, Grafik-Befehle, Pulldowns, Mausbefehle, Einlesen und Abspeichern von IFF-Bildern, sequentielle und relative Dateien und eine Beschreibung des AC-/BASIC-Compilers.

AmigaBASIC
Hardcover, 792 Seiten
inklusive Diskette, DM 59,-

RUND UM PUBLIC DOMAIN.

Hier sind sie alle aufgelistet und detailliert beschrieben: die besten und aktuellsten Public-Domain-Programme, die zur Zeit für den Amiga erhältlich sind. Von der Installation über den Leistungsumfang bis zum praktischen Einsatz. Mit jeder Menge Tips. Kurz: ein Buch, mit dem Sie den Überblick be- und wichtige Tips erhalten.

Die besten Amiga-PD-Programme
ca. 300 Seiten, DM 39,-



BESTELL COUPON!

Einsenden an:
DATA BECKER
Merowingerstr. 30
4000 Düsseldorf 1

Hiermit bestelle ich für meinen Amiga

Name, Vorname

Straße

PLZ/Ort

Ich zahle
☐ per Nachnahme
☐ mit beiliegendem Verrechnungsscheck

(zzgl. DM 5,- Versandkosten unabhängig von der bestellten Stückzahl)

Robert Haas

C's Vor-Arbeiter

Jeder, der in C programmiert, wird im Handbuch seines Compilers schon mal auf das Wortungetüm C-Preprozessor getroffen sein. Zumeist ist die Erklärung recht kurz und in englisch gehalten, oder sie ist so verklausuliert, daß man sie kaum verstehen kann. Deshalb nun die Erklärung: Was ist ein 'C-Preprozessor' eigentlich, und was kann er alles?

Die Hauptaufgabe eines C-Preprozessors (im folgenden kurz CPP genannt) ist das Entfernen von allem für den Compiler Überflüssigen.

Dazu gehören Kommentare, Tabulatoren und Einrückungen im Programmtext. Er besitzt jedoch auch einige spezielle Anweisungen, die eine größere Zahl von Möglichkeiten für den C-Programmierer eröffnen. Mit dem CPP ist es möglich,

- Textersatz im Quellprogramm vorzunehmen,
- Makros zu schreiben (eine etwas speziellere Form des Textersatzes),
- Dateien einzufügen,
- nur bestimmte Teile eines Programmes zu compilieren (bedingte Compilierung),
- im Falle eines Fehlers eine benutzereigene Meldung auszugeben und
- Compileranweisungen zu übersteuern.

Die beiden letzten Punkte entsprechen dem neuen ANSI-C-Standard (ANSI = American National Standard Institute), der auch noch andere Erweiterungen für die Sprache C mit sich brachte. Falls Sie einen älteren Compiler besitzen, oder der Compiler-Hersteller meint, ihn ginge dieser neue Standard nichts an, so können Sie dennoch ruhig weiterlesen. Es geht hier nicht um diese Neuerungen durch den ANSI-Standard, die beim CPP ohnehin nur marginalen Charakter haben, sondern um die Möglichkeiten des CPP im allgemeinen. In Tabelle 1 werden die einzelnen Befehle des CPP kurz vorgestellt. Wie auf den ersten Blick zu erkennen ist, beginnen alle Kommandos, die den CPP betreffen, mit dem #-Zeichen. Ob die Befehle ab der ersten Spalte beginnen müssen, oder ob Einrückun-

gen erlaubt sind, schlägt man am besten im Handbuch seines Compilers nach.

#define

Die einfachste (und wahrscheinlich auch bekannteste) Form des CPP-Einsatzes ist der einfache Textersatz mit #define. Die Anweisung

```
#define EINS 1
```

veranlaßt den CPP, sämtliche Vorkommen von EINS durch 1 zu ersetzen. Ausgenommen hiervon sind Zeichenketten (z.B. "Hallo EINS").

Fällt dem Programmierer ein, daß EINS in der ersten Hälfte des Programmes 1, danach aber unbedingt den Wert 2 annehmen muß, dann birgt dies auch kein größeres Problem. Man 'undefiniert' EINS ab der Stelle im Text, wo es 2 werden soll mit

```
#undef EINS
```

für den CPP und definiert danach EINS (in unserem Fall als 2) neu.

Wer sich die .H-Dateien seines Compilers angesehen hat, wird festgestellt haben, daß der Compiler selbst auch ausgiebig Gebrauch von den Möglichkeiten des CPP macht. Wichtige Konstanten werden per #define mit aussagekräftigen Namen versehen. Damit werden Programmänderungen erleichtert und Portierungen auf anderen Rechnern vereinfacht.

Aus diesem Beispiel sollte man lernen, selbst Konstanten mit #define an den Programmstart, oder in eigenen .H-Dateien, zu definieren, um sich das Ändern der Programme zu vereinfachen. In Listing 1 werden einige allgemeine Definitionen vorgeschlagen, die Ihnen bei der Arbeit helfen sollen. Wie zu sehen ist, gehen wir nicht gerade sparsam mit Klammern um. Das hilft aber bei komple-

xen Ausdrücken, schwer festzustellende Auswertungsfehler zu vermeiden.

Einfach nur vorhanden sein

Eine Definition der Form #define TEST definiert das Symbol TEST als vorhanden. Diese Form der Definition eignet sich besonders für die bedingte Übersetzung von Programmen. Darunter ist zu verstehen, daß mit Hilfe bestimmter Anweisungen (siehe Tab. 1) nur ein Teil des Programmtextes kompiliert wird. Ein kurzes Beispiel dafür:

```
#ifdef HALLO
printf("Hallo");
#else
printf("Dann eben nicht");
#endif
```

Ist HALLO definiert, wird printf("Hallo") kompiliert. Ist dies nicht der Fall, dann wird der dem #else folgende Programmteil kompiliert. Eine Anwendung für die bedingte Kompilation sind Programmteile, die nur für Testzwecke im Programm enthalten sein sollen. Werden solche Teile in Klammern gestetzt, reicht es aus, eine Definition zu streichen, um sie nicht mitzukompilieren.

Makros

Das vielleicht größte Einsatzgebiet von #define ist die Erzeugung von parametrisiertem Textersatz, kurz Makro genannt. Hier hängt der Ersetztext von dem Aufruf des Makros ab. Als Beispiel für solch ein Makro wird hier MAX (s. Listing 1) genauer erklärt. MAX kann folgendermaßen definiert werden:

```
#define MAX(x,y) \
(((x) < (y))?(y):(x))
```

Zweck dieses Makros ist die Feststellung, welcher der zwei Werte (x oder y) der größere ist.

Es ist wichtig darauf zu achten, daß in der Definition zwischen dem Makronamen und der linken Klammer kein Zwischenraum steht, damit der CPP erkennen kann, daß eine Parameterliste folgt. Unter normalen Umständen endet eine Definition mit dem Zeilenende. Reicht, wie hier, der Platz nicht, oder soll das ganze übersichtlicher sein, fügt man einen \ (Backslash) mit folgendem Zeilenvorschub ein. Man kann die Definition dann in der nächsten Zeile fortsetzen usw..

Kommt jetzt in einem Programm mit obiger Definition von MAX die Zeile

```
i = MAX(g,h);
```

vor, ersetzt der CPP diese durch

```
i = (((g) < (h))?(h):(g));
```

Was zuerst wie ein normaler Funktionsaufruf aussah, hat sich nach dem Textersatz durch den CPP in einen Ausdruck verwandelt (es wird hier mit Hilfe des C-Operators ?: getestet, welcher der zwei Werte der größere ist. Dieser wird dann der Variablen i zugewiesen).

Vor- und Nachteile

Makros haben (wie alles im Leben) Vorteile, Nachteile und einige böse Fallstricke. Zuerst die positiven Seiten:

1. Mit Hilfe von Makros kann man leichter lesbare Programme erstellen.
2. Da keine Funktion (bei MAX und ähnlichen Makros) aufgerufen wird, läuft das Programm schneller.
3. MAX (sowie ähnliche) Makro(s) läßt sich ohne Änderung auf verschiedene Zahlentypen wie double, int, long, etc. anwenden. Dies ist bei C-Funktionen, die nur einen Zahlentyp zurückgeben können, unmöglich.

Dem Vorteil der schnelleren Ausführung steht als Nachteil gegenüber, daß kompilierte Programme bei regem Gebrauch von Makros einiges mehr an Größe zeigen können als erwartet. Makros werden bei jeder Benutzung in dem Programmcode eingefügt. Bei Funktionen reicht es aus, ihren Programmcode nur einmal einzufügen. Wer kurze Programme benötigt, muß das bzw. die entsprechende(n) Makro(s) als Funktion schreiben und mit einem etwas langsameren Programm leben.

Fallgruben

Wer sich über das Klammergebirge bei MAX schon gewundert hat, dem sei gesagt, daß man Makros eigentlich nicht mit genug Klammern versehen kann, um eine der übelsten ihrer Fallen zu entgehen. Als kleines Beispiel hierfür dieses Makro:

```
#define QUAD(x) x*x
```

Auf den ersten Blick fällt manchem vielleicht noch nichts auf. Bei einer Benutzung in der Form:


```
i = QUAD(j);
stimmt auch alles. Nur was
passiert nun?
i = QUAD(j + 1);
```

Na? Der CPP macht daraus:

```
i = j + 1 * j + 1;
```

Was nun nach den Regeln der Mathematik etwas anderes ergibt, als das erwartete Resultat.

Besser definiert, um solch Ungemach zu verhindern, müßte QUAD folgendermaßen aussehen:

```
#define QUAD(x) ((x)*(x))
```

Aus obigem Beispiel würde der CPP folgendes machen:

```
i = ((j + 1)*(j + 1));
```

Was auch endlich zu dem erwarteten Ergebnis führt.

Eine andere böse Fallgrube sind die Seiteneffekte (nicht sichtbare Änderung von Variablen, usw.) mancher Funktionen und Operatoren, besonders bei den - und ++ Operatoren. Ein Beispiel für diese unliebsamen Seiteneffekte sei hier anhand des bereits erklärten MAX-Makro gezeigt. Bei dem Aufruf

```
i = MAX(g + +, h);
```

wird vom CPP folgendes erzeugt:

```
..fi = (((g + +) < (h))?(h):(g + +));
```

Wie man sieht, wird g zweimal um eins erhöht, so daß (wenn g > h war), g + 1 anstatt g der Variablen i zugewiesen wird.

Leider gibt es keine Möglichkeit, Makros dergestalt zu formulieren, daß Effekte dieser Art ausgeschlossen werden können. Deshalb sollte man Funktionsaufrufe, die Seiteneffekte oder die -- und ++ Operatoren nicht in Makroaufrufen verwenden.

Und noch eine Grube...

Es ist natürlich auch möglich, mehrere Anweisungen zu einem Makro zusammenzufassen. Hierbei trifft man aber auch auf eine Falle der Sprache C. Nehmen wir als Beispiel folgendes Makro:

```
#define ETWAS(a) \
    printf("%d",a); \
    a = a + 1
```

Meist wird alles funktionieren, was dann die Fehlersuche jedoch umso schwieriger macht. Ein Beispiel:

```
if(b = 0)
    ETWAS(b);
```

Hier passiert jetzt folgendes: Der CPP macht aus dem obi-

gen Programmfragment

```
if(b = 0)
    printf("%d",b);
    b = b + 1;
```

(sieht nicht genauso aus, zeigt das Problem aber besser). Die Variable b wird also, egal ob es vor der if-Abfrage 0 war oder nicht, um 1 erhöht. Das Makro so zu schreiben

```
#define ETWAS(a) \
    {printf("%d",a); \
    a = a + 1;}
```

bringt zwar bei obigem Beispiel die Lösung, aber wie sieht's hiermit aus:

```
if(b = 0)
    ETWAS(b);
else
    WASANDERES(b);
```

Dieses Programmfragment wird nun erweitert:

```
if(b = 0)
{
    printf("%d",b);
    b = b + 1;
}
else
    WASANDERES(b);
```

Hier wird das else anstatt des if-Blocks dem Semikolon zugeordnet, was syntaktisch auch möglich ist. Der Compiler wird das Ganze aber nun mit einem "else without if" Fehler bemängeln.

Abhilfe kann hier mit dem Komma-Operator geschaffen werden. Er wird hauptsächlich in for-Schleifen verwendet, um mehrere Variablen auf einmal initialisieren zu können. Ausdrücke, die mit seiner Hilfe getrennt werden, werden von links nach rechts bewertet und als eine Anweisung angesehen. Unser ETWAS-Makro sähe nun folgendermaßen aus:

```
#define ETWAS(a) \
    printf("%d",a), \
    a = a + 1
```

Jetzt klappt das Makro auch bei allen angesprochenen Problemen. Trotzdem sollte man mit dem Komma-Operator sehr sparsam umgehen, da er die Lesbarkeit von Programmen nachteilig beeinflussen kann (es gibt nichts Schlimmeres, als nach zwei Tagen vor seinem Programm zu sitzen und keine Zeile zu verstehen).

ANSI-C, Parameter und Makros

Zum Abschluß noch einige Makros für etwas ANSI-Fee-ling bei älteren Compilern. Zur Erklärung dieser Makros muß etwas ausgeholt werden. Während es früher ausreichte, als Deklaration einer Funk-

tion **long funk()**; zu schreiben, um dem Compiler die Funktionswerte mitzuteilen, erlauben die meisten neueren Compiler eine Deklaration der einzelnen Parameter einer Funktion, um Typproblemen auf die Spur zu kommen. Hier ein Beispiel:

long funk(long);

Die Funktion funk, die einen long-Wert zurückgibt, erhält auch einen long-Wert als Parameter. Wenn dieser Parameter einen anderen Typ besitzt, vergessen wird, oder zwei Parameter übergeben werden, meldet dies der Compiler. Der ANSI-Standard erlaubt, den Parametern auch Namen zu geben, um die Fehlermeldungen klarer zu machen. Soweit das Vorgeplänkel.

Es ist ohne weiteres möglich, in C Funktionen zu schreiben, die eine variable Anzahl von Parametern nutzen. Die bekannteste dieser Funktionen dürfte wohl printf sein. Eine Funktionsdeklaration bei neueren Compilern sieht folgendermaßen aus:

```
int printf(char *,);
```

Das Komma gibt dem Compiler an, daß noch weitere Parameter folgen können (aber nicht müssen). In ANSI-C sähe das so aus:

```
int printf(char *fmt,... );
```

Hier sagen die drei Punkte, daß Parameter folgen können. Bei ANSI-C-Compilern gibt es eine .H-Datei mit Namen stdarg.h, welche mehrere Makros enthält, die den Zugriff auf optionale Parameter einer Funktion erleichtern. Die in Listing 2 enthaltenen Makros sollen älteren Compilern erlauben, den gleichen vereinfachten Zugriff auf Parameter zu erreichen.

va_list ist eine Typdefinition für einen Zeiger auf ein char, um an die Parameter zu kommen. Wenn Ihr Compiler den Typ void für Zeiger kennt, setzen Sie diesen anstatt char ein. Man hat dann einen typfreien Zeiger auf beliebige Objekte. Hier sollte man die Dokumentation seines Compilers befragen.

va_start setzt den Parameterzeiger auf den ersten optionalen Parameter. Hierzu muß va_start wissen, welcher Parameter der letzte nicht optionale ist. Bei ANSI-C reicht es aus, einen va_list-Zeiger und den Namen des letzten nicht optionalen Parameters anzugeben. Den Rest erledigt der Compiler selbständig. Bei

- Einfügen von Dateien	
#include "filename"	
#include <filename>	
- Textersetzung	
#define identifier	Ersatztext
- Makrodefinition	
#define identifier(id, ..., id)	Ersatztext
- Löschen einer Definition	
#undef identifier	
- Bedingte Compilierung	
#if konstante	
#ifdef identifier	
#ifndef identifier	
#else	
#elif konstante	
#endif	
Erweiterungen des ANSI-C-Standards	
- Compilersteuerung	
#pragma Compiler-Direktive	
- Benutzereigene Fehlermeldung	
#error Benutzer-Meldung	

Tabelle 1: Alle CCP-Kommandos auf einen Blick

dem Makro in Listing 2 muß aber auch der Typ dieses Parameters angegeben werden.

Das nächste Makro ist `va_arg`. Es übergibt den nächsten dynamischen Parameter in der Parameterliste. Dazu wird der mit `va_start` initialisierte `va_list` Zeiger und der Typ des Parameters an das Makro übergeben, damit der Zeiger entsprechend weiterbewegt werden kann.

Bei diesem Makro ist etwas sehr wichtiges zu beachten. Wenn es sich bei dem zu holenden Parameter um einen Wert vom Typ `char` handelt, muß man ihn als Typ `int` an-

geben. C übergibt Parameter über den Stack. `char`, ist aber normalerweise nur ein Byte groß. Parameter werden jedoch minimal in Wortgröße (2 Byte) an eine Funktion übergeben (Stack).

Bei `char` als Typ wird der Zeiger nur ein Byte weiterbewegt, was beim nächsten Zugriff mit einem Buserror (schlimmster Fall) oder falschen Werten beim nächsten Parameter bestraft wird (auch nicht besser).

Genaueres zur Parameterübergabe, Größe der verschiedenen Typen können Sie im Handbuch ihres Compilers

nachlesen. Als zweite Gefahr ist zu sagen, daß dieses Makro mit Seiteneffekten arbeitet (s.o.). Also sollte man es nicht innerhalb anderer Makros verwenden.

Das letzte Makro mit Namen `va_end` ist nur wegen der Kompatibilität im ANSI-Standard enthalten. Sonst hat es keinen besonderen Zweck. Zum Testen können Sie das Programm von Listing 3 verwenden, das als Beispiel zur Benutzung der `va_*` Makros dienen soll.

(br)

Literatur:

Kernighan/Ritchie
Programmieren in C
Hanser Verlag 1983

Turbo C 2.0 Handbuch
Borland Intl. 1988
Mark Williams Company

ANSI C, A lexical Guide
Prentice Hall 1988

```
*****
* AMIGA DOS INFO
* Programmname : Listing 1
* Funktion : siehe Text
* Sprache : Standard-C
* Besonderheiten : keine
*****
1: /* #defines zur Arbeitserleichterung. Könnten in einer
2: /* Header Datei wie mydefs.h enthalten sein.
3: /* Einige einfache #defines zur Vereinfachung des
4: /* Lesen eines Programms, für Ex-Pascaler, und zur
5: /* Vereinfachung von Portierungen auf andere Rechner */
6:
7: #define FALSE ( 0 )
8: #define TRUE (!FALSE)
9: #define BOOLEAN int
10: #define BYTE unsigned char
11: #define REAL double
12: #define WORD16 unsigned short int
13: #define WORD32 unsigned long int
14: #define INT16 short int
15: #define INT32 long int
16:
17: /* Eine Endlosschleife */
18: #define FOREVER for( ; ; )
19:
20: /* Makros, bei einigen Compilern bereits definiert */
21: #define ABS(x) ((x)<0)?-(x):(x)
22: #define MAX(x, y) ((x)<(y)?(y):(x))
23: #define MIN(x, y) ((x)<(y)?(x):(y))
```

Listing 1: Definitionen

```
*****
* AMIGA DOS INFO
* Programmname : Listing 2
* Funktion : siehe Text
* Sprache : Standard-C
* Besonderheiten : keine
*****
1: /* ANSI-C ähnliche Makros und Definition zum
2: /* Leichterem Zugriff auf optionale Parameter einer
3: /* Funktion. Sollten in eine Header-Datei mit Namen
4: /* stdarg.h gepackt werden.
5: /* Um mehrfaches einlesen zu verhindern.
6:
7: #ifndef __STDARG_H__
8: #define __STDARG_H__
9:
10: /* Definition eines Zeigers für die va_* Makros
11: typedef char *va_list;
12:
13: /* Makro mit dem ein va_list Zeiger auf den ersten
14: /* optionalen Parameter einer Funktion gesetzt wird.
15: /* Nicht so ganz ANSI.
16: /* ptr va_list Zeiger
17: /* type der Typ des letzten nicht optionalen
18: /* Parameters
19: /* last der letzte nicht optionale Parameter
20:
21: #define va_start( ptr, type, last ) \
22: ( ( ( type * ) ptr = ( &(last) ) ), \
23: ( ( type * ) ( ptr ) )++ )
24:
25: /* Makro das den nächsten optionalen Parameter
26: /* zurückgibt. ( ANSI )
27: /* ptr va_list Zeiger
28: /* type Typ des zuholenden optionalen Parameters
29:
30: #define va_arg( ptr, type ) \
31: ( * ( ( type * ) ( ptr ) )++ )
32:
33: /* Makro zur Beendigung der va_* Routinen. ANSI.
34: /* Dient eigentlich dazu dem Compiler mitzuteilen,
35: /* das keine optionalen Parameter mehr gebraucht
36: /* werden. Nur der Vollständigkeit zu ANSI halber.
```

```
37: /* ptr va_list Zeiger
38:
39: #define va_end( ptr )
40:
41: #endif /* Von #ifndef __STDARG_H__ */
```

Listing 2: Etwas ANSI-Feeling für Parameter

```
*****
* AMIGA DOS INFO
* Programmname : Listing 3
* Funktion : siehe Text
* Sprache : Standard-C
* Besonderheiten : keine
*****
1: /* Programmbeispiel für die va_* Makros */
2: #include "stdio.h" /* Standard Input Output
3: #include "stdarg.h" /* Die va_* Makros
4: #include "mydef.h" /* s. List. 1
5:
6: /* Deklaration einer Funktion mit optionalen Para-
7: /* metern.
8:
9: WORD16 test( WORD16, );
10:
11: /* Hauptprogramm zum Test der Funktion test und
12: /* der va_* Makros. test wird mit 4 Parametern
13: /* unterschiedlichen Typs aufgerufen.
14:
15: main()
16: {
17: WORD32 j = 24;
18:
19: printf("\n%d\n", test( 8, (char)4, 6L, &j ));
20: }
21:
22: /* Die Funktion test hat einen festen Parameter und
23: /* beliebig viele optionale. Um das Beispiel zu
24: /* vereinfachen, wird angenommen, das genau 3 Para-
25: /* meter mit bekanntem Typ folgen.
26:
27: WORD16 test( first, )
28: WORD16 first;
29: {
30: va_list zeiger; /* Zeiger für va_start
31: WORD16 i; /* Zum Zusammenzählen
32:
33: /* Initialisierung von zeiger auf den ersten
34: /* optionalen Parameter
35: va_start( zeiger, WORD16, first );
36:
37: /* Den festen Parameter nach i
38: i = first;
39:
40: /* Den ersten optionalen Parameter zu i dazu-
41: /* zählen. Er ist in main mit einem typecast,
42: /* einen char Wert umgewandelt worden, muß hier
43: /* aber als int angegeben werden. Siehe hierzu
44: /* Text.
45: i += va_arg( zeiger, WORD16 );
46:
47: /* Nun die beiden anderen Parameter auch zu i
48: /* zählen
49: i += ( WORD16 )va_arg( zeiger, WORD32 );
50: i += ( WORD16 )*va_arg( zeiger, WORD32 * );
51:
52: /* i als Funktionswert zurückgeben
53: return( i );
54: }
55:
56: /* Als Ergebnis sollte, nach dem Compilieren,
57: /* links usw., 42 als Antwort auf dem Bildschirm
58: /* erscheinen.
```

Listing 3: Ein Test (Die Antwort ist 42, doch wie war die Frage?)

START...

...in neue
Amiga-
Dimensionen



Amiga-Tuning durch Hardwareerweiterungen, Systemprogrammierung, Tips und Tricks

Dieses neue Nachschlagewerk erläutert Ihnen übersichtlich und detailliert, wie Sie durch gezieltes Tuning die Fähigkeiten Ihres Amiga erweitern und optimal ausschöpfen.

Aus dem Inhalt:

- **Lernen Sie die Bausteine Ihres Amiga genau kennen:** In einem eigenen Hardwarekapitel erschließen Sie sich den Aufbau Ihres Amiga. Der umfassenden Beschreibung der Motorola-CPU und der Customchips Agnus, Paula, Denise und Gary entnehmen Sie Architektur, Register und Befehlssatz sowie die Abläufe bei Adressierung und Befehlsausführung.
- **Amiga-Tuning durch Hardwareerweiterungen:** Für alle, die gern mit dem Lötkolben hantieren, hält das Werk präzise Bauanleitungen mit Platinenlayout, Schaltbild und Bestückungsplan für preiswerte Selbstbau-Erweiterungen bereit, z.B. Sound-Digitizer, EPROM-Brenner.
- **Sie haben die Wahl: Basic, C, Assembler...** In Kursen eignen Sie sich eine ausgefeilte Programmierertechnik an. Mit den Übungsbeispielen verfügen Sie gleichzeitig über Module für eigene Programme.
- **Praxiserprobte Musterlösungen zum Sofort-einsatz,** u. a. zu Desktop-Publishing und Genlock-Anwendungen, Dateiverwaltung, Meßwerterfassung und -verarbeitung.
- **3D-Grafik – Animation – Video: lassen Sie Ihrer Kreativität freien Lauf!** Erschließen Sie sich das breite Spektrum der Amiga-Grafik: von einfachen geometrischen Körpern über 3D bis hin zur Animation.
- **Software-Engineering** ist gefragt: Hinweise zum Einbinden von Intuition-Funktionen für Menüs, Alerts od. Systemrequester ermöglichen Ihnen, den Anwenderkomfort Ihrer Programme zu erhöhen.

Fordern Sie noch heute an:

Hard- und Softwaretuning für die Amiga-Familie

praktischer Ringbuchordner, DIN A4, Grundwerk ca. 480 Seiten, inkl. Diskette, Bestell-Nr.: 4800/74800, Preis: DM 98,-

Alle 2 – 3 Monate erhalten Sie Ergänzungsausgaben zum Grundwerk mit je ca. 130 Seiten, Seitenpreis 38 Pfennig (Abbestellung jederzeit möglich).



...in neue
68000er
Dimensionen

Perfektes Anwender-, System- und Bausteinprogrammieren in Assembler und C

Was bisher nur Minis und Mainframes vorbehalten war, schaffen jetzt auch Mikrocomputer mit 680xx CPUs. Nutzen Sie diese Vorzüge! Das neue 68000er-Handbuch vermittelt Ihnen hierzu detailliertes Hardwarewissen, fundierte Betriebssystemkenntnisse und ausgefeiltes Programmier-Know-how in Assembler und C.

Die wichtigsten Themen auf einen Blick:

- **Detaillierte Hardwarebeschreibungen** der 68000er-Prozessorfamilie: Die Prozessoren 68000, 68008, 68020 und 68030 sowie die 8-, 16-, 32-Bit-Peripheriebausteine werden ausführlich beschrieben.
- **Vertieftes Betriebssystem-Know-how:** Sie lernen die Strukturen der 68000er-Betriebssysteme (wie GEM-TOS, OS-9, Unix und Amiga-DOS) kennen.
- **C-Programmierung:** In C erstellen Sie schnelle und kompakte Programme und bedienen sich zugleich einer Programmiersprache, die sich bestens zur Systemprogrammierung eignet.
- **Programmierkurs Assembler:** Durch maßgeschneiderte Befehlsfolgen, die direkt die Hardware und das Betriebssystem ansprechen, wird optimaler Code generiert.
- **Tools und Utilities** inkl. einem universellen symbolischen Disassembler zur Bearbeitung fremder Programme, die nicht im Quellcode vorliegen.
- **Bewährte Musterlösungen** wie zum Erstellen von 3D-Grafiken, Auswertung statistischer Daten.
- **Systemtuning durch Hardwareerweiterungen:** Ausführliche Anleitungen zum sicheren Nachbau z.B. von Ein-/Ausgangsinterfaces für Atari ST oder DFÜ-Modem für Amiga.

Fordern Sie noch heute an:

Erfolgreiches Programmieren von 68000er Systemen in Assembler und C

praktischer Ringbuchordner, DIN A4, Grundwerk ca. 550 Seiten, Bestell-Nr.: 3400, Preis: DM 92,-

Alle 2 – 3 Monate erhalten Sie Ergänzungsausgaben zum Grundwerk mit je ca. 130 Seiten, Seitenpreis 38 Pfennig (Abbestellung jederzeit möglich).



INTEREST-VERLAG

Fachverlag für
Special Interest Publikationen
und Anwendersoftware
Industriestraße 21 · D-8901 Kissing
Telefon 0 82 33/21 01-0



Mit Deluxe Paint wie ein Profi arbeiten

Allein der Umstand, daß Sie, lieber Leser, zu Hause einen Amiga stehen haben, gewährleistet, daß Sie schon einmal die faszinierenden Grafikmöglichkeiten dieses leistungsfähigen Computers gesehen haben. Zum Erstellen solcher Bilder gibt es Werkzeuge, Grafikprogramme, die den Computer in eine elektronische Zeichenwerkstatt verwandeln. Eines der leistungsfähigsten Programme dieser Art ist Deluxe Paint. Im Laufe dieses Kurses möchten wir Ihnen zeigen, was Sie alles mit Deluxe Paint anstellen können.

Kursfahrplan

Teil 1: Das Basismenü und seine Funktionen

Teil 2: Die Kopfleiste und die dort enthaltenen Menüs

Teil 3: Die Anwendung der in der Kopfleiste enthaltenen Funktionen

Teil 4: Animation mit dPaint III

Teil 5: Arbeitstips vom Profi

Einzelne Pixel mittels einer Programmiersprache zu setzen, ist nicht leicht und sehr arbeitsintensiv, von der Verwendung von Assembler ganz zu schweigen. Es mußten Werkzeuge geschaffen werden, die geeignet waren, diese Aufgabe zu erleichtern.

Um diese Lücke zu schließen, wurde Deluxe Paint entwickelt. Aufgrund seines enormen Leistungsumfangs eroberte sich Deluxe Paint einen festen Platz im Rahmen der Amiga-Anwendersoftware. Wie dieses mächtige Werkzeug benutzt wird, das wollen wir Ihnen im Rahmen

dieses Artikels in Wort und Bild zeigen.

Ein Buch mit sieben Siegeln – Deluxe Paint und seine Dokumentation

Obwohl die Dokumentation von dPaint nur als ausgezeichnet beschrieben werden kann, gibt es gerade hier viele Probleme, da das ganze Handbuch in englischer Sprache gehalten ist. Viele Anwender sind deshalb nicht in der Lage, alle Funktionen und Optionen zu nutzen. Dieser Beitrag möchte Ihnen die Arbeit mit dPaint nahebringen und Sie in die Anwendung des Programms einführen. Wir werden Ihnen zeigen, was Sie mit einem Pinsel alles anstellen können, was mit den verschiedenen Fill-

Typen machbar ist, wie Sie mit der Perspektive Funktion erreichen können und vieles mehr. Folgen Sie uns also auf eine interessante und lehrreiche Reise durch die weite Welt der dPaint-Anwendung.

Am Anfang ist das Menü – mit kleinen Icons Großes leisten

Gehen wir davon aus, daß Sie Ihren Amiga bereits angeschaltet haben, dPaint gestartet und im 'Screenformat Requester' das Bildschirmformat 320x256 bei 32 verschiedenen Farben eingestellt haben. Nach Bestätigung dieser Einstellung gelangen Sie in den Zeichenbildschirm von dPaint. Hier finden wir im wesentlichen einen schwarzen Bildschirm, eine Kopfzeile und ein vertikal am rechten Rand positioniertes Menü vor, das wir im folgenden als Basismenü bezeichnen wollen. In diesem Menü sind hauptsächlich die elementaren Funktionen von dPaint enthalten und können von hier aus durch Mausklick eingeschaltet werden. Denken Sie daran, eine einmal eingeschaltete Funktion wird nicht durch nochmaliges Anklicken des Icons wieder ausgeschaltet, sondern durch die Auswahl der nun gewünschten Funktion. Ganz oben im Basismenü finden Sie eine kleine Sammlung von unterschiedlich großen Kreisen und Vierecken. Dabei handelt es sich um die 'Zeichenstifte', die Ihnen dPaint von vornherein zur Verfügung stellt. (Natürlich können Sie auch eigene Stifte definieren und benutzen, doch dazu später.) Nach dem Start von dPaint ist der erste, links oben im Stiftwindow angezeigte Stift aktiv und somit invertiert dargestellt. Die Auswahl anderer Stifte geschieht einfach über Mausklick. Das untere Ende des Basismenüs nimmt der Farbindikator ein. Die 32 Farbfelder, die Sie hier sehen, stellen die Farben dar, die momentan eingestellt sind und somit zur Erstellung Ihres Bildes zur Verfügung stehen. Die gerade aktive Farbe wird durch einen kleinen weißen Rand gekennzeichnet. Über dem Farbindikator sehen Sie ein etwas größeres Rechteck, in dessen Mitte ein Kreis angezeigt ist. Die Farbe des Kreises stellt die aktivierte Zeichenfarbe, und die um-

gebende Farbe die aktuelle Hintergrundfarbe dar. Die Hintergrundfarbe wird auch oft als 'Transparent-Farbe' bezeichnet. Unmittelbar nach dem Starten von dPaint ist die Zeichenfläche schwarz. Dieses Schwarz nimmt in der Palette das Farbbregister 0 ein. Register 0 ist immer die Transparent-Hintergrundfarbe. Außer dem Stiftmenü und dem Farbindikator sind noch weitere 18 Icons im Basismenü enthalten, die nun, entsprechend ihrer Abfolge (siehe Abb. 1), erläutert werden sollen.

Das Who is Who der Icons – was sich so alles im Basismenü versteckt

Nun wollen wir uns den restlichen Icons des Grundmenüs widmen und deren grundsätzliche Funktionsweise erklären. Am besten ist sicherlich, wenn Sie anhand der folgenden Ausführungen mit jeder Funktion ein wenig herumexperimentieren. Auf diese Weise gewinnen Sie gleich zu Anfang mehr Sicherheit im Umgang mit den einzelnen Funktionen.

Punktirtes Freihandzeichnen
Ist die Funktion aktiviert, können Sie mittels der Maus auf dem Bildschirm zeichnen. Mit welcher Farbe Sie zeichnen, können Sie am Farbindikator ablesen. Drücken Sie die linke Maustaste und halten Sie sie fest. Bewegen Sie nun den Mauszeiger über den Monitor, erscheint eine Punktklinie, die Ihren Mausbewegungen entspricht. Lassen Sie den Mausknopf wieder los, endet der Strich, Sie haben quasi den Stift vom Papier genommen.

Kontinuierliches Freihandzeichnen
Im Grunde funktioniert diese Option wie die vorhergehend beschriebene, nur hinterläßt sie beim Zeichnen eine ununterbrochene Linie. Hinter dem kontinuierlichen Freihandzeichnen-Icon verbirgt sich aber noch eine weitere Option. Wenn Sie sich das Icon genau ansehen (solange es nicht aktiviert ist), werden Sie feststellen, daß eine diagonale Linie das Icon von rechts oben nach links unten zweiteilt. Ein Mausklick links oben aktiviert das kontinuierliche Freihandzeichnen. Ein Mausklick rechts unten akti-

viert eine andere. Mit dieser Funktion können Sie freihändig eine Fläche beliebiger Form zeichnen, die, wenn die Fläche geschlossen wurde, automatisch mit der aktuellen Zeichenfarbe gefüllt wird. In der Menüleiste, mit der wir uns gerade beschäftigen, werden wir noch mehreren anderen Icons mit Doppelfunktion begegnen. Ihnen allen ist der diagonale Trennstrich von rechts oben nach links unten gemeinsam.

Gerade Linie

Mit dieser Funktion können Sie beliebig lange, gerade Linien auf den Bildschirm zeichnen. Führen Sie den Mauszeiger an die gewünschte Anfangsposition der Linie, klicken Sie die rechte Maustaste und halten Sie sie fest. Solange Sie die Maustaste nun festhalten, ziehen Sie eine Linie zwischen der angewählten Startposition der Linie und Ihrem Cursor. Bewegen Sie den Cursor nun an die Stelle, an der die Linie enden soll, und lassen Sie die rechte Maustaste los. Voilà, die Linie steht.

Kurven – Linien

Mit dieser Funktion können Sie gebogene Linien zeichnen. Wählen Sie diese Funktion im Grundmenü aus und klicken Sie sie an. Bewegen Sie den Cursor auf die Zeichenfläche. Steht der Cursor an der Stelle, die der Ausgangspunkt der gewünschten Linie sein soll, betätigen Sie die linke Maustaste und halten Sie sie fest. Nun bewegen Sie den Cursor an die gewünschte Endposition der Linie und lassen Sie die Maustaste los. Der Computer verbindet die beiden Punkte nun mit einer Linie. Diese Linie können Sie nun so wölben bis die gewünschte Kurve erreicht ist. Ein weiterer Klick auf die rechte Maustaste, und die Operation ist beendet.

Füllen

Hiermit können Sie eingegrenzte Flächen einfärben. Neben der Benutzung einer einfachen Farbe, bietet dPaint noch einige andere Fülloptionen, auf die wir an anderer Stelle noch zu sprechen kommen werden. Aktivieren Sie das Füll-Icon via Mausklick und wählen Sie im Farbindikator eine Farbe aus, die Sie bisher noch nicht auf der Zeichenfläche haben. Bewegen Sie nun den Cursor auf die Zeichenfläche, werden Sie feststellen, daß er sich verän-

dert hat. Merken Sie sich, daß Ihr Cursor, solange die Füll-Funktion aktiviert ist, immer so aussieht. Betätigen Sie nun die linke Maustaste und beobachten Sie, was passiert.

Spritzen

Diese Funktion ist das elektronische Äquivalent zu der bei Grafikern sehr beliebten Spritzpistole. Aktivieren Sie das Icon durch Betätigen der linken Maustaste. Führen Sie nun den Cursor auf die Zeichenfläche, drücken Sie die

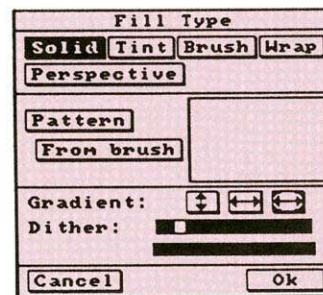


Schaubild 1: Der Fill-Type-Requester, ein leistungsfähiges Werkzeug

linke Maustaste, halten Sie diese fest und bewegen Sie nun den Cursor über die Zeichenfläche. Sie werden sehen, daß eine dicke Linie, bestehend aus unregelmäßig gesetzten Farbpunkten, entsteht. Halten Sie den Cursor mit gedrückter Maustaste länger an einer Stelle, können Sie beobachten, wie immer mehr Punkte entstehen.

Rechtecke – gefüllt, ungefüllt

Das Rechteck-Icon steht eigentlich für zwei verschiedene, jedoch ähnliche Funktionen. Wird das Icon in der linken oberen Ecke aktiviert, verändert sich das Icon in eine schwarze Fläche, in der ein hohles Rechteck steht. Ist diese Darstellung aktiv, werden hohle Rechtecke, die nur aus den vier Begrenzungslinien bestehen, gezeichnet. Aktivieren Sie das Icon in der rechten unteren Ecke, wird ein inverses, gefülltes Rechteck dargestellt. Bewegen Sie den Cursor mit aktiviertem Rechteck-Icon auf die Zeichenfläche, erscheint ein Fadenkreuz, dessen Zentrum immer die Position des Cursors ist. Führen Sie nun den Cursor an die Stelle, an der die linke obere Ecke des gewünschten Rechtecks sein soll, betätigen Sie die linke Maustaste, und halten Sie sie fest. Jetzt brauchen Sie nur

noch den Cursor an die Stelle führen, an der sich die rechte untere Ecke des Rechtecks befinden soll und lassen die Maustaste los. Aktivieren Sie nun die Option gefülltes Rechteck und verfahren Sie genauso. Jetzt sollten Sie ein gefülltes und ein ungefülltes Rechteck auf der Zeichenfläche haben.

Kreise – gefüllt, ungefüllt

Auch dieses Icon steht für zwei Funktionen. Wie schon beim Rechteck wird hier durch die Icon-Aktivierung in der linken oberen Ecke die Funktion hohler Kreis und in der rechten unteren Ecke die Funktion gefüllter Kreis aktiviert. Auch die Kreisfunktion legt ein Fadenkreuz unter den Cursor, bzw. das Fadenkreuz an die Stelle der Zeichenfläche, an der sich der Mittelpunkt des gewünschten Kreises befinden soll. Betätigen Sie die linke Maustaste und halten Sie sie fest. Bewegen Sie jetzt den Cursor vom Mittelpunkt weg, und Sie werden sehen, wie ein Kreis entsteht. Lassen Sie nun die Maustaste wieder los, und der gewünschte Kreis wird endgültig auf die Zeichenfläche gebannt. Die Arbeitsweise mit der Option gefüllter Kreis ist die gleiche.

Ellipsen – gefüllt, ungefüllt

Diese Funktion arbeitet wie die Kreis-Option, nur daß wir hier als Ergebnisse gefüllte, bzw. ungefüllte Ellipsen erhalten.

Vielecke – gefüllt, ungefüllt

Da in Grafiken mitunter auch Formen auftauchen, die mit Kreisen, Rechtecken und Ellipsen nichts zu tun haben, bedarf es für ein Grafikprogramm weiterer Werkzeuge. Mit dieser Funktion können vieleckige, geschlossene Formen, gefüllt oder ungefüllt, gezeichnet werden (z.B. Sterne). Auch hier ist die Funktionsweise denkbar einfach. Sie aktivieren das Icon (links oben ungefüllt, rechts unten gefüllt) und bewegen Ihren Cursor auf die Zeichenfläche. Wenn Sie nun einmal die linke Maustaste anklicken und wieder loslassen, haben Sie damit den Ausgangspunkt der die Fläche umgebenden Linie gesetzt. Wenn Sie nun den Cursor bewegen, werden Sie feststellen, daß immer eine Linie zwischen dem Ausgangspunkt und dem Cursor ist. Bewegen Sie nun den Cursor an die Stelle, an der sich

der nächste Eckpunkt der von Ihnen gewünschten Form befinden soll. Ein weiterer Klick auf der linken Maustaste und der zweite Eckpunkt ist gesetzt. Fahren Sie in gleicher Art und Weise fort, bis das gewünschte Vieleck komplett ist, bzw. Sie mit Ihrer Linie wieder am Ausgangspunkt angekommen sind. Die Arbeitsweise mit der <gefülltes Vieleck>-Funktion ist gleich.

Zeichenstifte definieren

Diese Funktion ist eine der meistbenutzten dPaint-Optionen überhaupt. Es ermöglicht die Benutzung frei wählbarer Bildausschnitte als Zeichenstift. Der Größe dieser Bildausschnitte sind nur durch den zur Verfügung stehenden Speicherplatz Grenzen gesetzt. Auch hier wurden wieder zwei Funktionen

in einem Icon vereint. Zum einen die Option, mit der rechteckige Bildausschnitte als Pinsel ausgewählt werden können, zum anderen die Option, dies mit vieleckigen Bildausschnitten zu tun. Wählen Sie die erste Möglichkeit aus, und bewegen Sie den Cursor auf die Zeichenfläche, sie werden feststellen, daß wieder ein Fadenkreuz erscheint. Wie bei der Rechteck-Funktion definieren Sie auch hier wieder erst die linke obere Ecke des gewünschten Bildausschnittes. Klicken Sie die linke Maustaste an und halten Sie sie fest. Nun bewegen Sie den Cursor an die rechte untere Ecke des gewünschten Ausschnittes und lassen Sie hier die Maustaste wieder los. Sie werden sehen, daß nun der eben von Ihnen definierte Bildausschnitt als Zeichen-

stift unter Ihrem Cursor dargestellt wird. Wählen Sie die 'Vieleck'-Option aus, verfahren Sie wie weiter oben unter dem Punkt 'gefülltes, ungefülltes Vieleck' beschrieben. Nur daß Sie diesmal als Ergebnis den definierten Bildausschnitt als Zeichenstift unter dem Cursor haben.

Text

dPaint ist zwar keine Textverarbeitung, doch immer wieder werden bei der Erstellung von Computerbildern auch Texte verwendet. Damit der Benutzer nun nicht jeden Buchstaben zeichnen muß, wurde die Textoption eingebaut. Klicken Sie das Icon an, und bewegen Sie den Cursor auf die Zeichenfläche. Nun hängt ein kleines Rechteck an Ihrem Mauszeiger, das Sie über den Bildschirm bewegen

können. Wählen Sie nun die Farbe aus, in der Ihr Text auf dem Monitor erscheinen soll, und führen Sie den Cursor, samt daranhängendem Rechteck an die Stelle des Bildschirms, an der der Text beginnen soll. Wenn Sie nun einmal die linke Maustaste betätigen, bleibt das Rechteck an der angeklickten Position stehen, und Sie können nun, wie man es von einer Schreibmaschine gewöhnt ist, über die Tastatur Texte eingeben.

Gitter

Diese Funktion ist eigentlich kein unmittelbares Zeichenwerkzeug. Sie können mit Gitter nichts auf die Zeichenfläche malen, sondern eine Art unsichtbares Raster über die Zeichenfläche legen. Die Abstände der einzelnen vertikalen und horizontalen Gitter

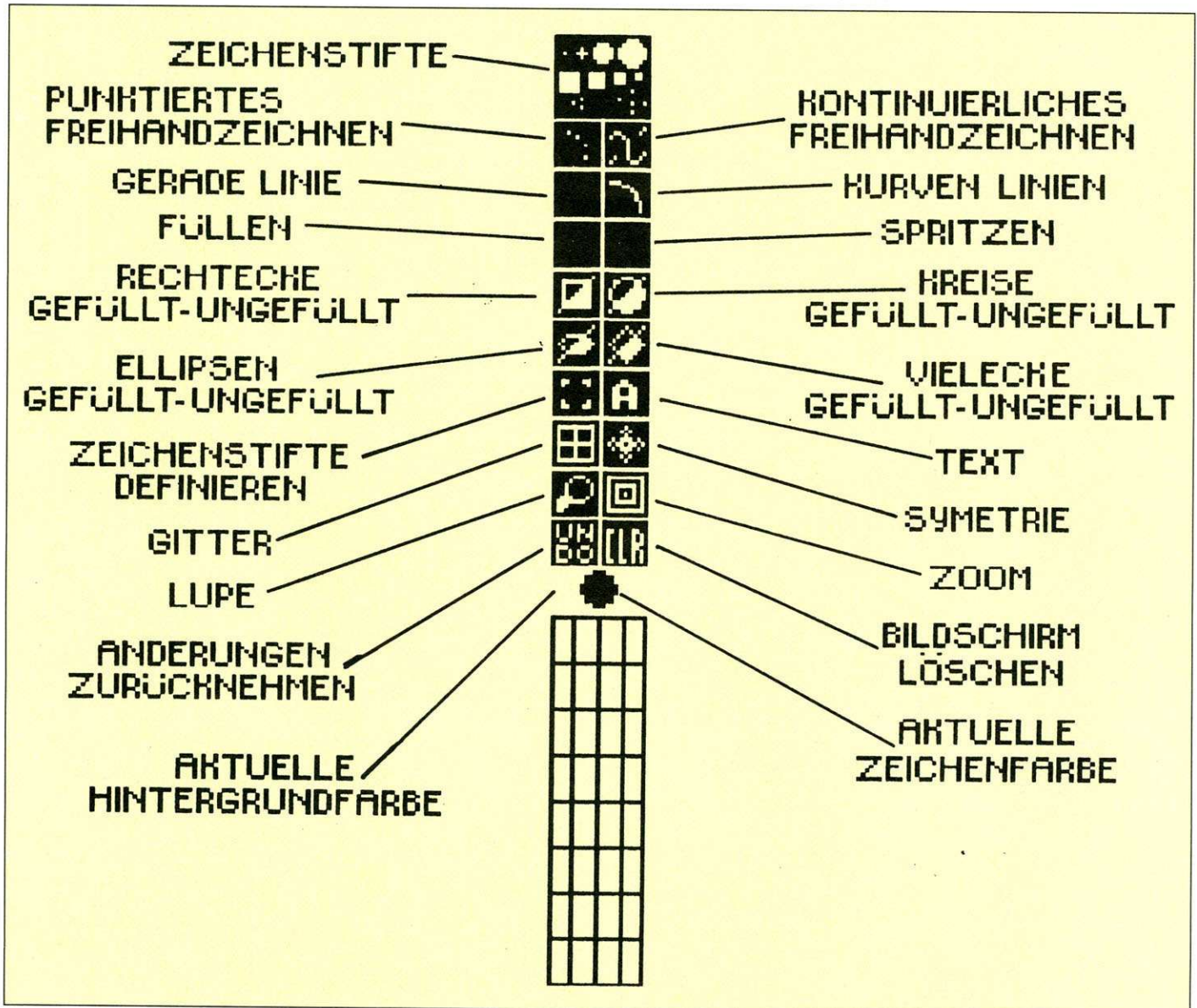


Schaubild 2: Das dPaint-Basismenü und die dort enthaltenen Funktionen

terlinien können frei gewählt werden (wie, erklären wir noch). Die Gitterfunktion ist nach dem Programmstart auf vertikal acht und horizontal acht eingestellt. Aktivieren Sie die Option <Freihandzeichnen> und danach <Gitter>. Sie werden feststellen, daß Sie zwar den Cursor immer noch an jede beliebige Stelle der Zeichenfläche positionieren können, die Darstellung Ihres Zeichenstiftes beim Bewegen der Maus den Eindruck erweckt, als würde sie über die Zeichenfläche hüpfen. Dieser Eindruck entsteht durch den Umstand, daß mit aktivierter Gitterfunktion der Pinsel nur an den Kreuzungen des Gitter erscheinen kann. Halten Sie nun die linke Maustaste gedrückt und bewegen Sie den Cursor über die Zeichenfläche. Der nun sichtbare Effekt veranschaulicht deutlich die Wirkungsweise der 'Gitter'-Funktion.

Lupe

Mitunter kommt es vor, daß man beim Erstellen und Bearbeiten von Grafiken an kleinen Bilddetails arbeiten muß. Damit der Benutzer sich nicht die Nase am Monitor plattdrücken muß, um zu sehen, was er dort gerade anstellt, haben die dPaint-Macher die Lupenfunktion eingebaut. Klicken Sie das Lupensymbol in der Menüleiste einmal mit der rechten Maustaste an, und bewegen Sie dann den Cursor auf die Zeichenfläche. Rund um Ihren Cursor sollte sich nun ein weißer Rahmen befinden, den Sie frei über die Zeichenfläche bewegen können. Positionieren Sie den Rahmen so, daß er die Bereiche der Grafik, die Sie vergrößert dargestellt haben möchten, umschließt. Betätigen Sie nun noch einmal die rechte Maustaste. Daraufhin wird der Bildschirm in zwei Teile gesplittet. In der rechten Hälfte der Zeichenfläche wird nun der angewählte Bildausschnitt vergrößert dargestellt, und links daneben derselbe Bereich, nur diesmal unvergrößert.

Zoom

(vergrößern - verkleinern)

Zoom ist eigentlich ein Begriff aus der Fotografie und meint einen Vergrößerungs- bzw. einen Verkleinerungsvorgang. Mittels der Zoom-Funktion kann der Anwender den Vergrößerungsfaktor der Lupenfunktion manipulie-

ren. Diese Funktion kann mit und ohne eingeschalteter Lupe benutzt werden. Um den Effekt der Funktion besser sehen zu können, sollten Sie die Lupe einschalten und einen beliebigen Bildausschnitt vergrößern. Bewegen Sie nun den Mauszeiger auf das Zoom-Icon. Hier können Sie durch Mausklick links im Zoom-Icon die vergrößerte Darstellung noch weiter vergrößern. Mausklick rechts hat die umgekehrte Funktion und verkleinert die Lupendarstellung.

Änderung zurücknehmen

Die Kreativität sprudelt, der dPaint-Benutzer tobt sich auf der Zeichenfläche aus, und rums, in der Hitze des Gefechtes passiert ein Fehler. Die letzte gezeichnete Linie ist vollkommen verkehrt und macht die Arbeit der letzten Stunde zunichte! An dieser Stelle kommt die 'Undo'-Funktion ins Spiel. Durch das Anklicken dieses Icons wird die letzte Änderung am Bild rückgängig, also ungeschehen gemacht. Bei dem eingangs angeführten Beispiel würde nach dem Mausklick im Undo-Icon eben die letzte unerwünschte Linie verschwinden. Zeichnen Sie nun irgendeine Linie auf die Zeichenfläche und klicken Sie das Undo-Icon an. Wie schon beschrieben, wird die eben gezeichnete Linie daraufhin verschwinden. Sie haben den letzten Zeichenschritt, also die Linie, ungeschehen gemacht. Nun klicken Sie noch einmal das Undo-Icon an. Nun erscheint die Linie wieder. Mit dem zweiten Undo-Klick haben Sie den ersten Mausklick auf das Undo-Icon zurückgenommen. Die Undo-Funktion ist eine der dPaint-Optionen, die in der praktischen Anwendung oft benutzt werden. In diesem Zusammenhang ist es interessant zu wissen, daß jeder Anschlag auf der Tastatur des Amigas von dPaint als letzte Änderung gespeichert wird. Wenn Sie also eine Linie zeichnen, dann eine Taste auf der Tastatur betätigen und nun mit Undo die Linie wieder wegnehmen wollen, geht das nicht mehr, da die letzte Änderung, die von dPaint gespeichert wurde, nicht mehr die Linie ist, sondern das danach getätigte Tastaturkommando.

(CLR) Bildschirm löschen

Eine dPaint-Funktion, die nur mit Bedacht eingesetzt wer-

den sollte. Ein Mausklick auf diesem Icon und der ganze Bildschirm wird gelöscht.

Eine Funktion also, die bei der Arbeit mit dPaint zwar unerlässlich ist, aber mit Umsicht eingesetzt werden sollte. Nun kennen Sie alle dPaint-Funktionen, die in der rechten, senkrechten Menüleiste angebracht sind, und können nach Belieben damit herumexperimentieren. Sie werden feststellen, daß die meisten Funktionen mit der linken Maustaste aktiviert werden. Benutzen Sie die rechte Maustaste, zeitigt das nicht immer den gewünschten Effekt. Tatsächlich verhält es sich so, daß durch das Anklicken eines Icons mit der rechten Maustaste Menüs aufgerufen werden können, in denen die Funktionen modifiziert werden können. Nachdem wir eben einen Streifzug durch das Menü und den einzelnen darin enthaltenen Funktionen gemacht haben, werden wir diesen Weg wiederholen. Nur diesmal werden wir sehen, welche Funktionen modifiziert werden können, wie dies geschieht und welche Resultate es zeitigt. Der Einfachheit halber lassen wir alle Funktionen aus, hinter denen sich keine Menüs verbergen, bzw. die keine Modifizierung zulassen.

Modifizieren leichtgemacht - wie Deluxe-Paint-Funktionen angepaßt werden können

Die erste Station im Basismenü ist das Zeichenstift-Auswahlfenster. Wie gesagt befinden sich hier die Stiftformen, die Ihnen dPaint von Anfang an zur Verfügung stellt. Klicken Sie nun einen beliebigen Stift mit der rechten Maustaste an und bewegen Sie nun den Mauszeiger wieder auf die Zeichenfläche.

Der Mauszeiger hat sich ein wenig verändert. Er besteht nun aus der Darstellung des Stiftes, unter dem senkrecht das Wort 'Size' (Größe) steht.

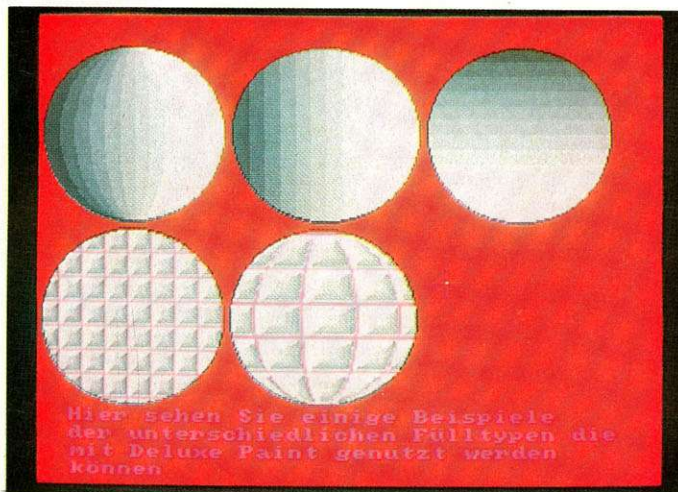
Halten Sie nun die linke Maustaste fest, und bewegen Sie die Maus. Die Größe des ausgewählten Stiftes verändert sich. Sie können auf diesem Wege jede der eingebau-

ten Zeichenstifte stufenlos vergrößern bzw. verkleinern.

Kontinuierliches Freihandzeichnen

Klicken Sie das Icon für kontinuierliches Freihandzeichnen mit der rechten Maustaste an, öffnet sich auf der Zeichenfläche der 'Fill Type Requester' (Füllart Nachfrager, Abb. 2). dPaint kennt verschiedene Wege, eine Fläche mit Farbe, Farben oder Mustern zu füllen. Im 'Fill Type Requester' können Sie einstellen, welche Füllart Sie benutzen möchten. Dieser Requester wird uns bei der Arbeit mit dPaint noch des öfteren begegnen; im Grunde genommen immer dann, wenn eine Funktion etwas mit dem Füllen von Flächen zu tun hat (z.B. bei <gefüllten Kreisen>, <gefüllten Rechtecken> oder <gefüllten Vielecken>). Ganz oben in dem Window finden Sie vier kleine Kästchen, die mit <Solid>, <Tint>, <Brush> und <Wrap> bezeichnet sind. Unmittelbar darunter befindet sich ein weiteres, etwas längeres Kästchen, das mit <Perspective> bezeichnet ist. Jedes Kästchen repräsentiert eine andere Füllart. Das Kästchen, das schwarz mit heller Schrift dargestellt ist, zeigt die gerade angewählte Füllart an. Im nächsten Teilbereich gibt es wieder zwei beschriftete Kästen, <Pattern> und <From Brush>. Rechts daneben positioniert ist ein Window, das momentan noch leer sein sollte. Im folgenden Teil des Requesters finden wir das Wort <Gradient>, gefolgt von drei Kästchen, in denen sich diverse Pfeile befinden. Darunter befindet sich die Bezeichnung <Dither>, neben der ein Schieberegler liegt. Unmittelbar darunter befindet sich ein weiteres Window. Auch dieses sollte noch leer sein. Das Ende des Requesters wird durch zwei weitere Kästchen gebildet, die mit 'Cancel' (Abbruch) und 'OK' bezeichnet sind. Wenn Sie in der Einstellung des Requesters Änderungen vorgenommen haben, mit denen Sie arbeiten möchten, klicken Sie 'OK' an. Sind die Änderungen hinfällig und sie sollen nicht benutzt werden, klicken Sie 'Cancel' an. Sehen wir uns zunächst die einzelnen Füllarten und deren Wirkungsweise an. <Solid> füllt eine Fläche mit der aktuellen Zeichenfarbe. Rufen

Sie den Fill Type Requester wie beschrieben auf und aktivieren Sie das Kästchen, das mit 'Solid' beschriftet ist. Bestätigen Sie die Einstellung durch einen Mausklick auf 'OK'. Aktivieren Sie nun das Fill-Icon und stellen Sie eine beliebige Farbe ein. Wenn Sie Ihren Cursor nun auf die Zeichenfläche bewegen und dort einmal die rechte Maustaste drücken, können Sie zusehen, wie Ihr Amiga die betreffende Fläche mit der angewählten Farbe füllt. Die <Tint>-Füllart unterscheidet sich grundlegend von der <Solid>-Füllart. <Tint> unterscheidet zwischen Zeichenfarben (Register 1 bis 31) und der Transparentfarbe (Register 0), die vom <Tint>-Füllmode nicht benutzt wird. Im Klartext heißt das: Wenn Sie versuchen, im <Tint>-Mode einen leeren Bildschirm (dieser ist ja mit der Transparentfarbe aus Register 0 dargestellt) zu füllen, wird Ihr Amiga einen Moment arbeiten. Hat er seine Operationen jedoch beendet, ist keine Veränderung festzustellen. Versuchen wir es anders. Wählen Sie den dunkelsten Grauton aus der original dPaint-Palette, und zeichnen Sie damit ein einfarbig gefülltes Rechteck. Wählen Sie nun den dunkelsten Blauton aus, rufen Sie den Fill Type Requester auf und stellen Sie <Tint> ein. Versuchen Sie nun, das graue Rechteck mit dem dunklen Blau zu füllen. Erstaunliches geschieht: Statt die graue Fläche blau einzufärben, verschwindet das graue Rechteck und hinterläßt einen leeren Bildschirm. Was ist passiert? Im <Tint>-Füllmodus ermittelt dPaint zunächst einmal die Farbwerte der Farbe, auf der gezeichnet werden soll und die Werte der Farbe, mit der gezeichnet werden soll. Basierend auf der Analyse der beiden Farbwerte ermittelt dPaint aus den übrigen Registern die Farbe, die den beiden angewählten Farben von den Farbwerten her am ähnlichsten ist. Im Falle unseres Beispiels war diese Farbe eben Schwarz. Die nächste Füllart ist <Brush>. Erinnern Sie sich an die Funktion <Zeichenstifte definieren>? Wenn nicht, sollten Sie schnell noch einmal weiter oben nachlesen. Denn der <Brush>-Füllmode ermöglicht es dem Anwender, Teile seines Bildes als Füllmuster



Daß Deluxe Paint ein außerordentlich leistungsfähiges Werkzeug ist, sieht man schon an den vielen verschiedenen Möglichkeiten, eine Fläche zu füllen

zu verwenden. Zeichnen Sie ein Quadrat mit einer Kantenlänge von fünf Pixeln. Definieren Sie dieses Quadrat als Zeichenstift. Nun wird der Fill Type Requester aufgerufen – klicken Sie hier das Feld <From Brush> an. Wie Sie sehen, füllt sich daraufhin das Window daneben mit vielen Quadraten, die alle eine Kantenlänge von fünf Pixeln haben. Klicken Sie 'OK' an und füllen Sie die Zeichenfläche. Jetzt verwendet dPaint keine Farbe zum Füllen, sondern das eben von Ihnen erstellte Quadratmuster. Der vorletzte Fülltyp ist <Wrap> (Wickeln). Zeichnen Sie zunächst einmal einen mit einer beliebigen Farbe gefüllten Kreis und daneben ein kleines Rechteck in einer anderen Farbe. Definieren Sie das Rechteck als Zeichenstift. Rufen Sie nun den Fill Type Requester auf und aktivieren Sie den <Wrap>-Mode. Bestätigen Sie die Eingaben im 'OK'-Feld und aktivieren Sie das <Fill>-Icon. Versuchen Sie nun, den zuvor gezeichneten Kreis zu füllen. Nun können Sie beobachten, wie das Rechteck über den Kreis 'gewickelt' wird, gerade so, als wäre unser Kreis ein rundes Bonbon und das Rechteck das Bonbonpapier, in das der Kreis eingewickelt wird. Natürlich können Sie mit dieser Option jeden nur denkbaren selbstdefinierten Stift benutzen. Allein der Speicherplatz stellt hier die Grenze des Möglichen dar. Der letzte Fülltyp ist <Perspective>. Dieser Fülltyp steht in engem Zusammenhang mit der <Perspective Option>, die Ihnen beim Durchsehen der Kopfzeilen-

menüs sicherlich schon einmal über den Weg gelaufen ist, und die wir an anderer Stelle des Kurses noch genau beleuchten werden. Da die Arbeit mit dem <Perspektive>-Fülltyp einige Grundkenntnisse über die Perspektive an sich erfordert, werden wir sie in Zusammenhang mit dieser Option behandeln. Somit hätten wir alle in dPaint enthaltenen Fülltypen behandelt. Auch <Pattern> und <From Brush> haben wir schon benutzt. Allein <Gradient> und <Dither> fehlen uns noch. dPaint ermöglicht es dem Benutzer, eine Fläche nicht nur mit einer Farbe oder einem Muster zu füllen. Es ist auch möglich, sich aus der aktuellen Palette einige Farben zusammenzustellen und die Fläche mit einem Verlauf aus den ausgewählten Farben zu füllen. Dies ist aber nicht ohne weiteres möglich. Zunächst einmal müssen Sie dPaint mitteilen, aus welchen Farben der Verlauf sich zusammensetzen soll. Dazu ist es unumgänglich, einen kleinen Ausflug in einen anderen Requester zu machen, nämlich den 'Color Requester'. Sie erinnern sich an den Color Indikator? Klicken Sie dort den Kreis, in dem die aktuelle Zeichenfarbe dargestellt wird, mit der rechten Maustaste einmal an. Den Requester, den Sie nun zu sehen bekommen, werden wir an anderer Stelle noch genau erklären. Momentan interessiert uns nur das Feld <Range>. Hier können Sie nun die Farben, aus denen sich der Verlauf zusammensetzen soll, bestimmen. Allerdings mit der Einschränkung, daß es nur Far-

ben sein können, die unmittelbar aufeinander folgen. Die Anzahl der Farben, die zusammengekommen werden sollen, unterliegt keinen Beschränkungen. Es ist also möglich, zwischen zwei und zweiunddreißig Farben jede beliebige Farbenanzahl zu benutzen. Klicken Sie zunächst die linke obere Farbe mit der rechten Maustaste einmal an. Nun das Feld <Range> einmal anklicken. Unter Ihrem Mauszeiger erscheint daraufhin das Wort 'To'. Klicken Sie nun noch einmal die rechte, untere Farbe an. Somit haben wir uns eine sogenannte 'Farbrange', also einen Farbbereich, definiert. Bestätigen Sie die Eingaben mit einem Klick auf 'OK'. Sie rufen nun wieder den 'Fill Type Requester' auf. Sie sehen, daß sich das gedrungene Fenster unter der Bezeichnung <Dither> nun mit einer wilden Farborgie gefüllt hat. Bewegen Sie den <Dither>-Schieberegler ganz nach rechts und lassen Sie ihn dort los. Beobachten Sie einmal, welche Auswirkung diese Aktion auf das kleine Window hat. Die Farborgie sollte jetzt noch wüster, noch wilder gemischt sein. Nun schieben wir den <Dither>-Regler ganz nach links. Voilà, alle Farben in dem kleinen Fenster sind streng gegeneinander abgegrenzt. Die Farben werden beim Verlauf nicht mehr untereinander gemischt. Je weiter Sie den <Dither>-Regler nach rechts schieben, desto stärker werden die Farben auch untereinander gemischt. Kommen wir nun zu dem Punkt <Gradient> und den dahinter positionierten Pfeilen. Diese Pfeile zeigen an, in welcher Richtung die Farben verlaufen sollen. Die erste Pfeilbox steht für senkrechte Verläufe, die zweite für waagrechte. Die dritte Pfeilbox steht für einen 'räumlichen' Verlauf. Wenn Sie z.B. einen Kreis mit Ihrem Verlauf füllen wollen, und der 'räumliche' Fülltyp ist aktiv, paßt sich der Farbverlauf der äußeren Form der zu füllenden Fläche an.

Gerade Linie

Wird dieses Icon mit der rechten Maustaste angeklickt, gelangen wir in den 'Spacing'-Requester. Hier finden Sie vier einstellbare Optionen vor, von welchen die ersten drei durch einen Zahlenparameter ergänzt werden. Fangen wir ganz oben an. Stellen

Sie den <N Total> auf Fünf. Klicken Sie einfach das Zahlenfeld hinter <N Total> an und geben Sie über die Tastatur Fünf ein. Bestätigen Sie auf 'OK' und suchen sich nun einen schönen dicken Zeichenstift aus. Wenn Sie nun die Option <Gerade Linie> benutzen, werden Sie feststellen, daß Linien, die Sie nun zeichnen, keine ununterbrochenen Linien mehr sind. Vielmehr setzen sich die entstehenden Linien, egal wie lang Sie von Ihnen gezeichnet wurden, aus fünf Abbildern des von Ihnen gewählten Pinsels zusammen. Rufen Sie noch einmal den 'Spacing' Requester auf und ändern Sie die Fünf in eine Zehn um. Wenn Sie nun eine Linie ziehen, besteht diese, egal wie lang Sie gezeichnet haben, aus zehn Abbildern des Stiftes. <Every Nth dot> zeitigt einen ähnlichen Effekt. Geben Sie in das betreffende Zahlenfeld eine Fünf ein. Bestätigen Sie wiederum die Eingabe im 'OK'-Feld. Wenn

Sie nun die gerade Linie benutzen, setzt der Amiga Abbilder der gewählten Brush im Abstand von fünf Pixeln (Bildpunkten) hintereinander. Wiederholen Sie den Versuch mit Zehn. Diesmal setzt dPaint die Punkte im Abstand von zehn Bildpunkten. <Airbrush> verbindet die schon beschriebene Airbrush-Funktion, die ja eigentlich nur freihändig benutzt werden kann, mit der Option, gerade Linien zu zeichnen. Der Zahlenparameter hinter dem <Airbrush>-Feld gibt an, wie dicht der Farbnebel sein soll. Am besten spielen Sie ein wenig mit unterschiedlichen Werten herum. Denn auf diesem Wege läßt sich am schnellsten ein Verständnis für die Wirkungsweise dieser Funktion gewinnen. Die letzte Option im 'Spacing' Requester, <Continuous>, stellt die ursprüngliche Funktionsweise der Option <Gerade Linien> wieder her.

Alles was Sie bis hierher gelesen und gelernt haben, stellt

nur einen kleinen Teil der Fähigkeiten von dPaint dar. Alle weiteren Teile dieses Kurses werden auf dem nun gewonnenen Wissen aufbauen.

Wie sicher sind Sie?

Wenn Sie genau wissen wollen, ob Sie bis hierher alles verstanden haben, dann versuchen Sie sich doch einmal an den folgenden kleinen Aufgaben. Natürlich werden wir im nächsten Teil dieses Kurses die Lösungen zu den hier gestellten Aufgaben bringen. In diesem Zusammenhang möchte ich nicht versäumen, Sie auf unsere Databox hinzuweisen. Dort finden Sie weiterführendes Anschauungsmaterial zur Arbeit mit dPaint bzw. mit diesem Kurs.

- 1.) Definieren Sie sich einen Farbbereich (Range) und füllen Sie einen Kreis mit einem räumlichen Verlaufsmuster.
- 2.) Zeichnen Sie mit der Funktion gefülltes Vieleck ein gefülltes Karo.

3.) Definieren Sie sich einen eigenen Zeichenstift. Übernehmen Sie diesen in den Fill Type Requester und füllen Sie damit eine beliebige Fläche.

4.) Zeichnen Sie mit Airbrush zwei gerade Linien mit unterschiedlicher Farbdichte.

5.) Erstellen Sie unter Verwendung der Gitter-Funktion ein Raster. Und zwar so, daß die einzelnen Matrixfelder 10x10 Bildpunkte groß sind, und jeweils nur an den Schnittpunkten der Matrix Bildpunkte gesetzt sind.

Damit sind wir am Ende des ersten Teiles unseres Deluxe-Paint-Kurses angelangt. Ich hoffe, Ihnen so manche Frage beantwortet zu haben und würde mich freuen Sie auch das nächste Mal wieder begrüßen zu dürfen.

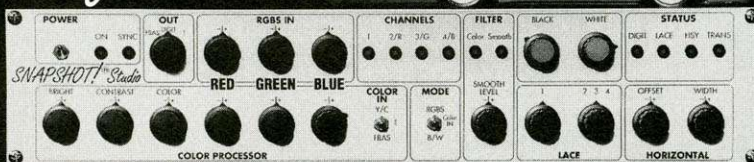


(hs)

Schneller geht's nicht...

Abbildung wurde digitalisiert

1/50
Sekunde



für ein professionelles Bild macht Schluß mit dem minutenlangen Warten.

- S/W-Videobilder in nur 1/50 s digitalisieren!
- Alle AMIGA-Grafikmodi werden unterstützt
- Auflösungen bis zu 704 x 552 Bildpunkten
- 2 bis 46 Grautöne oder 16 bis 4096 Farben
- Bis zu 7 Video-Eingänge, für alle Videoquellen

SNAPSHOT!® PRO (S/W-Echtzeitdigitizer) **895,-**

SNAPSHOT!® RGB (Farb-Adapter für PRO) **395,-**

SNAPSHOT!® Studio (Farb-Komplettgerät) **2375,-**

Kostenlose Informationen anfordern!

... ein deutsches Spitzen-Produkt der **VIDEOTECHNIK DIEZEMANN** Dammstraße 42, 2300 Kiel 1, Telefon (0431) 9 44 24, Telefax (0431) 9 24 32

Einladung...

...ins Software-Paradies!



Software-Katalog!

Fordern Sie unseren kostenlosen Software-Katalog an!

- über 100 Seiten stark
- randvoll mit Programmen und Computerbüchern

Heute noch nebenstehenden Gutschein ausfüllen und einsenden.

Riesenauswahl!

Wählen Sie aus über 5000 Programmen und über 2000 Computerbüchern Ihren Artikel.

In Selbstbedienung, oder mit fachkundiger Beratung, ganz wie Sie wünschen.

SOFTSHOP
Ihr Computer träumt von uns!

4100 Duisburg 1, Sonnenwall 83
Tel. 0203/224 09, Fax: 0203/29756

Gutschein

für einen der umfangreichsten Software-Kataloge Deutschlands!

Kostenlos

Name: _____

Straße: _____

PLZ+Ort: _____

Telefon: _____

NEUE TOP-ANGEBOTE

- ☐ Indiana Jones Adventure dt. 68,-
- ☐ Hillsfar dt. 63,-
- ☐ Space Quest III 89,-

Datum _____ Unterschrift _____

Sie möchten bestellen? Einfach ankreuzen, Absender eintragen, unterschreiben und an untenstehende Adresse schicken. Versand erfolgt per Nachnahme zuzüglich 6,-DM Versandkosten.

(Katalog alleine Versandkostenfrei!)
SOFTSHOP, Sonnenwall 83, 4100 Duisburg 1

Amiga Intern (Band I & II)

Die Intern-Reihe von Data Becker ist schon fast zur Legende geworden. Kaum erscheint ein neuer Computer auf dem Markt, der auch der Öffentlichkeit zugänglich ist, bringt Data Becker kurze Zeit später schon ein Intern-Buch für diesen heraus, in dem alles Wichtige über Hardware und Betriebssystem ausführlich beschrieben ist. Diese Bücher haben sich schon oft als unschätzbare Informationsquellen sowohl für Programmierer als auch für Bastler erwiesen.

Jedesmal, wenn eine wesentliche Neuerung an dem Computer vorgenommen wird, erscheint eine neue Auflage des Intern, in dem auch auf diese Neuerungen eingegangen wird. Seit neuestem erscheinen auch zweite Bände, die für verschiedene Computertypen erhältlich sind, in denen dem Leser die Systemprogrammierung näher gebracht wird. Im Band 1 wird zunächst einmal auf die Hardware des Amiga genauer eingegangen. Nach einer Einleitung erfährt der Leser grundlegende über den Hauptprozessor Motorola 68000, die Custom-Chips Agnus, Denise und Paula sowie die beiden CIAs 8520, die für die Rückmeldung der Übertragung zum seriellen und parallelen Port sowie der Diskettenlaufwerke zuständig sind.

Dabei wird immer zuerst die Pinbelegung und ein Blockschaltbild gezeigt und anschließend ausführlich über diese Chips gesprochen. Viel Beachtung finden auch die Schnittstellen und die Tastatur des Amiga, auf die in ähnlicher Weise eingegangen wird. Der Programmierung der Hardware ist der nächste Abschnitt gewidmet. Nach der Speicherbelegung und den Grundlagen geht es dann gleich richtig ran: Sprites, Blitter, Tonausgabe, Maus, Joystick und Paddles finden hier reichlich Beachtung.

Das zweite große Kapitel des Amiga Intern 1 trägt den Namen EXEC, worin nach Hinweisen zum allgemeinen Programmierstil in C und Assembler auf das Betriebssystem genauer eingegangen wird. Die wichtigsten Bibliotheksfunktionen sind hier ausführlich beschrieben. Im dritten Kapitel wird auf das Amiga-

DOS genauer eingegangen. Unter anderem erfährt man hier Wissenswertes über das CLI, Handler, Devices, Diskettenoperationen, Dateistrukturen und das IFF-Format. Die weiteren Kapitel sind der Expansions-Architektur und dem Januskonzept der PC-Brückenkarte gewidmet.

Der erste Band des Amiga Intern beinhaltet eine Fülle von Informationen, die dem Leser

lung". Gemeint ist damit die Datenübergabe durch CLI-Parameter und das Arbeiten mit den Einträgen im Info-File. Das Kapitel "Ein- und Ausgabe über Devices" enthält Wissenswertes über die Devices (virtuelle Geräte) mit vielen Programmbeispielen. Den Standard-Austausch-Formaten ist ein eigenes Kapitel zugeteilt worden, das zu den IFF-Formaten kaum Fragen offenläßt. Im nächsten Abschnitt werden alle



Hintergrundwissen über die Architektur und die hardwarenahe Programmierung des Amiga bieten.

Der zweite Band von Amiga Intern stammt von den gleichen Autoren, die das große C-Buch geschrieben haben, und richtet sich auch hauptsächlich an die Programmierer dieser Sprache. Aber auch die Assembler-Programmierer kommen hier nicht zu kurz.

Nach der Einleitung erfährt der Leser alles über die Konventionen im Programmstil (strukturierte Programmierung), wie man mit der Multitaskingfähigkeit des Amiga umgeht, Assembler-Richtlinien und "Datenübermitt-

Library-Funktionen beschrieben. Dieses Kapitel ist mit seinen fast 500 Seiten das umfangreichste des Buches. Last but not least wird auf die Basis- und Grundstrukturen im System eingegangen.

(Robert Marz/
Jürgen Seibel/br)

Name: Amiga Intern (Band 1)
Autoren: B. Jennrich, P. Schulz,
W.-G. Bleek
ISBN: 3890111041
(Band 2)
Autoren: J. Schemmel, S. Dittrich,
R. Gelfand
Verlag: DATA Becker
ISBN: 3890112684
Verlag: DATA Becker
Preis: jeweils 69,- DM

Mit C zum Ziel

"C" ist ein Zauberwort, das die Herzen etlicher Programmierer höher schlagen läßt, da man mit dieser Programmiersprache sehr effektiv arbeiten und mit wenig Source-Zeilen jede Menge aus dem Computer herausholen kann. Ein anderer Grund ist die fast widersprüchliche Verbindung von hardwarenaher Programmierung mit hoher Portabilität zwischen verschiedenen Rechnertypen, die durch die genormten Grundlibraries, auf die in C immer wieder zurückgegriffen wird, gegeben ist. So sind die Betriebssysteme des Amiga und des Atari ST sowie das Betriebssystem Unix zum großen Teil in C geschrieben worden. Trotzdem schrecken viele Computerbesitzer vor dieser Sprache zurück. Warum dies so ist, darüber kann man geteilter Meinung sein; vielleicht auch nur deswegen, da doch viele frischgebackene Amigabesitzer mit den 8-Bit-Rechnern angefangen haben und sich mit den Compilersprachen noch nicht auseinandergesetzt haben.

Vielleicht auch deswegen, weil öfter behauptet wird, daß C keine Anfängersprache sei, da die "Versuch-und-Fehler-Methode", wie man sie von Interpretersprachen wie BASIC gewohnt ist, bei C nur schlecht funktionieren würde. Diese Argumente stimmen jedoch nur teilweise, denn man kann C schon mit relativ wenig Programmiererfahrung (z.B. in BASIC oder Pascal) erlernen.

Eine gute Hilfe um einen Einstieg in die Sprache C zu finden, ist das Buch "Mit C zum Ziel" von Arnold Hickersberger aus dem Hüthig-Verlag. In diesem Buch beschreibt der Autor in sehr sachlicher Form den Aufbau der Sprache C.

Das Buch beginnt mit der Entstehungsgeschichte sowie einer Auflistung der in C enthaltenen und vor allen den nicht enthaltenen Sprachkomponenten, behandelt die allgemeine Schreibweise und gibt einen Überblick über die Programmstruktur von C. Des weiteren werden sämtliche Unklarheiten über Variablen und ihre Datentypen sowie die vom Compiler durchgeführten Konvertierungen bei Operationen von Variablen unterschiedlichen Typs beseitigt.

Auch die nächsten Kapitel beschäftigen sich mit der Theorie: Hier werden Operatoren und deren Prioritäten, Blöcke, bedingte bzw. direkte Sprünge und Schleifenprogrammierung angesprochen.

Der Abschnitt sechs wird von der Praxis beherrscht. Das bis hierhin Erlernte wird anhand von ausführlich erläuterten Beispielen geübt.

Weiter geht es mit der Erklärung und Einbindung von Speicherklassen, Zeigern, Adressen und Feldern, aber auch Funktionen, die eine genau definierte Aufgabe (wie Datenein- und -ausgabe, Berechnung von Werten etc.) zu erfüllen haben, kommen in diesem Buch nicht zu kurz.

Die nächsten Abschnitte beschäftigen sich mit Strukturen, Varianten, Bitfeldern und Aufzählungen, während dem Preprozessor das Kapitel dreizehn gewidmet ist. Den Abschluß bilden die Grundzüge der Dateibearbeitung, Portabilität der Sourcecodes, wobei sowohl auf compiler-, als auch auf hardware- und betriebsystembedingte Abhängigkeiten eingegangen wird und der neue ANSI-C-Standard, dem ein eigenes Kapitel gewidmet ist.

Der Inhalt des Buches ist wegen des verbreiteten Standards der Sprache C sehr allgemein gehalten und auf jedes Betriebssystem und jeden Computertyp, für den die Sprache C erhältlich ist, übertragbar. Auf eventuelle Unterschiede von Betriebssystemen und bei Compilern wird gesondert hingewiesen. In diesen Fällen hilft aber oft ein Blick in das Compiler-Handbuch weiter. Nach dem Durcharbeiten dieses Buches verfügt der Leser über ein fundiertes Grundwissen der Sprache und Sprachstruktur und sollte ohne weiteres in der Lage sein, auch komplizierte Projekte zu verwirklichen. Allerdings wendet sich das Buch nicht an absolute Computereinsteiger, da das Buch sehr theoretisch beginnt und sehr Kenntnisse in einer anderen Sprache wie Pascal, BASIC oder Assembler voraussetzt.

In einem Satz: Ein optimaler C-Kurs, der jedem, der sich in die Sprache C einarbeiten will, ans Herz gelegt werden sollte.

(Robert Marz/
Jürgen Seibel/br)

Name: Mit C zum Ziel, das C-Buch
für Ein- und Umsteiger
Autor: A. Hickersberger
Verlag: Dr. Alfred Hüthig Verlag
Heidelberg.
ISBN: 3295984
Preis: 38,- DM

Das Programmierbuch

Das Buch ist eine Übersetzung der amerikanischen Ausgabe "Programmer's Guide to the Amiga" und liegt in der zweiten Auflage vor. Der Inhalt des Buches und die Kombination aus den Büchern "Amiga Rom Kernel Manual", dem "AmigaDOS Developers Manual" und dem "Intuition Manual" geben dem Leser einen umfassenden Überblick über das Software-System des Amiga.

Im ersten Kapitel findet der Programmierer eine Kurzbeschreibung der Software- und Hardware-Hierarchie des Amiga. Es folgen einige Anmerkungen zum Gebrauch der Funktionsbibliotheken der Programmierung in C und Assembler sowie den Include-Dateien der Compiler. Das zweite Kapitel beschäftigt sich mit der Programmierung und Nutzung von AmigaDOS und den Funktionen des CLI. Hier werden beispielsweise das Öffnen von Fenstern für Ein- und Ausgabeoperationen, Parameterübergabe beim Programmaufruf oder Schreib- und Lesezugriffe auf Dateien ausführlich mit ablauffähigen Programmen erläutert.

Das dritte Kapitel widmet sich Exec und den Informationen, die zur Arbeit mit Exec nötig sind. Es werden jedoch nicht alle Funktionen von Exec zur Verfügung gestellt, da dies den Rahmen des Buches gesprengt hätte. So werden Tasks und Prozesse, die Speicherreservierung (Chip/Fast), Messages, Bibliotheken, und virtuelle Geräte (z.B. console.device und trackdisk.device) ausführlich erläutert.

Im vierten Kapitel, das mit Grafik betitelt ist, lernt man Grafikbereiche zu selektieren und initialisieren, wie Farben erzeugt, Bereiche mit diesen gefüllt und wie man Objekte wie z.B. Linien, Kreise oder Rechtecke zeichnet. Innerhalb dieses Kapitels werden ein kleines Malprogramm sowie ein Programm zur Darstellung von Balkengrafiken erstellt.

Die weiteren Kapitel des Buches setzen sich dann sehr intensiv mit der Benutzeroberfläche Intuition, den Devices, der Animation, dem Sound und der Multitaskingfähigkeit des Amiga auseinander. Dies erwies sich deshalb als nötig, weil in den vorangegangenen Kapiteln teilweise nur sehr kurz auf den entsprechenden Sachverhalt eingegangen wurde.

Den Anhang bilden zwei Kapitel, die besonders für den weniger versierten Anwender von Nutzen sind. Im ersten Kapitel wird der Texteditor ED ausführlich mit allen Befehlen erläutert. Den Abschluß bildet eine kleine Anleitung für den Amiga C-Compiler und einer kleinen Programmsammlung die das Arbeiten mit dem C-Compiler erleichtert. Abschließend läßt sich feststellen, daß das Buch einen Platz in der Buchsammlung verdient hat. Besonders zu empfehlen ist das Buch all denjenigen, die schon in C programmieren.

(Klaus Rupp/br)

Name: Das Programmierbuch
Autor: Robert A. Peck
Verlag: Sybex
ISBN N 3-88745-520-7
Preis: 49,- DM

Das Amiga Guru-Buch

Das Buch wendet sich an den fortgeschrittenen Programmierer, insbesondere an C-Programmierer. Assembler-Freaks können aber hier und dort dem Buch auch noch Wissenswertes entnehmen. Besonders hervorzuheben ist, daß der gesamte Inhalt des Buches auf das Betriebssystem Version 1.3 ausgerichtet ist.

Unterteilt ist das Buch in 22 Kapitel, die sich wiederum in die drei Hauptgebiete Programmierung, Systemintern, und AmigaDOS gliedern. Abschnitt eins behandelt die verschiedenen Datentypen der MC68xxx Prozessoren. Es folgen Richtlinien für die Programmierung in C (Lattice C-5.0 u. Aztec-C-3.6a) und Assembler. Auch der ins System integrierte Debugger WACK wird kurz erläutert.

Abschnitt zwei behandelt die Speicherbelegung, Kickstart, Boot-Vektoren und als Leckerbissen die Alarmzustände, besser bekannt als Alerts und Gurus des Amiga.

Anhand der Guronummer sollte es nun jedem möglich sein, die Ursache einer solchen Meldung zu lokalisieren.

Der dritte Abschnitt beschäftigt sich sehr ausführlich mit dem CLI, dem Console-Handler, Datenstrukturen und Fehlercodes des AmigaDOS. Alle Kapitel werde durch eine große Anzahl von C-Programmen ausführlich unterstützt.

Am Ende des Buches findet der Leser in einem Glossar Fachbegriffe in einfacher Form erläutert. Insgesamt ist das Amiga Guru-Buch ein echtes Nachschlagewerk für den erfahrenen Programmierer. Mit einem Preis von 48,- DM empfiehlt sich das Buch nicht nur vom günstigen Preis her als Alternative zu diversen anderen Amiga-Büchern.

(Klaus Rupp/br)

Name: Das Amiga Guru-Buch
Autor und Vertrieb: Ralph Babel
Falkenweg 3, 6204 Taunusstein
Preis: 48,- DM



Martin Schlöter

Texterstellung für jedermann

Die DMV-Textverarbeitung 'ConText' jetzt auch auf dem Amiga

Schon bekannt vom Amstrad CPC erfreut sich die Portierung der Textverarbeitung ConText auf dem PC ungemeiner Beliebtheit. Es wurde allerhöchste Zeit, daß eine Version für den Amiga entstand, die auch die Philosophie von ConText konsequent weiterführt – WYSIWYG-Darstellung reinsten Wassers, die Bildschirmdarstellung muß maximale Ähnlichkeit mit dem Druckbild haben.

Um ConText-Amiga einem maximalen Benutzerkreis zur Verfügung zu stellen, kam als Sprache, in der das Programm geschrieben werden sollte, eigentlich nur BASIC in Frage. Mit C oder Modula-2 arbeiten noch (?) relativ wenig Amiga-Anwender und eine Textverarbeitung in Assembler zu codieren, ist eine Arbeit für jemanden, der Vater und Mutter erschlagen hat (Anmerkung des zuständigen Redakteurs: Bei Modula2 reichen schon Bruder und Schwester).

Die Probleme, die sich mit dem, beim Gerät mitgelieferten, AmigaBASIC ergeben, sind wohl hinlänglich bekannt: Ohnehin schon der langsamste mir bekannte BASIC-Dialekt ist zu allem Überfluß dort auch die Textausga-

be extrem langsam. Probeläufe selbst mit compilierten Programmversionen ergaben ein unbefriedigendes Geschwindigkeitsverhalten, zumal der BASIC-Editor bei Programmen, die mehr als 100 Zeilen lang sind, Programmierer zu Vendetta-(Familienfehde-)Gelüsten hinreißen kann.

GFA – Die Rettung eines in Zeitnot stehenden Programmierers naht!

GFA sei Dank, stand rechtzeitig zum Projektstart das GFA-BASIC für den Amiga zur Verfügung und die ersten Beta-Versionen des GFA-BASIC-Compilers zeigten sich als Silberstreif am Horizont (siehe auch den Testbericht in dieser Ausgabe). Erste Test-

läufe zeigten, daß mit der interpretierten Version ein erträgliches (soll heißen: schneller als der Amiga-BASIC-Editor) und mit compilierten Programmversionen ein zufriedenstellendes Zeitverhalten zu erreichen war, ohne auf Assemblersequenzen zurückgreifen zu müssen.

Das Pflichtenheft – was soll eine Amiga-Textverarbeitung können?

Die Anforderungen an ConText-Amiga waren von vorneherein recht klar:

- WYSIWYG-Darstellung ('What You See Is What You Get') ist Pflicht, also Schriftattribut wie fett, unterstrichen und kursiv sollen so auf dem Bildschirm dargestellt werden, wie sie hinterher auf dem Drucker erscheinen.
- ConText-Amiga soll bis zu 80 Zeichen breite Texte bearbeiten können, mit frei wählbarem linken und rechten Rand.
- Die Anzahl der Zeilen pro Seite soll frei wählbar sein.
- Zentriert, linksbündiger, rechtsbündiger Text und Blocksatz sollen möglich sein.
- Abspeicherung in einem ConText-Format, einem Layout-Format (mit Randeinstellungen und Zeilen pro Seite) und natürlich ASCII.
- Nachträgliches Formatieren von Absätzen
- Suchen und Ersetzen von Zeichenfolgen

- Block-Operationen
- Amiga-typische Bedienung mit Maus und Menüs, soweit in einer Textverarbeitung sinnvoll.
- Tastenbedienung nach Wordstar-Prinzip, mit Ausnutzung von Cursor- und sonstigen Sonder-Tasten
- Drucken, während Text bearbeitet wird
- Konfigurierbar für beliebige Drucker

Das Nadelöhr und die Brechstange für das Kamel

Wer die Programmlogik analysiert hat, wird wahrscheinlich auch schnell durchschaut haben, wo die zeitkritischen Pausen des Programms sitzen: Viel Zeit wird natürlich in der Prozedur "showline" verbraten, was darauf zurückzuführen ist, daß innerhalb des Attributstring jedes einzelne Zeichen untersucht werden muß, um eine Änderung des aktuellen Schriftattribut festzustellen. Bei einem 80-Zeichen langen String muß dazu 80-mal die MID\$-Funktion aufgerufen werden, mit dem ganzen Rattenschwanz, der dadurch ausgelöst wird:

Ermitteln der Adresse der String-Variablen innerhalb der Interpreter-Variablen-tabelle, aus der String-Variablen (dem "Deskriptor"), die Adresse des Zeichen-Arrays lesen, dort mit dem passenden Offset (der Zeichennummer) auf den Speicher zugreifen).

Dazu kommen noch die Fallunterscheidungen, ob es sich um ein Leerzeichen oder um

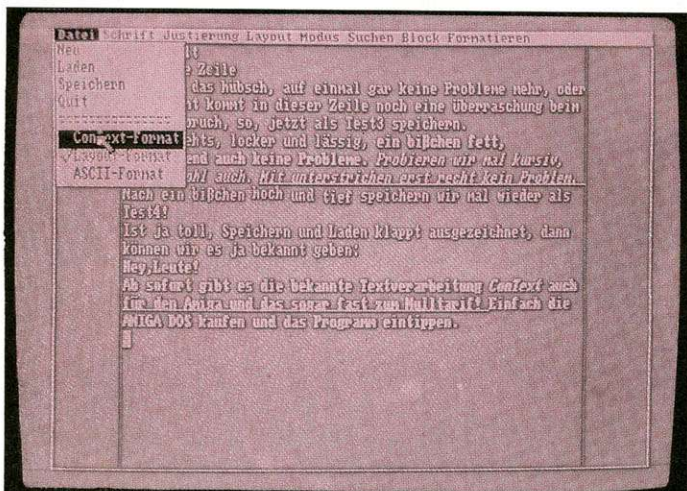


Bild 1: ConText- Amiga zeigt seine Funktionen in einer Menueleiste.

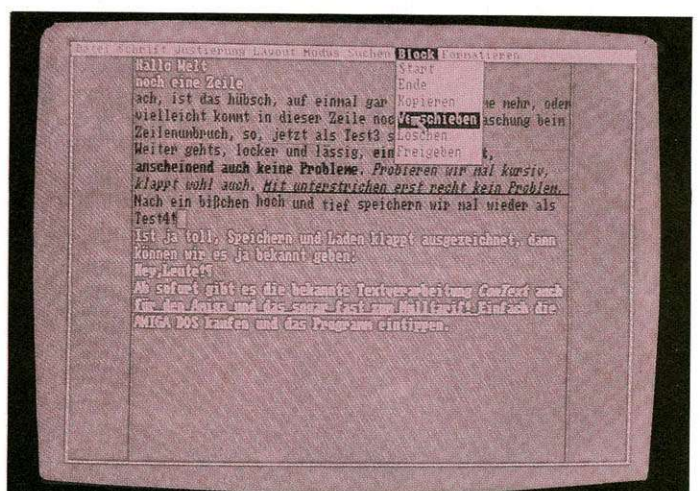


Bild 2: Blöcke werden durch eine andere Zeichenfarbe gekennzeichnet.

ein normales Zeichen handelt. Daß die Prozedur rekursiv arbeitet, um die Zeile in Abschnitte mit gleichen Attributen zu zerlegen, spielt hingegen keine große Rolle, trotz anderslautender Gerüchte.

Auch die Prozedur "stripline", deren Aufgabe es ist, am Ende einer fertig bearbeiteten Zeile überflüssige Füllzeichen zu entfernen, beziehungsweise die Füllzeichen am Anfang einer Zeile in echte Leerzeichen umzuwandeln, muß extensiv auf einzelne Zeichen innerhalb von Strings zugreifen.

Der "Hebel" mit dem man dieses Nadelöhr passierbar kriegt, ist die Anwendung einer "schlankeren" Programmiersprache. Lange Rede kurzer Sinn ist BASIC mit seiner mächtigen, aber auch Overhead-belasteten Stringverarbeitung nicht besonders geeignet, einzelne Zeichen zu bearbeiten (siehe oben dreifach indirekte Adressierung). Der naheliegende Ansatz, statt Strings mit Arrays von Bytes zu arbeiten, ist im Ansatz durchaus vernünftig (kann etwa 30% bringen), macht einen aber immer noch von den etwas ineffizienten Speicherzugriffsmechanismen von BASIC abhängig. Um wirklich Speed in die Geschichte zu bringen, bietet es sich an, die beiden erwähnten Prozeduren in Assembler zu kodieren, beziehungsweise Hilfsprozeduren in Assembler zu erstellen, die nur noch die Adresse und die Länge des Strings übergeben bekommen und diesen Byte für Byte "abgrasen".

Hier sind auch Sie gefragt, lieber Leser: Wir werden in einer der nächsten Ausgaben die schönsten Verbesserungen für ConText-Amiga prämiieren, und auch den anderen Lesern zur Verfügung stellen - also, Assembler-Programmierer an die Front.

Die Bedienungsanleitung zu ConText - Auch Schreiben will gelernt sein

Die **Cursortasten** haben Ihre Funktion behalten: Up-Down-Right-Left. Alternativ besteht ein weiteres Cursor-Kreuz in mit den Tasten 'S' (Links), 'E' (Oben), 'D' (Rechts) und 'X' (Unten). <BackSpace> (<- DEL) hat ebenfalls die übliche Wirkung, das links neben dem

Cursor stehende Zeichen zu löschen.

Kommen wir jetzt zu den CTRL- und SHIFT-Funktionen. Hier rufen die Control-(CTRL-) Taste und die SHIFT-Tasten zusammen mit anderen Tasten eine bestimmte Funktion auf. Schauen wir dies uns näher an:

<CTRL> + <N>

Durch gleichzeitiges Drücken von CTRL und der Taste 'N' wird eine Zeile in den bestehenden Text eingefügt. Diese Funktion ist nicht in der Menue-Leiste enthalten.

<CTRL> + <Y>

Diese Kombination bildet das Gegenteil: Es wird die aktuelle Zeile (in der der Cursor steht) komplett gelöscht. Diese Funktion ist ebenfalls nicht in der Menueleiste zu finden!

<SHIFT> + <Pfeil auf>

Hier haben Sie die Möglichkeit, mit dem Cursor eine Seite nach oben (Page up) zu gehen. Das Gegenteil ist

<SHIFT> + <Pfeil ab>

Hier wird eine Seite nach unten (Page down) geblättert. Genaue gesagt bildet <SHIFT> + <Pfeil auf> das Rückwärtsblättern, während <SHIFT> + <Pfeil ab> für das Vorwärtsblättern steht.

<SHIFT> + <Pfeil rechts>

Mit der SHIFT- und der '<' Taste bringt man den Cursor an den Anfang einer Zeile, während <SHIFT> + <Pfeil links> den Cursor an das Ende der aktuellen Zeile bringt.

<CTRL> +

Wenn Sie Ihren Text formatieren wollen, müssen Sie diese Tastenkombination drücken. Danach wird der Text bis zum nächsten Absatzende, das durch ein sogenanntes Hard-Return gekennzeichnet ist, in die gewünschte Form gebracht.

Als nächstes wenden wir uns den Block-Befehlen zu. Sie werden durch gleichzeitiges Drücken der Tasten 'CTRL' und 'K' und einer nachfolgenden Kenntaste erreicht:

<CTRL> + <K> markiert den Blockanfang,

<CTRL> + <K> <K> wiederum markiert das Blockende

Mit <CTRL> + <K> <Y> wird der vorher markierte Block gelöscht, mit <CTRL>

+ <K> <C> wird der markierte Block kopiert. Dabei bleibt der markierte Teil an seinem Platz erhalten. Dies ist jedoch nicht so bei <CTRL>

+ <K> <V>, der Funktion 'Block verschieben'. Hier wird

der markierte Block an die Cursorposition gesetzt und an der ursprünglichen gelöscht.

<CTRL> + <K> <H> dient dazu, die durch Setzen des Blockanfangs und -endes angezeigte Markierung zu entfernen. Die Blöcke sind zeilenorientiert, das heißt, daß nur komplette Zeilen in den Block übernommen werden können. Blöcke können nicht in sich selbst kopiert oder verschoben werden.

Nun jedoch weiter mit den Funktionen:

<CTRL> + <V> Mit dieser Tastenkombination wird zwischen dem Insert- und dem Overwrite-Modus umgeschaltet. Im Insert-Modus werden Zeichen zwischen bestehende Wörter eingesetzt, im Overwrite-Modus werden die Zeichen auf dem Bildschirm bei Neueingabe überschrieben.

Nun noch einige Bemerkungen zu den Menue-Funktionen:

1.) Der wichtigste Menuepunkt wird wohl das 'Datei'-Menue sein. Hier gibt es zuerst die Punkte NEU (hier wird nach nochmaliger Nachfrage der vorhandene Text gelöscht, um einem neuen Text Platz zu machen), weiter sind hier die Funktionen 'LADEN' und 'SPEICHERN' für Textdateien vorhanden, sowie 'QUIT', dessen Sinn sich wohl selbst erklärt.

Dateien können in drei verschiedenen Formaten abgespeichert werden:

○ 'CONTEXT-FORMAT' - Hier werden alle Schriftattribute mit abgespeichert, wie Kursiv-, Fett-, oder Hoch-/Tief-Schrift.

○ 'LAYOUT-FORMAT' - Zusätzlich zu den Attributen wird auch das Seitenformat mit Rand- und Seiteneinstellungen abgespeichert.

○ 'ASCII-FORMAT' - Hier wird nur der reine Text in ASCII-Form ohne Steuerzeichen abgespeichert. Man kann also mit ConText-Amiga auch die Startup-Sequence oder Execute-Dateien erstellen.

2.) Der Menuepunkt 'Schrift' enthält die Schriftarten Normal, Fett, Unterstrichen, Kursiv, Hoch und Tief.

3.) Der Menuepunkt 'Justierung' existiert zur Zeit noch nicht, hier werden wir später noch weitere Layout-Funktionen liefern.

4.) Im Menuepunkt 'Layout' läßt sich der linke sowie der rechte Rand eintragen, sowie

die Anzahl der Zeilen pro Bildschirmseite.

5.) Der Punkt 'Modus' entspricht der Tastenkombination CTRL + V, hier kann zwischen 'INSERT' und 'Overwrite' gewählt werden.

6.) Beim Menuepunkt 'Suchen' gibt es die Funktionen 'Suchen', 'Ersetzen' und 'Weitersuchen'. Bei 'Suchen' wird nur der eingegebene Begriff gesucht, der Cursor bleibt an der betreffenden Stelle stehen. Mit 'Weitersuchen' kann die Suchfunktion fortgeführt werden. Bei 'Ersetzen' wird zuerst das gesuchte und danach das Ersatzwort eingegeben, nach dem Finden werden Such- und Ersatzbegriff ausgetauscht. Auch hier kann mit 'Weitersuchen' weitergemacht werden.

7.) Im Menuepunkt 'Block' sind alle Blockoperationen noch einmal mit der Maus erreichbar.

8.) 'Formatieren' bringt den Text in die gewünschte Form bis zum nächsten Absatz.

Bei Ausgabe auf den Drucker müssen wir jetzt noch auf unsere Preferences-Druckertreiber oder auf separate Druckprogramme zurückgreifen, jedoch wird in der nächsten Ausgabe ein separates, Multitasking-fähiges Druckerprogramm erscheinen.

Dieses wird dann durch die Tastenkombination <AMIGA> + <N> vor das Hauptprogramm gestellt, bekommt seine auszudruckende Datei, und druckt danach im Hintergrund aus, während man vorne weiter den Text bearbeiten kann. Der Zeilenabstand in ConText-Amiga ist nicht Amiga-kompatibel, zwei Pixel werden freigelassen für die Hoch- und Tief-schrift. Die Ausgabe geschieht dabei komplett im Grafikmodus.



Wichtig!!!

Beim BASIC-Programm sollten die Zeilennummern nicht mit eingetippt werden, da diese nur die Übersicht im Listing verbessern. Sie können zur Eingabe auch einen Texteditor und im Anschluß daran unseren Checksummer benutzen.

Listings

```

1: ' *****
2: ' ***
3: ' ConText-Amiga Version 1.0
4: '
5: ' Die WYSIWYG-Textverarbeitung fuer den Amiga
6: '
7: '
8: ' (C) 1989,1990 Amiga-DOS & M.Schloeter
9: '
10: ' *****
11: '
12: ' ***** Voreinstellungen *****
13: $&3
14: $&
15: CLEAR
16: RESERVE 200000
17: DEFWRD "A-Z"
18: ' ***** Konstanten *****
19: progtitel$="ConText-Amiga"
20: winheight&=252
21: maxlines&=1000 ! 1000 Zeilen Text maximal
22: linespp&=24
23: lineoffs&=10
24: lrandoffs&=2
25: parac$=CHR$(182)
26: ' --- Attribut-Bits ---
27: boldb!=2
28: italicb!=4
29: underlineb!=1
30: subscriptb!=8
31: superscriptb!=16
32: ' --- Tastaturcodes ---
33: pfeillr$=CHR$(155)+CHR$(67)
34: pfeilll$=CHR$(155)+CHR$(68)
35: pfeillu$=CHR$(155)+CHR$(65)
36: pfeilld$=CHR$(155)+CHR$(66)
37: shpfeillu$=CHR$(155)+CHR$(84)
38: shpfeilld$=CHR$(155)+CHR$(83)
39: shpfeillr$=CHR$(155)+" "+CHR$(64)
40: shpfeilll$=CHR$(155)+" "+CHR$(65)
41: instoggle$=CHR$(22)
42: ' ***** Variablen *****
43: DIM text$(maxlines&)
44: DIM attr$(maxlines&)
45: DIM menu$(100)
46: workfile$=""
47: bold!=FALSE
48: italic!=FALSE
49: underline!=FALSE
50: dwidth!=FALSE
51: subscript!=FALSE
52: superscript!=FALSE
53: somethinghappend!=FALSE
54: lrand&=10
55: rrand&=70
56: zeilenzahl!=66
57: scrx&=lrand&
58: scry&=1
59: texty&=1
60: textx&=1
61: lastline&=0
62: scriptoffset&=0
63: attribut!=0
64: ascii!=FALSE
65: contextformat!=TRUE
66: mitlayout!=FALSE
67: insertmode!=FALSE
68: cursormovement!=FALSE
69: wrapped!=FALSE
70: header$=""
71: lastsearchx&=1 ! Die letzten Suchpositionen
72: lastsearchy&=1
73: lasteqsearch!=TRUE ! Letzte Operation = Suchen
74: such$="" ! Such-String
75: ersetz$="" ! Ersetzungsstring
76: blockstart&=0
77: blockende&=0
78: '
79: ' HIRES-Screen oeffnen
80: OPENS 1,0,0,640,winheight&,2,32786
81: OPENS 1
82: '
83: ' WINDOW oeffnen
84: OPENW #&H0,0,0,640,winheight&,0,0,1
85: headline
86: FRONTW #&H0
87: rastport$=FN getrastport(0)
88: GRAPHMODE 1
89: '
90: ON BREAK GOSUB ctrlc
91: ON ERROR GOSUB generor
92: ' Menues initialisieren
93: initmenu(menu$())
94: MENU menu$()
95: MENU 6,256 ! ConText-Format default
96: MENU 11,256 ! Text Normal default

```

```

94: MENU 19,256 ! Linksbuendig default
95: MENU 20,192 ! rechtsbuendig disabeln
96: MENU 21,192 ! zentriert disabeln
97: MENU 22,192 ! Blocksatz disabeln
98: MENU 31,256 ! Overwrite default
99: ON MENU GOSUB menureact
100: '
101: '
102: '
103: COLOR 1,0
104: justamoment
105: drawborder
106: CLOSEW #&H1
107: OPENW #&H0
108: stretchline
109: cursoron
110: DO
111: SLEEP
112: keytrap
113: mousetrap
114: LOOP
115: GOSUB closeall
116: END
117: '
118: '
119: > PROCEDURE mousetrap
120: LOCAL lasty&,maux&,mausy&,mausk
121: mymouse(maux&,mausy&,mausk)
122: IF mausk=1
123: cursoroff
124: lasty&=scry&
125: scrx&=maux&
126: scry&=mausy&
127: updatecurspos
128: stripline
129: texty&=texty&+scry&-lasty&
130: IF LEN(text$(texty&))>0 THEN
131: stretchline
132: findtextx
133: findscrx
134: IF scrx&>lrand& THEN
135: IF NOT (MID$(text$(texty&),textx&,1)=parac$) T
HEN
136: DEC scrx&
137: DEC text&
138: ENDIF
139: ENDIF
140: ELSE
141: stretchline
142: textx&=scrx&-lrand&+1
143: ENDIF
144: cursoron
145: ENDIF
146: RETURN
147: > PROCEDURE keytrap
148: LOCAL c$
149: c$=INKEY$
150: IF c$<>"" THEN
151: cursoroff
152: IF NOT somethinghappend! THEN
153: somethinghappend!=TRUE
154: ENDIF
155: IF LEN(c$)<>1 THEN
156: GOSUB extendedkey(c$)
157: ELSE IF c$<CHR$(32)
158: GOSUB ctrlcode(c$)
159: ELSE IF c$=" "
160: IF cursormovement! THEN
161: cursormovement!=FALSE
162: stripline
163: IF (textx&>LEN(text$(texty&))) AND (LEN(text$(
texty&))>0) THEN
164: findrpos
165: ENDIF
166: stretchline
167: ENDIF
168: checkwrap(c$)
169: IF NOT wrapped! THEN
170: GOSUB blank
171: ENDIF
172: wrapped!=FALSE
173: ELSE
174: IF cursormovement! THEN
175: cursormovement!=FALSE
176: stripline
177: IF (textx&>LEN(text$(texty&))) AND (LEN(text$(
testy&))>0) THEN
178: findrpos
179: ENDIF
180: stretchline
181: ENDIF
182: checkwrap(c$)
183: insertchar(c$)
184: IF NOT insertmode! THEN
185: printchar(c$)
186: ELSE
187: showline(lrand&,scry&,text$(texty&),attr$(text
y&),(texty& MOD zeilenzahl&)=0)
188: drawborder
189: INC scrx&
190: ENDIF
191: ENDIF

```



```

192: cursoron
193: ENDIF
194: RETURN
195: > PROCEDURE blank
196: IF NOT insertmode! THEN
197: MID$(text$(texty&),textx&,1)=" "
198: MID$(attr$(texty&),textx&,1)=CHR$(1)
199: ELSE
200: text$(texty&)=LEFT$(text$(texty&),textx&-1)+" "+MI
D$(text$(texty&),textx&)
201: attr$(texty&)=LEFT$(attr$(texty&),textx&-1)+CHR$(1
)+MID$(attr$(texty&),textx&)
202: ENDIF
203: INC textx&
204: IF NOT insertmode! THEN
205: printchar(" ")
206: ELSE
207: showline(lrand&,scry&,text$(texty&),attr$(texty&),
(texty& MOD zeilenzahl&)=0)
208: drawborder
209: INC scrx&
210: ENDIF
211: RETURN
212: > PROCEDURE extendedkey(c$)
213: cursmovement!=TRUE
214: IF c$=pfeilu$ THEN
215: DEC scry&
216: stripline
217: DEC texty&
218: stretchline
219: IF texty&<1 THEN
220: stripline
221: texty&=1
222: stretchline
223: scry&=1
224: ENDIF
225: IF FN rechtsraus THEN
226: findrpos
227: ENDIF
228: updatecurpos
229: ELSE IF c$=pfeild$
230: INC scry&
231: stripline
232: INC texty&
233: stretchline
234: IF texty&>maxlines& THEN
235: stripline
236: texty&=maxlines&
237: stretchline
238: DEC scry&
239: ENDIF
240: IF FN rechtsraus THEN
241: findrpos
242: ENDIF
243: updatecurpos
244: ELSE IF c$=pfeill$
245: INC scrx&
246: INC textx&
247: updatecurpos
248: ELSE IF c$=pfeill$
249: DEC scrx&
250: DEC textx&
251: updatecurpos
252: ELSE IF c$=shpfeild$
253: pagedown
254: ELSE IF c$=shpfeilu$
255: pageup
256: ELSE IF c$=shpfeill$
257: endlne
258: ELSE IF c$=shpfeill$
259: posl
260: ENDIF
261: RETURN
262: > PROCEDURE ctrlcode(c$)
263: SELECT ASC(c$)
264: CASE 2 ! Ctrl-B
265: GOSUB absatzformatieren
266: CASE 4 ! Ctrl-D
267: GOSUB extendedkey(pfeill$)
268: CASE 5 ! Ctrl-E
269: GOSUB extendedkey(pfeilu$)
270: CASE 24 ! Ctrl-X
271: GOSUB extendedkey(pfeild$)
272: CASE 19 ! Ctrl-S
273: GOSUB extendedkey(pfeill$)
274: CASE 13 ! <Return>
275: GOSUB paragraph
276: CASE 8 ! BackSpace
277: GOSUB backspace
278: CASE 11 ! Ctrl-K
279: REPEAT
280: c$=INKEY$
281: UNTIL c$<>" "
282: GOSUB ctrlk(c$)
283: CASE 22 ! Ctrl-V = Insert Ein/Aus
284: GOSUB toggleinsert
285: CASE 25 ! Ctrl-Y = Delete Line
286: GOSUB delline(texty&)
287: CASE 14 ! Ctrl-N = Insert Line
288: GOSUB insline(texty&)
289: ENDSELECT
290: RETURN

```

```

291: > PROCEDURE backspace
292: DEC textx&
293: IF textx&>0 THEN
294: MID$(text$(texty&),textx&,1)=" "
295: MID$(attr$(texty&),textx&,1)=CHR$(0)
296: showline(lrand&,scry&,text$(texty&),attr$(texty&),
((texty& MOD zeilenzahl&)=0))
297: drawborder
298: DEC scrx&
299: ELSE
300: IF texty&>1 THEN
301: DEC scry&
302: DEC texty&
303: findrpos
304: IF textx&>1 THEN
305: DEC scrx&
306: ENDIF
307: MID$(text$(texty&),textx&,1)=" "
308: MID$(attr$(texty&),textx&,1)=CHR$(0)
309: stripline
310: showline(lrand&,scry&,text$(texty&),attr$(texty&),
(texty& MOD zeilenzahl&)=0)
311: stretchline
312: drawborder
313: ELSE
314: textx&=1
315: ENDIF
316: ENDIF
317: RETURN
318: > PROCEDURE paragraph
319: stripline
320: IF textx&<LEN(text$(texty&)) THEN
321: INSERT text$(texty&+1)=MID$(text$(texty&),textx&)
322: INSERT attr$(texty&+1)=MID$(attr$(texty&),textx&)
323: scrolldownarea(texty&+1,linespp&,1)
324: ENDIF
325: text$(texty&)=LEFT$(text$(texty&),textx&-1)+parac$
326: attr$(texty&)=LEFT$(attr$(texty&),textx&-1)+CHR$(0)
327: INC texty&
328: IF texty&>maxlines& THEN
329: texty&=maxlines&
330: ELSE
331: INC scry&
332: ENDIF
333: scrx&=lrnd&
334: textx&=1
335: updatecurpos
336: DEC texty&
337: stretchline
338: showline(scrx&,scry&-1,text$(texty&),attr$(texty&),
(texty& MOD zeilenzahl&)=0)
339: stripline
340: INC texty&
341: stretchline
342: showline(scrx&,scry&,text$(texty&),attr$(texty&),
(texty& MOD zeilenzahl&)=0)
343: drawborder
344: RETURN
345: > PROCEDURE printchar(c$)
346: LOCAL style&
347: style&=AND(attribut!,underline!+boldb!+italicb!)
348: GOSUB setstyle(rastport%,style&)
349: TEXT (scrx&-1)*8+lrndoffs&,scry&*lineoffs&+8+script
offset&,c$
350: INC scrx&
351: GOSUB updatecurpos
352: RETURN
353: > PROCEDURE insertchar(c$)
354: IF insertmode! THEN
355: text$(texty&)=LEFT$(text$(texty&),textx&-1)+c$+MID
$(text$(texty&),textx&)
356: attr$(texty&)=LEFT$(attr$(texty&),textx&-1)+CHR$(a
ttribut!)+MID$(attr$(texty&),textx&)
357: ELSE
358: MID$(text$(texty&),textx&,1)=c$
359: MID$(attr$(texty&),textx&,1)=CHR$(attribut!)
360: ENDIF
361: INC textx&
362: RETURN
363: > PROCEDURE menureact
364: LOCAL mindex&,menstr$,res&,filename$
365: cursoroff! <---- Cursor aus
366: mindex&=MENU(0)
367: menstr$=menu$(mindex&)
368: IF menstr$="Neu" THEN
369: res&=1
370: IF somethinghappend! THEN
371: ALERT 0,"Alles loeschen?",2,"Na klar ! ; Bloß n
icht !",res&
372: ENDIF
373: IF res&=1 THEN
374: GOSUB closeall
375: RUN
376: ENDIF
377: ELSE IF menstr$="Quit"
378: res&=1
379: IF somethinghappend! THEN
380: ALERT 0,"Programme beenden?",2,"Na klar ! ; Blo
ß nicht !",res&
381: ENDIF
382: IF res&=1 THEN
383: GOSUB closeallend

```



```

384:     ENDIF
385:     ELSE IF menstr$="Laden"
386:         GOSUB textladen
387:     ELSE IF menstr$="Speichern"
388:         GOSUB textspeichern
389:     ELSE IF menstr$=" Context-Format"
390:         IF NOT contextformat! THEN
391:             somethinghappend!=TRUE
392:             contextformat!=TRUE
393:             mitlayout!=FALSE
394:             ascii!=FALSE
395:             MENU 6,256
396:             MENU 7,16+64
397:             MENU 8,16+64
398:         ENDIF
399:     ELSE IF menstr$=" Layout-Format"
400:         IF NOT mitlayout! THEN
401:             somethinghappend!=TRUE
402:             mitlayout!=TRUE
403:             contextformat!=FALSE
404:             ascii!=FALSE
405:             MENU 7,256
406:             MENU 6,16+64
407:             MENU 8,16+64
408:         ENDIF
409:     ELSE IF menstr$=" ASCII-Format"
410:         IF NOT ascii! THEN
411:             somethinghappend!=TRUE
412:             ALERT 0,"Keine Speicherung von Schriftattributen
",1,"Alles klar",res&
413:             mitlayout!=FALSE
414:             contextformat!=FALSE
415:             ascii!=TRUE
416:             MENU 8,256
417:             MENU 6,16+64
418:             MENU 7,16+64
419:         ENDIF
420:     ELSE IF menstr$="Linker Rand"
421:         GOSUB linkerrand
422:     ELSE IF menstr$="Rechter Rand"
423:         GOSUB rechterrang
424:     ELSE IF menstr$="Zeilenzahl"
425:         GOSUB zeilenzahl
426:     ELSE IF menstr$=" Normal"
427:         bold!=FALSE
428:         underline!=FALSE
429:         italic!=FALSE
430:         superscript!=FALSE
431:         subscript!=FALSE
432:         scriptoffset&=0
433:         attribut!=0
434:         MENU 12,16+64
435:         MENU 13,16+64
436:         MENU 14,16+64
437:         MENU 15,16+64
438:         MENU 16,16+64
439:         MENU 17,16+64
440:         GOSUB checkmenunormal
441:     ELSE IF menstr$=" Fett"
442:         bold!=NOT bold!
443:         attribut!=XOR(attribut!,boldb!)
444:         IF bold! THEN
445:             MENU 12,256+16+64
446:         ELSE
447:             MENU 12,16+64
448:         ENDIF
449:         GOSUB checkmenunormal
450:     ELSE IF menstr$=" Unterstrichen"
451:         underline!=NOT underline!
452:         attribut!=XOR(attribut!,underlineb!)
453:         IF underline! THEN
454:             MENU 13,256+16+64
455:         ELSE
456:             MENU 13,16+64
457:         ENDIF
458:         GOSUB checkmenunormal
459:     ELSE IF menstr$=" Kursiv"
460:         italic!=NOT italic!
461:         attribut!=XOR(attribut!,italicb!)
462:         IF italic! THEN
463:             MENU 14,256+16+64
464:         ELSE
465:             MENU 14,16+64
466:         ENDIF
467:         GOSUB checkmenunormal
468:     ELSE IF menstr$=" Hoch"
469:         superscript!=NOT superscript!
470:         attribut!=XOR(attribut!,superscriptb!)
471:         IF superscript! THEN
472:             scriptoffset&=-1
473:             MENU 15,256+16+64
474:             IF subscript! THEN ! Hoch und tief nicht gleichz
eitig
475:                 subscript!=FALSE
476:                 attribut!=XOR(attribut!,subscriptb!)
477:                 MENU 16,16+64
478:             ENDIF
479:         ELSE
480:             scriptoffset&=0
481:             MENU 15,16+64
482:         ENDIF

```

```

483:         GOSUB checkmenunormal
484:     ELSE IF menstr$=" Tief"
485:         subscript!=NOT subscript!
486:         attribut!=XOR(attribut!,subscriptb!)
487:         IF subscript! THEN
488:             scriptoffset&=1
489:             MENU 16,256+16+64
490:             IF superscript! THEN ! Hoch und tief nicht gleic
hzeitig
491:                 superscript!=FALSE
492:                 attribut!=XOR(attribut!,superscriptb!)
493:                 MENU 15,16+64
494:             ENDIF
495:         ELSE
496:             scriptoffset&=0
497:             MENU 16,16+64
498:         ENDIF
499:         GOSUB checkmenunormal
500:     ELSE IF menstr$=" Insert"
501:         toggleinset
502:     ELSE IF menstr$=" Overwrite"
503:         toggleinset
504:     ELSE IF menstr$="Suchen"
505:         suchen
506:     ELSE IF menstr$="Ersetzen"
507:         ersetzen
508:     ELSE IF menstr$="Weitersuchen"
509:         weitersuchen
510:     ELSE IF menstr$="Absatz"
511:         absatzformatieren
512:     ELSE IF menstr$="Start"
513:         ctrlk("B")
514:     ELSE IF menstr$="Ende"
515:         ctrlk("K")
516:     ELSE IF menstr$="Kopieren"
517:         ctrlk("C")
518:     ELSE IF menstr$="Verschieben"
519:         ctrlk("V")
520:     ELSE IF menstr$="Freigegeben"
521:         ctrlk("H")
522:     ELSE IF menstr$="Loeschen"
523:         ctrlk("Y")
524:     ENDIF
525:     updatecurspos
526:     cursoron ! <---- Cursor wieder an
527:     RETURN
528: > PROCEDURE checkmenunormal
529:     IF (bold! OR underline! OR italic! OR dwidth! OR sup
erscript! OR subscript!) THEN
530:         MENU 11,16+64
531:     ELSE
532:         MENU 11,256
533:     ENDIF
534:     RETURN
535: > PROCEDURE updatescr
536:     LOCAL start&,ende&,i&,temp&
537:     start&=texty&-(linespp& DIV 2)
538:     IF start&<1 THEN
539:         start&=1
540:     ENDIF
541:     ende&=start&+linespp&-1
542:     IF ende&>maxlines& THEN
543:         ende&=maxlines&
544:     ENDIF
545:     mycls
546:     temp&=texty&
547:     stripline
548:     FOR i&=start& TO ende&
549:         texty&=i&
550:         showline(lrand&,i&-start&+1,text$(i&),attr$(i&),((
i& MOD zeilenzahl!&)=0))
551:     NEXT i&
552:     texty&=temp&
553:     stretchline
554:     GOSUB drawborder
555:     IF textx&>LEN(text$(texty&)) THEN
556:         findrpos
557:     ENDIF
558:     scry&=texty&-start&+1
559:     RETURN
560: > PROCEDURE drawborder
561:     COLOR 3,0
562:     IF lrand&<>1 THEN
563:         LINE (lranda&-1)*8-2+1randoffs&,10,(lranda&-1)*8-2+1
randoffs&,winheight&-2
564:     ENDIF
565:     LINE rrand&*8+1randoffs&,10,rranda&*8+1randoffs&,winh
eight&-2
566:     COLOR 1,0
567:     RETURN
568: > PROCEDURE updatecurspos
569:     IF scrx&>rranda&+1 THEN
570:         scrx&=lranda&
571:         textx&=1
572:         INC scry&
573:         stripline
574:         INC texty&
575:         stretchline
576:     ENDIF
577:     IF textx&>1 THEN
578:         IF MID$(text$(texty&),textx&-1,1)=parac$ THEN

```



```

579:      scrx=&lrand&
580:      textx=&l
581:      INC scrx&
582:      stripline
583:      INC texty&
584:      stretchline
585:    ELSE
586:      cursormovement!=FALSE
587:    ENDIF
588:  ENDF
589:  IF scrx<&lrand& THEN
590:    scrx=&rrand&
591:    stripline
592:    DEC texty&
593:    stretchline
594:    IF texty<&l THEN
595:      texty=&l
596:    ELSE
597:      DEC scrx&
598:    ENDIF
599:    textx=&LEN(text$(texty&))
600:  ENDF
601:  IF scrx>&linespp&
602:    DEC scrx&
603:    scrollup(1)
604:  ENDF
605:  IF scrx<&l THEN
606:    INC scrx&
607:    scrolldown(1)
608:  ENDF
609:  RETURN
610: > PROCEDURE scrolluparea(starty&,endy&,nr&)
611:  VOID ScrollRaster(rastport%,0,lineoffs&*nr&,2,10+(st
arty&-1)*lineoffs&,639-4,10+endy&*lineoffs&)
612:  GOSUB drawborder
613:  RETURN
614: > PROCEDURE scrollup(nr&)
615:  VOID ScrollRaster(rastport%,0,lineoffs&*nr&,2,10,639
-4,linespp&*10+10)
616:  GOSUB showline(lrand&,linespp&,text$(texty&),attr$(t
exty&),(texty& MOD zeilenzahl&)=0)
617:  GOSUB drawborder
618:  RETURN
619: > PROCEDURE scrolldown(nr&)
620:  VOID ScrollRaster(rastport%,0,-lineoffs&*nr&,2,10,63
9-4,linespp&*10+10)
621:  GOSUB showline(lrand&,1,text$(texty&),attr$(texty&),
((texty& MOD zeilenzahl&)=0))
622:  GOSUB drawborder
623:  RETURN
624: > PROCEDURE scrolldownarea(starty&,endy&,nr&)
625:  VOID ScrollRaster(rastport%,0,-lineoffs&*nr&,2,10+(s
tarty&-1)*lineoffs&,639-4,10+endy&*lineoffs&)
626:  GOSUB drawborder
627:  RETURN
628: > PROCEDURE showline(x&,y&,line$,attrstr$,seitenumbruc
h!)
629:  LOCAL i&,xx&,attr!,laenge&,c$,temp$
630:  laenge&=LEN(line$)
631:  IF (laenge&>0) THEN
632:    IF laenge&+x&>80 THEN
633:      line$=LEFT$(line$,80-x&)
634:      laenge&=LEN(line$)
635:    ENDIF
636:    i&=1
637:    xx&=x&
638:    IF MID$(line$,i&,1)<>" " THEN
639:      attr!=ASC(MID$(attrstr$,i&,1))
640:    ELSE
641:      attr!=0
642:    ENDIF
643:    IF AND(attr!,superscriptb!)=superscriptb! THEN
644:      scriptoffset&=-1
645:    ELSE IF AND(attr!,subscriptb!)=subscriptb!
646:      scriptoffset&=1
647:    ELSE
648:      scriptoffset&=0
649:    ENDIF
650:    GOSUB setstyle(rastport%,AND(attr!,boldb!+italicb!
+underlineb!))
651:    temp$=""
652:    WHILE ((ASC(MID$(attrstr$,i&,1))=attr!) OR (MID$(l
ine$,i&,1)=" ") AND (i&<=laenge&))
653:      c$=MID$(line$,i&,1)
654:      IF c$=" " THEN
655:        temp$=temp$+SPACE$(ASC(MID$(attrstr$,i&,1)))
656:        xx&=xx&+ASC(MID$(attrstr$,i&,1))
657:        INC i&
658:      ELSE
659:        temp$=temp$+c$
660:        INC xx&
661:        INC i&
662:      ENDIF
663:    WEND
664:    IF texty&=blockstart& AND texty&<=blockende& AND
blockstart&<>0 AND blockende&<>0 THEN
665:      COLOR 2,0
666:    ELSE
667:      COLOR 1,0
668:    ENDIF
669:    TEXT (x&-1)*8+1randoffs&,y&*lineoffs&+8+scriptoffs
et&,temp$

```

```

670:      drawborder
671:      GOSUB showline(xx&,y&,MID$(line$,i&),MID$(attrstr$
,i&),seitenumbruch!)
672:    ELSE
673:      clearline(x&,80,y&)
674:    ENDIF
675:    IF seitenumbruch! THEN
676:      COLOR 2,0
677:      LINE 2,y&*lineoffs&+10,637,y&*lineoffs&+10
678:      COLOR 1,0
679:    ENDIF
680:  RETURN
681: > PROCEDURE clearline(x&,x2&,y&)
682:  COLOR 0,0
683:  PBOX (x&-1)*8-2*(x&=1)+1randoffs&,y&*lineoffs&+10,(x
2&)*8-4,(y&-1)*lineoffs&+11
684:  COLOR 1,0
685:  RETURN
686: > PROCEDURE initmenu(VAR menue$())
687:  LOCAL i&
688:  RESTORE menuetexte
689:  i&=0
690:  DO
691:    READ menue$(i&)
692:    EXIT IF menue$(i&)=""
693:    INC i&
694:  LOOP
695:  menue$(i&)=""
696:  RETURN
697: > PROCEDURE ctrlc
698:  GOSUB closeallend
699:  STOP
700:  RETURN
701: > PROCEDURE generror
702:  GOSUB closeallend
703:  RETURN
704: > PROCEDURE closeall
705:  CLOSEW #0
706:  CLOSES 1
707:  RETURN
708: > PROCEDURE closeallend
709:  GOSUB closeall
710:  END
711:  RETURN
712: > PROCEDURE cursor(x&,y&)
713:  GRAPHMODE 2
714:  COLOR 3,0
715:  PBOX (x&-1)*8+1randoffs&,(y&-1)*lineoffs&+11,x&*8+1r
andoffs&,y&*lineoffs&+10
716:  COLOR 1,0
717:  GRAPHMODE 1
718:  RETURN
719: > PROCEDURE setstyle(rport%,style&)
720:  LOCAL old&,new&
721:  old&=AskSoftStyle(rport%)
722:  new&=SetSoftStyle(rport%,style&,old&)
723:  IF (style& AND 4)=4 THEN
724:    VOID SetDrMd(rport%,0)
725:  ELSE
726:    VOID SetDrMd(rport%,1)
727:  ENDIF
728:  RETURN
729: > PROCEDURE textladen
730:  LOCAL res&,i&,l1div2&,dummy$
731:  res&=1
732:  IF somethinghappend! THEN
733:    ALERT 0,"Laden und alten Text loeschen?",2,"Na kl
ar !! Bloß nicht !",res&
734:  ENDIF
735:  IF res&=1 THEN
736:    FILESELECT "Datei Laden","Laden",workfile$,filenam
e$
737:    IF filename$<>" " THEN
738:      somethinghappend!=FALSE
739:      blockstart&=0
740:      blockende&=0
741:      scrx=&lrand&
742:      texty=&l
743:      textx=&l
744:      OPEN "i",#1,filename$
745:      LINE INPUT #1,header$
746:      IF LEFT$(header$,7)<>"CONTEXT" THEN
747:        header$=""
748:        asci!=TRUE
749:        contextformat!=FALSE
750:        mitlayout!=FALSE
751:        CLOSE #1
752:        OPEN "i",#1,filename$
753:      ELSE
754:        asci!=FALSE
755:        IF LEN(header$)>7 THEN
756:          mitlayout!=TRUE
757:          CLOSE #1
758:          OPEN "i",#1,filename$
759:          INPUT #1,dummy$,lrand&,rrand&,zeilenzahl&
760:          LINE INPUT #1,dummy$
761:        ELSE
762:          contextformat!=TRUE
763:        ENDIF
764:      ENDIF
765:    ENDIF
766:    i&=1

```



```

767: WHILE NOT EOF(#1)
768: LINE INPUT #1, text$(i&)
769: INC i&
770: WEND
771: lastline=i&-1
772: IF (lastline MOD 2<>0) AND (NOT asciil) THEN
773: wrongformaterror
774: ENDIF
775: lldiv2=lastline DIV 2
776: IF NOT asciil THEN
777: FOR i&=lldiv2+1 TO lastline&
778: attr$(i&-lldiv2&)=text$(i&)
779: IF LEN(text$(i&))<>LEN(text$(i&-lldiv2&)) TH
EN
780: wrongformaterror
781: ENDIF
782: NEXT i&
783: lastline=lldiv2&
784: ELSE
785: insertattributes(0)
786: ENDIF
787: FOR i&=lastline+1 TO maxlines&
788: text$(i&)=""
789: attr$(i&)=""
790: NEXT i&
791: updatescr
792: CLOSE #1
793: ENDIF
794: workfile$=filename$
795: ENDIF
796: scrx=lrand&
797: textx=1
798: texty=1
799: RETURN
800: > PROCEDURE textspeichern
801: IF somethinghappend! THEN
802: FILESELECT "Datei Speichern", "Speichern", workfile$
, filename$
803: IF filename$<>"" THEN
804: OPEN "o", #1, filename$
805: IF mitlayout! THEN
806: PRINT #1, "CONTEXT,";
807: WRITE #1, lrand&, rrand&, zeilenzahl&
808: ENDIF
809: IF context$format! THEN
810: PRINT #1, "CONTEXT"
811: ENDIF
812: lastline=maxlines&
813: WHILE LEN(text$(lastline&))=0
814: DEC lastline&
815: WEND
816: STORE #1, text$( ), 1 TO lastline&
817: IF NOT asciil THEN
818: STORE #1, attr$( ), 1 TO lastline&
819: ENDIF
820: CLOSE #1
821: ENDIF
822: workfile$=filename$
823: ENDIF
824: RETURN
825: > PROCEDURE insertattributes(attr!)
826: LOCAL i&, j&
827: FOR i&=1 TO maxlines&
828: attr$(i&)=STRING$(LEN(text$(i&)), CHR$(attr!))
829: FOR j&=1 TO LEN(text$(i&))
830: IF MID$(text$(i&), j&, 1)=" " THEN
831: MID$(attr$(i&), j&, 1)=CHR$(1)
832: ENDIF
833: NEXT j&
834: NEXT i&
835: asciil=FALSE
836: RETURN
837: > PROCEDURE stretchline
838: IF LEN(text$(texty&))<80 THEN
839: text$(texty&)=text$(texty&)+SPACE$(80-LEN(text$(te
xty&)))
840: attr$(texty&)=attr$(texty&)+STRING$(80-LEN(attr$(t
exty&)), CHR$(0))
841: ENDIF
842: RETURN
843: > PROCEDURE stripline
844: LOCAL i&, j&, line$, comp$, asciil$, attrstr$
845: line$=text$(texty&)
846: attrstr$=attr$(texty&)
847: comp$=CHR$(0)
848: asciil$=CHR$(1)
849: i&=LEN(line$)
850: IF i&>1 THEN
851: WHILE MID$(attrstr$, i&, 1)=comp$ AND (MID$(line$,
i&, 1)=" ")
852: DEC i&
853: EXIT IF i&<1
854: WEND
855: ENDIF
856: line$=LEFT$(line$, i&)
857: attrstr$=LEFT$(attrstr$, i&)
858: FOR j&=i& TO 1 STEP -1
859: IF (MID$(attrstr$, j&, 1)=comp$) AND (MID$(line$, j&,
1)=" ") THEN
860: MID$(attrstr$, j&, 1)=asciil$
861: ENDIF
862: NEXT j&

```

```

863: text$(texty&)=line$
864: attr$(texty&)=attrstr$
865: RETURN
866: > PROCEDURE findrpos
867: LOCAL pos&
868: pos&=INSTR(text$(texty&), parac$)
869: IF pos&<>0 THEN
870: scrx=lrand&
871: FOR i&=1 TO LEN(text$(texty&))
872: EXIT IF MID$(text$(texty&), i&, 1)=parac$
873: IF MID$(text$(texty&), i&, 1)=" " THEN
874: scrx=scrx+ASC(MID$(attr$(texty&), i&, 1))
875: ELSE
876: INC scrx&
877: ENDIF
878: NEXT i&
879: IF MID$(text$(texty&), i&, 1)=parac$ THEN
880: textx=i&
881: ELSE
882: textx=LEN(text$(texty&))-1
883: ENDIF
884: IF textx=0 THEN
885: textx=1
886: ENDIF
887: ELSE
888: scrx=lrand&
889: FOR i&=1 TO LEN(text$(texty&))
890: EXIT IF (MID$(text$(texty&), i&, 1)=" ") AND (MID$
(attr$(texty&), i&, 1)=CHR$(0))
891: IF MID$(text$(texty&), i&, 1)=" " THEN
892: scrx=scrx+ASC(MID$(attr$(texty&), i&, 1))
893: ELSE
894: INC scrx&
895: ENDIF
896: NEXT i&
897: textx=i&
898: ENDIF
899: RETURN
900: > PROCEDURE pageup
901: LOCAL temp&
902: texty&=texty&-linespp&
903: IF texty&<0 THEN
904: texty&=1
905: ENDIF
906: IF rechtsraus THEN
907: findrpos
908: ENDIF
909: temp&=texty&
910: updatescr
911: texty&=temp&
912: updatecurspos
913: RETURN
914: > PROCEDURE pagedown
915: LOCAL temp&
916: texty&=texty&+linespp&
917: IF texty&>maxlines& THEN
918: texty&=maxlines&
919: ENDIF
920: IF rechtsraus THEN
921: findrpos
922: ENDIF
923: temp&=texty&
924: updatescr
925: texty&=temp&
926: updatecurspos
927: RETURN
928: > PROCEDURE wrongformaterror
929: LOCAL res&
930: ALERT 0, "! Falsches Textformat !", 2, "Weitermachen !"
Vergiß es !", res&
931: IF res&=2 THEN
932: RUN
933: ENDIF
934: RETURN
935: > PROCEDURE justamoment
936: OPENW #1, 180, 90, 280, 50, 0, 0, 1
937: TITLEW #1, "Momentchen ", "Momentchen "
938: PRINT
939: PRINT
940: PRINT " Einen Augenblick Geduld bitte"
941: OPENW #0
942: RETURN
943: > PROCEDURE linkerrand
944: OPENW #1, 180, 100, 280, 22, 0, 0, 1
945: TITLEW #1, "Linker Rand ", "Linker Rand "
946: REPEAT
947: PRINT " Linken Rand auf (1..;rrand&);": ";
948: INPUT lrand&
949: UNTIL (lrand&=1) AND (lrand&<=rrand&)
950: CLOSEW #1
951: OPENW #0
952: mycls
953: scrx=lrand&
954: textx=1
955: updatescr
956: RETURN
957: > PROCEDURE rechterrhand
958: OPENW #1, 180, 100, 280, 22, 0, 0, 1
959: TITLEW #1, "Rechter Rand ", "Rechter Rand "
960: REPEAT
961: PRINT " Rechter Rand auf (;lrand&;..80): ";
962: INPUT rrand&
963: UNTIL (rrand&=lrand&) AND (rrand&<=80)

```



```

964: CLOSEW #1
965: OPENW #0
966: textx&=1
967: scrx&=1rand&
968: mycls
969: updatescr
970: RETURN
971: > PROCEDURE zeilenzahl
972: OPENW #1,180,100,280,22,0,0,1
973: TITELW #1," Zeilen pro Seite ", " Zeilen pro Seite "
974: REPEAT
975: PRINT " Zeilen pro Seite (30..150): ";
976: INPUT zeilenzahl&
977: UNTIL (zeilenzahl& >= 30) AND (zeilenzahl& <= 150)
978: CLOSEW #1
979: OPENW #0
980: mycls
981: updatescr
982: RETURN
983: > PROCEDURE mycls
984: COLOR 0,0
985: PBOX 1randoffs&,linespp&*lineoffs&+10,639-4,11
986: COLOR 1,0
987: RETURN
988: > PROCEDURE headline
989: LOCAL headline$
990: IF NOT insertmode! THEN
991: headline$=progtitel$+SPACES(50)+"OVERWRITE"
992: ELSE
993: headline$=progtitel$+SPACES(50)+"INSERT"
994: ENDIF
995: TITELW #0,headline$,headline$
996: RETURN
997: > PROCEDURE endlne
998: findrpos
999: RETURN
1000: > PROCEDURE pos1
1001: textx&=1
1002: scrx&=1rand&
1003: RETURN
1004: > PROCEDURE toggleinsert
1005: insertmode!=NOT insertmode!
1006: headline
1007: IF insertmode! THEN
1008: MENU 30,256
1009: MENU 31,16+64
1010: ELSE
1011: MENU 30,16+64
1012: MENU 31,256
1013: ENDIF
1014: RETURN
1015: > PROCEDURE delline(i&)
1016: DELETE text$(i&)
1017: DELETE attr$(i&)
1018: scrolluparea(scrx&,linespp&,1)
1019: stretchline
1020: showline(1rand&,linespp&,text$(textx&+linespp&-scry&
),attr$(textx&+linespp&-scry&),(textx&+linespp&-scry&)
MOD zeilenzahl&=0)
1021: drawborder
1022: textx&=1
1023: scrx&=1rand&
1024: RETURN
1025: > PROCEDURE insline(i&)
1026: stripline
1027: INSERT text$(i&)=""
1028: INSERT attr$(i&)=""
1029: scrolldownarea(scrx&,linespp&,1)
1030: drawborder
1031: stretchline
1032: textx&=1
1033: scrx&=1rand&
1034: RETURN
1035: > PROCEDURE checkwrap(c$) ! Umbrechen in naechste Ze
ile ?
1036: LOCAL lastl&,riesenwort!
1037: lastl&=textx&
1038: ltextx&=textx&
1039: IF scrx&>rrand& THEN
1040: IF textx&<maxlines&
1041: IF NOT FN nextlineempty THEN
1042: INSERT text$(textx&+1)=MID$(text$(textx&),text
x&)
1043: INSERT attr$(textx&+1)=MID$(attr$(textx&),text
x&)
1044: text$(textx&)=LEFT$(text$(textx&),textx&-1)
1045: attr$(textx&)=LEFT$(attr$(textx&),textx&-1)
1046: scrolldownarea(textx&+1,linespp&,1)
1047: showline(1rand&,scry&,text$(textx&),attr$(text
y&),(textx&) MOD zeilenzahl&=0)
1048: stripline
1049: INC textx&
1050: drawborder
1051: showline(1rand&,scry&+1,text$(textx&),attr$(te
xty&),(textx&) MOD zeilenzahl&=0)
1052: textx&=1
1053: DEC textx&
1054: ELSE
1055: text$(textx&+1)=MID$(text$(textx&),textx&)
1056: attr$(textx&+1)=MID$(attr$(textx&),textx&)
1057: text$(textx&)=LEFT$(text$(textx&),textx&-1)
1058: attr$(textx&)=LEFT$(attr$(textx&),textx&-1)
1059: showline(1rand&,scry&,text$(textx&),attr$(text
y&),(textx&) MOD zeilenzahl&=0)

```

```

1060: stripline
1061: INC textx&
1062: showline(1rand&,scry&+1,text$(textx&),attr$(te
xty&),(textx&) MOD zeilenzahl&=0)
1063: textx&=1
1064: DEC textx&
1065: ENDIF
1066: INC textx&
1067: INC scry&
1068: textx&=1
1069: scrx&=1rand&
1070: IF c$=" " THEN
1071: stretchline
1072: MID$(text$(lastl&),ltextx&,1)=" "
1073: MID$(attr$(lastl&),ltextx&,1)=CHR$(1)
1074: wrapped!=TRUE
1075: ELSE
1076: i&=LEN(text$(lastl&))+1
1077: WHILE (MID$(text$(lastl&),i&,1)<>" ") OR ((MID
$(text$(lastl&),i&,1)=" " AND MID$(attr$(lastl&),i&,1)
=CHR$(0)))
1078: DEC i&
1079: EXIT IF i&=0
1080: WEND
1081: riesenwort!=(i&=0)
1082: INC i&
1083: IF NOT riesenwo! THEN
1084: text$(textx&)=MID$(text$(lastl&),i&)
1085: attr$(textx&)=MID$(attr$(lastl&),i&)
1086: text$(lastl&)=LEFT$(text$(lastl&),i&-1)
1087: attr$(lastl&)=LEFT$(attr$(lastl&),i&-1)
1088: ENDIF
1089: stripline
1090: stretchline
1091: ENDIF
1092: DEC textx&
1093: stripline
1094: INC textx&
1095: clearline(1rand&,80,scry&)
1096: showline(1rand&,scry&,text$(textx&),attr$(textx&
),(textx& MOD zeilenzahl&=0)
1097: clearline(1rand&,80,scry&-1)
1098: showline(1rand&,scry&-1,text$(lastl&),attr$(last
l&),(lastl& MOD zeilenzahl&=0)
1099: drawborder
1100: findrpos
1101: ENDIF
1102: ENDIF
1103: RETURN
1104: > PROCEDURE mymouse(VAR mausx,mausy,mausk) ! Maus mit
Textkoordinaten
1105: MOUSE mausx,mausy,mausk
1106: mausx=(mausx DIV 8)+1
1107: mausy=((mausy-11) DIV lineoffs&)+1
1108: IF mausy<1 THEN
1109: mausy=1
1110: ENDIF
1111: IF mausy>linespp& THEN
1112: mausy=linespp&
1113: ENDIF
1114: RETURN
1115: > PROCEDURE cursoron
1116: cursor(scrx&,scry&)
1117: RETURN
1118: > PROCEDURE cursoroff
1119: cursor(scrx&,scry&)
1120: RETURN
1121: > PROCEDURE findtextx ! Text-X-Position suchen
1122: LOCAL i&,pos&
1123: pos&=INSTR(text$(textx&),parac$)
1124: textx&=1
1125: i&=1rand&
1126: WHILE i&<scrx&
1127: IF MID$(text$(textx&),textx&,1)=" " THEN
1128: i&=i&+ASC(MID$(attr$(textx&),textx&,1))
1129: ELSE
1130: INC i&
1131: ENDIF
1132: INC textx&
1133: EXIT IF (MID$(text$(textx&),textx&,1)=" ") AND (MI
D$(attr$(textx&),textx&,1)=CHR$(0))
1134: WEND
1135: IF textx&>pos& AND (pos&<>0) THEN
1136: textx&=pos&
1137: ENDIF
1138: RETURN
1139: > PROCEDURE findscrx ! Bildschirm-X-Position such
en
1140: LOCAL i&
1141: scrx&=1rand&
1142: FOR i&=1 TO textx&
1143: IF MID$(text$(textx&),i&,1)=" " THEN
1144: scrx&=scrx&+ASC(MID$(attr$(textx&),i&,1))
1145: ELSE
1146: IF MID$(text$(textx&),i&,1)<>parac$ THEN
1147: INC scrx&
1148: ENDIF
1149: ENDIF
1150: NEXT i&
1151: RETURN
1152: > PROCEDURE suchen
1153: LOCAL pos&
1154: OPENW #1,180,100,280,22,0,0,1
1155: TITELW #1," Suchen ", " Suchen "

```



```

1156: PRINT " Suchbegriff ";
1157: INPUT such$
1158: IF such$<>"" THEN
1159:   ersetze$=""
1160:   lastsearch:=TRUE
1161:   stripline
1162:   REPEAT
1163:     pos:=INSTR(text$(texty&),such$,textx&)
1164:     IF pos=0 THEN
1165:       INC texty&
1166:       textx:=1
1167:     ELSE
1168:       textx:=pos&
1169:     ENDIF
1170:   UNTIL texty>lastline& OR pos<>0
1171:   IF pos=0 THEN
1172:     DEC texty&
1173:     stretchline
1174:     findrpos
1175:   ELSE
1176:     stretchline
1177:     findscrx
1178:   ENDIF
1179:   lastsearchx:=textx&
1180:   lastsearchy:=texty&
1181: ENDIF
1182: CLOSEW #1
1183: OPENW #0
1184: mycls
1185: updatescr
1186: RETURN
1187: > PROCEDURE ersetzen
1188: LOCAL pos&
1189: OPENW #1,180,100,280,22,0,0,1
1190: TITLEW #1," Suchen "," Suchen "
1191: PRINT " Suchbegriff ";
1192: INPUT such$
1193: IF such$<>"" THEN
1194:   OPENW #2,180,130,280,22,0,0,1
1195:   TITLEW #2," Ersetzen durch "," Ersetzen durch "
1196:   PRINT " Ersetzen durch ";
1197:   INPUT ersetze$
1198:   IF ersetze$<>"" THEN
1199:     lastsearch:=FALSE
1200:     stripline
1201:     REPEAT
1202:       pos:=INSTR(text$(texty&),such$,textx&)
1203:       IF pos=0 THEN
1204:         INC texty&
1205:         textx:=1
1206:       ELSE
1207:         textx:=pos&
1208:       ENDIF
1209:     UNTIL texty>lastline& OR pos<>0
1210:     IF pos=0 THEN
1211:       DEC texty&
1212:       stretchline
1213:       findrpos
1214:     ELSE
1215:       text$(texty&)=LEFT$(text$(texty&),textx-1)+er
1216:       setz$+MID$(text$(texty&),textx+LEN(such$))
1217:       stretchline
1218:       findscrx
1219:     ENDIF
1220:     lastsearchx:=textx&
1221:     lastsearchy:=texty&
1222:   ENDIF
1223: ENDIF
1224: CLOSEW #2
1225: CLOSEW #1
1226: OPENW #0
1227: mycls
1228: updatescr
1229: RETURN
1230: > PROCEDURE weitersuchen ! Weitersuchen/Ersetzen
1231: LOCAL pos&
1232: texty&:=lastsearchy&
1233: textx&:=lastsearchx&
1234: INC textx&
1235: IF such$<>"" THEN
1236:   stripline
1237:   REPEAT
1238:     pos:=INSTR(text$(texty&),such$,textx&)
1239:     IF pos=0 THEN
1240:       INC texty&
1241:       textx:=1
1242:     ELSE
1243:       textx:=pos&
1244:     ENDIF
1245:   UNTIL texty>lastline& OR pos<>0
1246:   IF pos=0 THEN
1247:     DEC texty&
1248:     stretchline
1249:     findrpos
1250:   ELSE
1251:     IF NOT lastsearch! THEN
1252:       text$(texty&)=LEFT$(text$(texty&),textx-1)+er
1253:       setz$+MID$(text$(texty&),textx+LEN(such$))
1254:     ENDIF
1255:     stretchline
1256:     findscrx
1257:   ENDIF

```

```

1256:   lastsearchx:=textx&
1257:   lastsearchy:=texty&
1258: ENDIF
1259: mycls
1260: updatescr
1261: RETURN
1262: > PROCEDURE absatzformatieren
1263: LOCAL start&,ende&
1264: stripline
1265: IF text$(texty&)<>parac$ THEN
1266:   start&:=FN findabsatzstart
1267:   ende&:=FN findabsatzende
1268:   IF start&<>ende& OR LEN(text$(texty&))>(rrand-1ra
1269:     nd+1) THEN
1270:     formatieren(start&,ende&)
1271:     updatecurspos
1272:     mycls
1273:     updatescr
1274:   ENDIF
1275: ENDIF
1276: > PROCEDURE formatieren(start&,ende&)
1277: LOCAL i&,breite&,altezeile$,neuezeile$,item$,alteatr
1278: r$,neueattr$,aktattr$,ziely&,zielende&,itemlen&,wrapped
1279: &,altstart&,altende&
1280: wrapped:=0
1281: breite&:=rrand-1rand+1
1282: ziely&:=start&
1283: zielende&:=ende&
1284: texty&:=start&
1285: DO
1286:   altezeile$=text$(texty&)
1287:   alteatr$=attr$(texty&)
1288:   neuezeile$=""
1289:   neueattr$=""
1290:   REPEAT
1291:     item$:=FN getnextitem$(altezeile$)
1292:     itemlen:=LEN(item$)
1293:     aktattr$:=FN getattr$(alteatr$,itemlen&)
1294:     IF item$="" THEN
1295:       INC texty&
1296:       EXIT IF texty>ende&
1297:       altezeile$=text$(texty&)
1298:       alteatr$=attr$(texty&)
1299:     ENDIF
1300:     neuezeile$=neuezeile$+item$
1301:     neueattr$=neueattr$+aktattr$
1302:   UNTIL ((LEN(neuezeile$)>breite&) AND (item$<>" "))
1303:   OR texty>ende&
1304:   ' Letztes Item von Zeile wieder entfernen
1305:   neuezeile$=LEFT$(neuezeile$,LEN(neuezeile$)-LEN(it
1306:     em$))
1307:   neueattr$=LEFT$(neueattr$,LEN(neueattr$)-LEN(item$
1308:     ))
1309:   ' Zurueckschreiben der formatierten Zeilen
1310:   text$(ziely&)=neuezeile$
1311:   attr$(ziely&)=neueattr$
1312:   INC ziely&
1313:   IF (ziely&>zielende&) AND (texty<=ende&) THEN
1314:     INSERT text$(texty&)=""
1315:     INSERT attr$(ziely&)=""
1316:     INC zielende&
1317:     INC wrapped&
1318:   ENDIF
1319:   ' bertrag der Restzeilen
1320:   INC texty&
1321:   IF texty>ende& AND (altezeile$<>"" OR item$<>")
1322:     THEN
1323:     INSERT text$(texty&)=item$+altezeile$
1324:     INSERT attr$(texty&)=aktattr$+alteatr$
1325:   ELSE
1326:     text$(texty&)=item$+altezeile$+text$(texty&)
1327:     attr$(texty&)=aktattr$+alteatr$+attr$(texty&)
1328:   ENDIF
1329:   EXIT IF texty>ende&
1330: LOOP
1331: ' Was zuviel oder zuwenig ?
1332: IF ziely&<texty& THEN
1333:   FOR i&=ziely& TO texty&-1
1334:     DELETE text$(ziely&)
1335:     DELETE attr$(ziely&)
1336:   NEXT i&
1337: ENDIF
1338: FOR i&=1 TO wrapped&
1339:   DELETE text$(ziely&+1)
1340:   DELETE attr$(ziely&+1)
1341: NEXT i&
1342: wrapped&:=0
1343: texty&:=ziely&+1
1344: textx&:=1
1345: scrx&:=1rand
1346: updatecurspos
1347: stripline
1348: MID$(text$(texty&),textx&,1)=parac$
1349: MID$(attr$(texty&),textx&,1)=CHR$(0)
1350: textx&:=1
1351: scrx&:=1rand&
1352: INC texty&
1353: INC start&
1354: WHILE LEN(TRIM$(text$(i&)))<=breite& AND (i&<texty&)
1355:   INC i&
1356: WEND

```



```

1351: IF i<texty& AND LEN(TRIM$(text$(i&)))>breite& THEN
1352:   formatieren(i&,texty&-1) ! Ggf. rekursiv nochmal f
      ormatieren
1353:   ENDIF
1354: RETURN
1355: > PROCEDURE ctrlk(c$)
1356:   stripline
1357:   IF UPPER$(c$)="B" OR c$=CHR$(2) THEN
1358:     blockstart&=texty&
1359:   ELSE IF UPPER$(c$)="K" OR c$=CHR$(11)
1360:     blockende&=texty&
1361:   ELSE IF UPPER$(c$)="H" OR c$=CHR$(8)
1362:     blockstart&=0
1363:     blockende&=0
1364:     updatescr
1365:   ELSE IF UPPER$(c$)="Y" OR c$=CHR$(25)
1366:     blockdelete
1367:     updatescr
1368:   ELSE IF UPPER$(c$)="V" OR c$=CHR$(22)
1369:     blockmove
1370:   ELSE IF UPPER$(c$)="C" OR c$=CHR$(3)
1371:     blockcopy
1372:   ENDIF
1373:   IF blockende&<>0 AND blockstart&<>0 THEN
1374:     IF blockende&<blockstart& THEN
1375:       SWAP blockende&,blockstart&
1376:     ENDIF
1377:     updatecurspos
1378:     updatescr
1379:   ENDIF
1380: RETURN
1381: > PROCEDURE blockdelete
1382:   LOCAL i&
1383:   IF blockende&<>0 AND blockstart&<>0 THEN
1384:     FOR i&=blockstart& TO blockende&
1385:       DELETE text$(i&)
1386:       DELETE attr$(i&)
1387:     NEXT i&
1388:     blockstart&=0
1389:     blockende&=0
1390:   ENDIF
1391: RETURN
1392: > PROCEDURE blockmove
1393:   LOCAL i&,dest&,offs&,vor!,blocklen&
1394:   IF blockende&<>0 AND blockstart&<>0 THEN
1395:     IF texty&<blockstart& OR texty&>blockende&
1396:       blocklen&=blockende&-blockstart&+1
1397:       vor!=(texty&<blockstart&)
1398:       dest&=texty&
1399:       IF NOT vor! THEN
1400:         offs&=0
1401:       ELSE
1402:         offs&=1
1403:       ENDIF
1404:       FOR i&=blockstart& TO blockende&
1405:         INSERT text$(dest&)=text$(i&+offs&)
1406:         INSERT attr$(dest&)=attr$(i&+offs&)
1407:         IF vor! THEN
1408:           INC offs&
1409:         ENDIF
1410:         INC dest&
1411:       NEXT i&
1412:       IF vor! THEN
1413:         DEC offs&
1414:       ENDIF
1415:       FOR i&=blockstart&+offs& TO blockende&+offs&
1416:         DELETE text$(blockstart&+offs&)
1417:         DELETE attr$(blockstart&+offs&)
1418:       NEXT i&
1419:       IF vor! THEN
1420:         blockende&=texty&+blockende&-blockstart&
1421:         blockstart&=texty&
1422:       ELSE
1423:         blockende&=texty&-1
1424:         blockstart&=texty&-blocklen&
1425:       ENDIF
1426:       textx&=1
1427:       scrx&=1rand&
1428:     ENDIF
1429:   ENDIF
1430: RETURN
1431: > PROCEDURE blockcopy
1432:   LOCAL i&,dest&,offs&,vor!,blocklen&
1433:   IF blockende&<>0 AND blockstart&<>0 THEN
1434:     IF texty&<blockstart& OR texty&>blockende&
1435:       blocklen&=blockende&-blockstart&+1
1436:       vor!=(texty&<blockstart&)
1437:       dest&=texty&
1438:       IF NOT vor! THEN
1439:         offs&=0
1440:       ELSE
1441:         offs&=1
1442:       ENDIF
1443:       FOR i&=blockstart& TO blockende&
1444:         INSERT text$(dest&)=text$(i&+offs&)
1445:         INSERT attr$(dest&)=attr$(i&+offs&)
1446:         IF vor! THEN
1447:           INC offs&
1448:         ENDIF
1449:         INC dest&
1450:       NEXT i&
1451:       IF vor! THEN
1452:         DEC offs&

```

```

1453:   ENDIF
1454:   blockende&=dest&-1
1455:   blockstart&=blockende&-blocklen&+1
1456:   textx&=1
1457:   scrx&=1rand&
1458:   ENDIF
1459:   ENDIF
1460: RETURN
1461: FUNCTION findabsatzstart
1462:   LOCAL weitersuchen!,i&
1463:   weitersuchen!=TRUE
1464:   i&=texty&
1465:   WHILE i&>1 AND weitersuchen!
1466:     DEC i&
1467:     IF RIGHT$(text$(i&))=parac$
1468:       weitersuchen!=FALSE
1469:     INC i&
1470:   ENDIF
1471:   WEND
1472:   RETURN i&
1473: ENDFUNC
1474: FUNCTION findabsatzende
1475:   LOCAL weitersuchen!,i&
1476:   weitersuchen!=RIGHT$(text$(texty&))<>parac$
1477:   i&=texty&
1478:   WHILE i&<lastline& AND weitersuchen!
1479:     INC i&
1480:     IF RIGHT$(text$(i&))=parac$
1481:       weitersuchen!=FALSE
1482:     ENDIF
1483:   WEND
1484:   RETURN i&
1485: ENDFUNC
1486: FUNCTION getnextitem$(VAR line$)
1487:   LOCAL i&,temp$,c$
1488:   IF (LEN(line$)=0) OR (line$=parac$) THEN ! Leerer St
      ring oder Absatzende
1489:     line$=""
1490:     RETURN "" ! Gleich wieder zurueckspringen
1491:   ENDIF
1492:   temp$=""
1493:   c$=""
1494:   DO
1495:     c$=LEFT$(line$,2)
1496:     EXIT IF c$=""
1497:     EXIT IF (c$=" ") AND (LEN(temp$)>0) ! Normale Item
      s enden am Blank
1498:     line$=MID$(line$,2)
1499:     temp$=temp$+c$
1500:     EXIT IF c$=" " ! Blank als eigenes Item
1501:   LOOP
1502:   RETURN temp$
1503: ENDFUNC
1504: FUNCTION getattr$(VAR line$,laenge&)
1505:   LOCAL temp$
1506:   temp$=LEFT$(line$,laenge&)
1507:   line$=MID$(line$,laenge&+1)
1508:   RETURN temp$
1509: ENDFUNC
1510: FUNCTION nextlineempty
1511:   RETURN (LEN(text$(texty&+1))=0)
1512: ENDFUNC
1513: FUNCTION getrastport(winnr&)
1514:   RETURN LPEEK(WINDOW(winnr&)+50)
1515: ENDFUNC
1516: FUNCTION rechtsaus
1517:   LOCAL i&,rrflag!
1518:   rrflag!=TRUE
1519:   FOR i&=textx& TO 80
1520:     IF ((MID$(text$(texty&),i&,1)<>" ") AND MID$(text$(
      texty&),i&,1)<>parac$) OR (MID$(attr$(texty&),i&,1)<>
      CHR$(0))) THEN
1521:       rrflag!=FALSE
1522:     ENDIF
1523:   NEXT i&
1524:   RETURN rrflag!
1525: ENDFUNC
1526: '
1527: '
1528: menuetexte:
1529: DATA Datei,Neu,Laden,Speichern,Quit,-=====
1530: DATA ConText-Format, Layout-Format, ASCII-Format,"
      "
1531: DATA Schrift, Normal, Fett, Unterstrichen, Kursiv,
      Hoch, Tief
1532: DATA ""
1533: DATA Justierung, Linksbuendig, Rechtsbuendig, Zentri
      ert
1534: DATA Blocksatz
1535: DATA ""
1536: DATA Layout,Linker Rand,Rechter Rand,Zeilenzahl
1537: DATA ""
1538: DATA Modus, Insert, Overwrite
1539: DATA ""
1540: DATA Suchen,Suchen,Ersetzen,Weitersuchen
1541: DATA ""
1542: DATA Block,Start,Ende,Kopieren,Verschieben,Loeschen,Fr
      egeben
1543: DATA ""
1544: DATA Formatieren,Absatz
1545: DATA "" ""
1546: DATA "*"

```


Bernd Rudolph

Assembler in BASIC einbinden

Oft ist ein Algorithmus in BASIC viel zu langsam, um ihn sinnvoll nutzen zu können. Doch was kann man dagegen tun? Die beste, aber auch schwierigste Methode heißt assemblerunterstützte BASIC-Programmierung. Das bedeutet nichts anderes, als daß von BASIC aus kleine Assembler-Programme aufgerufen werden, die verschiedene Aufgaben wesentlich schneller erledigen, als es BASIC tun kann.

Ein gutes Beispiel dafür sind IFF-Grafiken. Welcher BASIC-Programmierer hat sich nicht schon einmal darüber geärgert, daß er Bilder, die mit namhaften Malprogrammen erstellt wurden, nur langsam, unvollständig oder gar nicht von BASIC aus laden kann. Doch warum geht es nicht so einfach? Die Grafiken sind nicht, wie beim "guten alten C-64" Byte für Byte binär gespeichert, sondern sie halten sich an das IFF-Format. IFF bedeutet Interchange-File-Format und ist erdacht worden, um die Kompatibilität der Grafikprogramme untereinander zu wahren. Wie dieses Format letztendlich aussieht, ist bereits an vielen anderen Stellen ausführlich beschrieben worden. Lediglich auf den Un-/Pack-Algorithmus werde ich später noch einmal genauer eingehen.

Der Loader (Listing 1)

Bevor man eine Grafik darstellen kann, muß man die Größe der Datei feststellen, Speicher reservieren und die Datei laden. Diese Aufgabe erledigt die erste Routine. Hier werden zum einen die Adressen des Dateinamens, zum anderen aber auch die Adressen zweier Ganzzahlvariablen übergeben. Diese enthalten nach dem Rücksprung ins BASIC die Adresse der Grafik und die Größe des belegten Speichers, um ihn am Ende wieder freigeben zu können. Wichtig ist, daß der Dateiname mit dem ASCII-Code 0 endet, da es sonst zu einem Absturz kommen könnte. Ist die Datei geladen, so wird von BASIC aus ein entsprechendes Screen mit Window geöffnet.

net. Danach tritt der Unpacker in Aktion.

Der Unpacker (Listing 2)

Nacheinander werden die erste Zeile der ersten Bitplane bis zur ersten Zeile der n-ten Bitplane gespeichert. Dann folgt die zweite Zeile der ersten Bitplane bis hin zur letzten Zeile der n-ten Bitplane. Jede Zeile wird nun noch gepackt, das heißt, enthält eine Zeile einer Bitplane, beispielsweise 40mal den Wert 0, so wird als erstes ein Anzahlbyte und dann ein Datenbyte gespeichert, in unserem Beispiel 40 und 0. Diese Methode ist bereits ab zwei gleichen Bytes sinnvoll. Unterscheiden sich jedoch einige Bytes voneinander, so werden als erstes die Anzahl der folgenden Datenbytes und dann die Datenbytes gespeichert. Um nun unterscheiden zu können, ob es sich bei einem Byte um die Anzahl von gepackten Bytes, von Einzelbytes oder sogar um ein Datenbyte handelt, geht man beim Unpacken folgendermaßen vor: Ist das erste Byte der Grafik kleiner als 128, so handelt es sich um die Anzahl der folgenden Einzelbytes. Ist es größer als 128, so liegt die Anzahl des folgenden gepackten Bytes vor. Die eigentliche Anzahl errechnet sich dann, indem man das Anzahlbyte von 257 abzieht. Hat man so ermittelt, um welche Art von Anzahl es sich bei dem Wert n handelt, so weiß man auch, ob nun n Datenbytes folgen, oder ob das nächste Datenbyte n-mal vorkommt. Es ist nur noch zu beachten, daß rechtzeitig nach 40 bzw. 80 Datenbytes (je nach Auflösung von 320 oder 640 Pixeln/Zeile) die

Bitplane gewechselt werden muß! Eigentlich ist das Unpacken nicht schwer. Viel haariger wird es, wenn es ans Packen geht.

Der Packer (Listing 3)

Hier treten wir auf unvorhergesehene Probleme mathematischer und struktureller Art. Da die Vorgehensweise des Packers das genaue Gegenteil des Unpackers ist, möchte ich mich auf diese Probleme beschränken. Es genügt zum Beispiel längst nicht, zwei Bytes der Grafik zu vergleichen und je nachdem als Einzelbytes oder gepackt zu speichern, um dann wiederum die nächsten zwei Bytes zu vergleichen. Das dritte und vierte Byte haben ebenso das Recht, verglichen zu werden. Deshalb darf nach jedem Vergleich nur das erste Byte abgespeichert werden. Der nächste Vergleich beginnt dann mit dem zweiten Byte des letzten Vergleiches. Eine weitere Schwierigkeit ist, festzustellen, wann das Ende der Grafik erreicht ist. Es ist viel zu kompliziert, die Bytes mitzuzählen, deshalb muß man auf das Zählen der Bitplane-wechsel ausweichen: Das Ende der Grafik ist nach

Höhe*PlaneAnz

Bitplane-Wechseln erreicht, wobei Höhe die Höhe der Grafik in Zeilen ist. Doch der größte Teil der Aufgabe ist noch nicht gelöst, denn jetzt geht es erst richtig los.

Einbinden in BASIC

Wir wollen die Routinen von BASIC aus aufrufen. Beim Amiga lautet die Devise "Erst anmelden, dann belegen". Was heißt das? Der Amiga ist bekanntlich multitaskingfähig, er kann also mehrere Programme "gleichzeitig" bearbeiten. Damit sich diese Programme nicht gegenseitig stören oder gar löschen, muß er die Speicherverwaltung selbst übernehmen. Denn woher soll der Programmierer wissen, wo er Speicher belegen darf. Er muß also seinen Amiga um Speicherplatz "bitten". Ist er vorhanden, so bekommt er ihn auch reserviert. Doch dieser Speicherbereich liegt irgendwo im RAM und nicht an der Stelle, wo das Programm, das dort gespeichert werden soll, vorher gelegen hat. Konkret bedeutet

das, daß die Programme relocatable sein müssen. Relocatable bedeutet "frei verschiebbar" und hat auf das Programm allerlei Auswirkungen. So dürfen zum Beispiel innerhalb des Programms keine Verzweigungen an absolute Adressen erfolgen, da dies unweigerlich zu einem Absturz führen würde. Denn wenn ein Programm bei Adresse \$2f000 nach \$2f100 verzweigt, muß es, wenn es bei Adresse \$31000 liegt, nach \$31100 verzweigen und nicht nach \$2f100 wie vorher. Die Anpassung der Sprungbefehle und Labels übernimmt im Normalfall das DOS. Wenn wir jedoch ein Assemblerprogramm in BASIC einbinden, hilft uns das DOS nicht. Wir müssen uns selber helfen, indem wir absolute Adressierungen vermeiden. Doch das ist leichter gesagt als getan. Die Sprungbefehle lassen sich sehr einfach anpassen: Man verwendet anstatt des Jump-Befehls (wie `jsr <label>`) den Branch-Befehl (`bsr <label>`). Dies gilt natürlich auch für alle anderen Verzweigungen. Bei den Labels wird es da schon sehr viel komplizierter. So sind zum Beispiel nur Labels erlaubt, die Sprungmarken definieren. Solche, die Variablen bezeichnen, müssen ersetzt werden. Der erfahrenere Programmierer denkt sofort an die Adressierungsart "Programmzähler indirekt", wird aber damit nicht weit kommen. Diese Adressierung erlaubt nämlich nur das Einlesen einer Variable mit (siehe Box).

Aus diesem Grund kann man so lediglich Konstanten wie "dosname" und "gfxname" oder ähnliche ansprechen, da auf sie nur ein lesender Zugriff erfolgt. Wollen wir dennoch Variablen benutzen, so müssen wir uns erst einmal Speicher reservieren, was über die Exec-Funktion AllocMem geschieht. Die Anfangsadresse des Speichers verwenden wir, um die Variablen mit der Adressierungsart "Adressregister indirekt mit 16 Bit Distanzwert" anzusprechen. In Listing 3 wird diese Vorgehensweise bei der Speicherung der Eingangsparameter deutlich: Im Adressregister a0 liegt die Adresse des Speichers, der für die Aufnahme der Variablen vorgesehen ist. Die einzelnen Variablen, jeweils Langwörter, sind mit entsprechendem Offset ab

dieser Adresse zu finden. So liegt zum Beispiel die Höhe der Grafik bei 4(a0), also an der Adresse des Speichers plus vier. Ebenso sieht es bei den anderen Variablen aus.

Um die Verständlichkeit der Listings zu verbessern, stehen die eigentlichen Labels der Variablen im Kommentar. Haben wir alle obengenannten Regeln beachtet, so ist unser Assembler-Programm für die Einbindung in BASIC gerüstet. Es fehlen nur noch die erste und vorletzte Zeile des Hauptprogramms. Sie dient der "Verträglichkeit" der beiden Sprachen. Zu diesem Zweck werden einfach die Register a0, d0 und d1 auf dem Stack zwischengespeichert, um am Programmende einen einwandfreien Rücksprung ins BASIC zu garantieren. Doch sehen wir uns einmal die BASIC-Listings an.

Wie man ein Assembler-Programm nachläßt, ist im BASIC-Handbuch ausführlich beschrieben. Was dort aber nicht steht ist, daß das Assembler-Programm unbedingt an einer geraden Adresse beginnen muß. Da man davon nicht einfach ausgehen kann, ist es unbedingt notwendig, die angegebene Datelänge so lange um eins zu erhöhen und den Ladevorgang zu wiederholen, bis das Programm an einer geraden Adresse beginnt. Alles weitere steht im Handbuch. Die Parameterübergabe sieht folgen-

dermaßen aus: Alle Parameter befinden sich nach der Übergabe auf dem Stack. Der erste Parameter steht dort auch an erster Stelle. Allerdings müssen wir berücksichtigen, daß wir am Schluß auch noch die drei Register auf den Stack abgelegt haben, so daß sich der Offset für den ersten Parameter auf 16(sp) ändert. Übergeben werden können nur Zahlen beziehungsweise Adressen. Es ist sinnvoll, nur Langwörter zu übergeben, damit man nicht mit den Offsets durcheinander kommt.

Tabelle 1 erläutert noch einmal alle Parameter im einzelnen. Beide BASIC-Listings enthalten noch diverse Print-Befehle, die die Vorgehensweise verdeutlichen sollen. Im normalen Gebrauch können sie allerdings weggelassen.

Die Eingabe

Abschließend ist noch zu bemerken, daß die Eingabe der Assembler-Listings am geeignetsten mit dem Seka-Assembler erfolgt. Ist das Listing fehlerfrei abgetippt und gespeichert, muß es assembliert werden. Danach wird es mit wi (WriteImage) als Maschinen-Code erneut gespeichert. Auf die Frage des Seka-Assemblers nach Start und End sind einfach die Labels "run" und "ende" einzugeben, dann ist die Eingabe komplett. In das BASIC-Programm muß schließlich

noch der Dateiname mit dem entsprechenden Pfad eingesetzt werden, und auf der Diskette muß sich die Datei "Intuition.Bmap" befinden, damit die Intuition-Library von BASIC aus geöffnet werden kann. Wie man die Datei erstellt, ist im BASIC-Handbuch unter der Anweisung "Library" beschrieben.

Der Unpacker und der Packer unterstützen alle Screenformate, die von BASIC aus darstellbar sind. Trifft man jedoch die nötigen Vorbereitungen, so lassen sich auch Extra-Halfbright-Bilder problemlos bearbeiten (siehe Box).

(mm)

```
move.l <label>(pc),d0
```

beispielsweise, nicht aber die umgekehrte Richtung mit

```
move.l d0,<label>(pc)
```

Eingangsparameter

Listing 1:

1. Adresse des Dateinamens
2. Adresse einer Ganzzahl-Variablen, an der die Anfangsadresse der Grafik geschrieben wird
3. siehe 2., beinhaltet die Größe des reservierten Speichers

Listing 2:

1. Anfangsadresse der Grafik
2. Breite der Grafik in Byte
3. Gesamte Größe in Byte
4. Zeiger auf Bitplane-Pointer
5. Anzahl der Bitplanes
6. Größe des reservierten Speichers

Listing 3:

1. Adresse des Dateinamens
2. Höhe der Grafik in Zeilen
3. Breite der Grafik in Pixeln
4. Anzahl der Bitplanes
5. Zeiger auf Bitplane-Pointer
6. Zeiger auf Bitmap
7. Viewmode des Screens
8. Größe des zu reservierenden Speichers, der die fertiggestellte Datei aufnimmt.

Listings

```
***** AMIGA DOS INFO *****
*      Programmname : LoadILBM      *
*      Funktion    : Laden von Bildern *
*      Sprache     : AmigaBASIC      *
*      Besonderheiten :               *
*****

100 REM maschinensprache-aufruf/ ass4.bas
101
102 LIBRARY "intuition.library"
103
104 ' ----- Routinen nachladen
105
106 flen=242
107 load1:
108
109 OPEN "df0:ilbm/examine5.2.obj" AS #1 LEN=flen
110 FIELD#1,flen AS a$
111 GET#1,1
112 examine$a$
113 CLOSE#1
114 routine1=SADD(examine$a$)
115 PRINT routine1&
116 IF routine1/2>INT(routine1/2) THEN flen=flen+1:GOTO 1
oad1
117
118 flen=286
119 load2:
120 OPEN "df0:ilbm/Unpack5.obj" AS #1 LEN=flen
121 FIELD#1,flen AS a$
122 GET#1,1
123 unpack$a$
124 CLOSE#1
125 routine2=SADD(unpack$a$)
126 PRINT routine2&
127 IF routine2/2>INT(routine2/2) THEN flen=flen+1:GOTO 1
oad2
```

```
128
129 ' Variableninitialisierung fuer Routine1
130
131 INPUT "Name des Bildes: ",fi$
132 fi$=fi$+CHR$(0)
133 spsize=2000000&
134 size=VARPTR(spsize&)
135 sp=2000000&
136 bild=VARPTR(sp&)
137 adr=SADD(fi$)
138 PRINT routine1&,routine2&
139
140
141 CALL routine1(adr,bild,size&)
142 size=PEEK(size&)
143 bild=PEEK(bild&)
144 PRINT bild&
145
146 ' ----- Auf ILBM-Kennung testen
147
148 curadr=bild&+8
149 IF PEEK(curadr&)<>1229734477& THEN
150 PRINT "Dies ist kein 'ILBM-File'!"
151 GOTO programmende
152 END IF
153
154 ' ----- Daten fuer Screen ermitteln
155
156 curadr=bild&+20
157 breite=PEEKW(curadr&)
158 hoehe=PEEKW(curadr&+2)
159 tiefe=PEEK(curadr&+8)
160 IF breite&=320 THEN modus&=1 ELSE modus&=2
161
162 IF hoehe&=400 THEN modus&=modus&+2
163 PRINT breite&,hoehe&,tiefe&
164
165 ' Screen u. Window oeffnen, Pointer ermitteln
166
167 SCREEN 2,breite&,hoehe&,tiefe&,modus&
168 WINDOW 2,,0,2
```



```

169 wndwbase%=WINDOW(7)
170 scrnbase%=PEEK(wndwbase+46)
171 scrnmap=scrnbase+184
172 scrnpnt=scrnbase+84
173 scrnpvt=scrnbase+44
174 scrnmode%=PEEKW(scrnpvt+32)
175 btplpnt=scrnmap+8
176 WINDOW OUTPUT 1
177 PRINT btplpnt
178 WINDOW OUTPUT 2
179 bytebreit=breite/8
180 bytes=bytebreit*hoehe*tiefe
181
182 ' Interlace nötig ?
183
184 IF hoehe=>400 THEN GOSUB setinterl
185
186 ' ----- CMAP-Chunk suchen
187 flag=0:i=1
188 WHILE (flag=0 AND i<400)
189   i=i+1
190   IF (bild+i)/2<>INT((bild+i)/2) THEN weiter
191     a%=PEEK(bild+i)
192     IF a%=1129136464 THEN flag=1
193   weiter:
194 WEND
195
196 ' Farben ermitteln und setzen
197
198 cmap=bild+i+4
199 farbanz%=PEEK(cmap)/3
200 cmap=cmap+4
201 FOR i=0 TO farbanz%-1
202   rot=PEEK(cmap+i*3)/(16*15)
203   gruen=PEEK(cmap+i*3+1)/(16*15)
204   blau=PEEK(cmap+i*3+2)/(16*15)
205   PALETTE i,rot,gruen,blau
206 NEXT
207
208 ' Unpack-Routine aufrufen
209
210 CALL routine2(bild,bytebreit,bytes,btplpnt,tiefe,
size)
211
212 programmende:
213   LIBRARY CLOSE
214   ende:
215   a$=INKEY$
216   IF a$="" THEN ende
217   SCREEN CLOSE 2
218 END
219
220 setinterl:
221   POKEW(scrnpvt+32),scrnmode% OR 4
222   CALL RemakeDisplay
223 RETURN
224

```

```

*****
*          AMIGA DOS INFO          *
*  Programmname : SaveILBM3.1.Source *
*  Funktion     : Quelltext der Routine *
*  Sprache      : SEKA-Assembler      *
*  Besonderheiten : -----          *
*****

```

```

100 Listing : SaveILBM3.1.Source
101
102
103 openlib=-408
104 closelibrary=-414
105 open=-30
106 close=-36
107 write=-48
108 execbase=4
109 allocmem=-198
110 freemem=-210
111
112
113 ;----- Saveilbm -----
114
115 run:
116   movem.l a0/d0-d1,-(sp)      ;Register retten
117
118   move.l execbase,a6          ;Speicher für
119   move.l #220,d0              ;Variablen
120   move.l #$10002,d1           ;reservieren
121   jsr allocmem(a6)
122   move.l d0,a0                ;Basis in a0
123   move.l a0,a5                ;und a5
124   beq ende3
125
126   bsr opengraph
127
128 ; ----- Eingangsparameter sichern
129
130   move.l 16(sp),(a0)          ;filename
131   move.l 20(sp),4(a0)         ;höhe

```

```

132   move.l 24(sp),8(a0)         ;breite
133   move.l 28(sp),12(a0)        ;tiefe
134   move.l 32(sp),16(a0)        ;bitplanes
135   move.l 36(sp),20(a0)        ;colors
136   move.l 40(sp),24(a0)        ;view_modes
137   move.l 44(sp),28(a0)        ;memsize
138
139   move.l ExecBase,a6
140   move.l 28(a0),d0             ;memsize
141   move.l #$10000,d1
142   jsr AllocMem(a6)
143   move.l a5,a0
144   move.l d0,32(a0)            ;memory1
145   beq ende2
146
147 ; hole und bytewidth errechnen
148
149   move.l 4(a0),d1             ;höhe
150   move.l 12(a0),d2            ;tiefe
151   mulu d1,d2
152   move.l d2,36(a0)            ;hole
153   move.l 8(a0),d1             ;breite
154   divu #8,d1
155   move.l d1,72(a0)            ;bytewidth
156
157 ; Chunks erstellen
158
159   move.l 32(a0),a4             ;memory1
160   move.l #"FORM",8(a4)
161   move.l #"ILBM",8(a4)
162   move.l #"BMHD",12(a4)
163   move.l #$14,16(a4)          ;Länge von BMHD
164   move.w 10(a0),20(a4)        ;breite
165   move.w 6(a0),22(a4)         ;hoehe
166   move.l #0,24(a4)            ;Position auf BS
167   move.w 14(a0),d1            ;tiefe
168   move.b d1,28(a4)
169   move.b #0,29(a4)            ;kein Masking
170   move.b #1,30(a4)            ;gepackt
171   move.b #0,31(a4)            ;keine Bedeutung
172   move.w #0,32(a4)            ;kein Transparent
173   move.w #$101,34(a4)         ;Verhältnis
174   move.w 10(a0),36(a4)        ;Screensize, breite
175   move.w 6(a0),38(a4)         ;höhe
176   move.l #"CMAP",40(a4)
177
178   move.l 12(a0),d1            ;tiefe
179   moveq #1,d0
180   lsl d1,d0
181   mulu #3,d0
182   move.l d0,44(a4)            ;Länge von CMAP
183
184
185   move.l #"CAMG",48(a4,d0)
186   move.l #4,52(a4,d0)
187   move.l 24(a0),d1            ;view_modes
188   move.l d1,56(a4,d0)
189   or.l #$4000,56(a4,d0)
190   move.l d0,76(a0)            ;farben
191
192   move.l #"CRNG",60(a4,d0)
193   move.l #8,64(a4,d0)
194   move.l #0,68(a4,d0)
195   move.l #0,72(a4,d0)
196   move.l #"CRNG",76(a4,d0)
197   move.l #8,80(a4,d0)
198   move.l #0,84(a4,d0)
199   move.l #0,88(a4,d0)
200   move.l #"CRNG",92(a4,d0)
201   move.l #8,96(a4,d0)
202   move.l #0,100(a4,d0)
203   move.l #0,104(a4,d0)
204   move.l #"CRNG",108(a4,d0)
205   move.l #8,112(a4,d0)
206   move.l #0,116(a4,d0)
207   move.l #0,120(a4,d0)
208
209   move.l #"BODY",124(a4,d0)
210   move.l a4,a3                ;Start von BODY
211   add.l #132,a3
212   add.l d0,a3
213
214 ; ----- Pack-Routine
215   bsr calcbitplanes
216   lea 48(a0),a5               ;adrbase
217   move.l (a5),a1              ;Zeiger auf Bitplane
218   moveq #0,d0                 ;Register initialisie
ren
219   moveq #0,d1
220   moveq #0,d2
221   moveq #0,d3
222   moveq #0,d4
223   moveq #1,d5                 ;1.Zeile/1.Bitplane
224   moveq #1,d6                 ;aktuelle Bitlane: 0
225   moveq #0,d7
226
227 schleife1:
228   move.l a3,a2                ;Anzahl-save
229   move.b (a1)+,d0             ;2 Datenbytes
230   move.b (a1)+,d1
231   moveq #2,d3
232   addq.l #2,d4
233   cmp.b d0,d1                 ;2 gleiche Bytes?
234   bne.s einzel2              ;Nein
235   continue1:
236   move.l 72(a0),d7            ;bytewidth
237   cmp.l d4,d7                ;Planewechsel nötig?

```



```

238 bgt.s schleife2 ;Nein, dann schleife2
239 move.l #257,d2 ;Anzahl
240 sub.l d3,d2 ;errechnen
241 move.b d2,(a3)+ ;Anzahl speichern
242 move.b d0,(a3)+ ;Bytewert speichern
243 bra changebtp1
244
245 schleife2: ;wieviele folgen
246 cmp.b (a1),d0 ;nächstes gleich?
247 bne.s continue2 ;Nein
248 addq.l #1,a1
249 addq.l #1,d3
250 addq.l #1,d4
251 bra.s continue1
252 continue2: ;nächstes Byte
253 move.l #257,d2 ;verschieden,
254 sub.l d3,d2 ;deshalb speichern
255 move.b d2,(a3)+ ;s.o.
256 move.b d0,(a3)+
257 moveq #0,d3
258 bra schleife1 ;weiter bei schleife1
259
260 einzel2:
261 subq.l #1,a1
262 moveq #1,d3
263 subq.l #1,d4
264 addq.l #1,a3 ;Byte für Anzahl frei
265 einzel3:
266 move.b d0,(a3)+
267 move.l 72(a0),d7 ;bytewidth
268 cmp.l d4,d7 ;Planewechsel nötig?
269 bls.s continue3 ;Ja, dann continue3
270
271 subq.l #1,d3 ;Zwischensave Anzahl
272 move.b d3,(a2)
273 addq.l #1,d3
274
275 move.b (a1)+,d0 ;2 Datenbytes
276 move.b (a1),d1
277 addq.l #1,d3
278 addq.l #1,d4
279 cmp.b d0,d1 ;2 gleiche Bytes?
280 bne.s einzel3 ;Nein
281 moveq #1,d3
282 bra continue1
283
284 continue3:
285 subq.l #1,d3
286 move.b d3,(a2)
287 addq.l #1,d3 ;>>>> ???
288 bra changebtp1 ;Bitplane-Wechsel
289
290
291
292 endel:
293 move.l 76(a0),d0 ;farben
294 move.l a4,a5
295 add.l d0,a5
296 add.l #132,a5
297 move.l a3,a1
298 sub.l a5,a1
299 move.l a4,a5
300 add.l d0,a5
301 add.l #128,a5
302 move.l a1,(a5) ;Länge d. BODY-Chunks
303
304 move.l a3,a1
305 move.l a4,a5
306 sub.l a5,a1
307 subq.l #8,a1
308 move.l a1,4(a4) ;File-Länge
309
310 move.l 76(a0),d0 ;farben
311 divu #3,d0
312 move.l d0,d4 ;anz. d. Farben
313 moveq #0,d5
314 move.l a0,a5 ;varbase retten
315 lea 80(a5),a2 ;cmap
316
317 do:
318 move.l 20(a5),a0 ;Zeiger auf Bitmap
319 move.l 180(a5),a6 ;gfxbase
320 move.l d5,d0
321 jsr -582(a6) ;getcolormap
322 moveq #0,d1
323 moveq #0,d2
324 lsl #4,d0 ;4 Bit links
325 or.b d0,d2 ;1. Byte holen
326 and.b #0,d0 ;löschen
327 lsr #4,d0 ;4 Bit rechts
328 or.b d0,d1 ;2. Byte holen
329 and.b #0,d0
330 lsr #4,d0
331
332 move.b d0,(a2)+
333 move.b d1,(a2)+
334 move.b d2,(a2)+
335 addq.l #1,d5
336 cmp.l d4,d5
337 bne.s do
338
339 mulu #3,d4
340 subq.l #1,d4
341 lea 80(a5),a1 ;cmap
342 move.l a4,a2
343 add.l #48,a2
344 do2:
345 move.b (a1)+,(a2)+ ;Farben kopieren

```

```

346 dbf d4,d02
347
348 bsr opendos
349 move.l 40(a5),a6 ;dosbase
350 move.l (a5),d1 ;filename
351 move.l #1006,d2
352 jsr open(a6)
353 move.l d0,44(a5) ;filehd
354
355 move.l 40(a5),a6 ;dosbase
356 move.l 44(a5),d1 ;filehd
357 move.l a4,d2
358 move.l a3,d3
359 sub.l d2,d3
360 jsr write(a6)
361
362 move.l 40(a5),a6 ;dosbase
363 move.l 44(a5),d1 ;filehd
364 jsr close(a6)
365
366 bsr closedos
367 bsr closegraph
368 cont:
369 move.l execbase,a6
370 move.l 32(a5),a1 ;memory
371 move.l 28(a5),d1 ;memsize
372 jsr freemem(a6)
373 ende2:
374 move.l execbase,a6
375 move.l a5,a1
376 move.l #220,d0
377 jsr freemem(a6)
378 ende3:
379 movem.l (sp)+,a0/d0-d1
380 rts
381
382
383 changebtp1:
384 move.l d6,d0
385 move.l d6,d4
386 subq #1,d4
387 mulu #4,d4
388 lea 48(a0),a5 ;adrbase
389 move.l a1,(a5,d4)
390 addq.l #1,d6
391 move.w 14(a0),d7 ;tiefe
392 addq #1,d7
393 cmp.l d6,d7
394 bne.s go_on9
395 move.l #1,d6
396 go_on9:
397 move.l d6,d4
398 subq #1,d4
399 lsl #2,d4
400 lea 48(a0),a5 ;adrbase
401 move.l (a5,d4),a1
402 moveq #0,d4
403 move.l 36(a0),d7 ;hole
404 cmp.l d5,d7 ;ende erreicht?
405 beq endel
406 addq.l #1,d5 ;nächste Zeile
407 bra schleife1
408
409 calcbitplanes:
410 move.l 16(a0),a1 ;bitplanes
411 lea 48(a0),a2 ;adrbase
412 move.l (a1),(a2)+
413 move.l 4(a1),(a2)+
414 move.l 8(a1),(a2)+
415 move.l 12(a1),(a2)+
416 move.l 16(a1),(a2)+
417 move.l 20(a1),(a2)+
418 rts
419
420 ;----- Routinen -----
421
422 opendos:
423 move.l execbase,a6
424 lea dosname(pc),a1
425 jsr openlib(a6)
426 move.l d0,40(a5) ;dosbase
427 rts
428
429 closedos:
430 move.l execbase,a6
431 move.l 40(a5),a1 ;dosbase
432 jsr closelibrary(a6)
433 rts
434
435 opengraph:
436 move.l execbase,a6
437 move.l a0,a5
438 lea gfxname(pc),a1
439 jsr openlib(a6)
440 move.l d0,180(a5) ;gfxbase
441 move.l a5,a0
442 rts
443
444 closegraph:
445 move.l execbase,a6
446 move.l 180(a5),a1 ;gfxbase
447 jsr closelibrary(a6)
448 rts
449
450 dosname: dc.b "dos.library",0
451 gfxname: dc.b "graphics.library",0
452
453 ende:

```



```
*****
*          AMIGA DOS INFO
*  Programmname : SaveILBM
*  Funktion      : Speichern von Bildern
*  Sprache       : BASIC
*  Besonderheiten :
*****

100 REM SaveILBM
101
102 DIM a$(1100)
103 flen=1037
104
105 OPEN "dfo:ilbm/saveilbm3.1.obj" AS #1 LEN=flen
106 FIELD#1,flen AS a$(1)
107 GET#1,1
108 ilbm$=a$(1)
109 CLOSE #1
110
111 routine3=SADD(ilbm$)
112 PRINT routine3
113 INPUT "Filename: ",filename$
114 filename$=filename$+CHR$(0):file=SADD(filename$)
115
116 SCREEN 2,640,256,4,2
117 WINDOW 2,,,0,2
118
119 LINE(10,10)-(300,100),,bf
120 CIRCLE (320,100),80,3
121 PAINT (320,100),3
122 AREA (50,150)
123 AREA (30,100)
124 AREA (170,130)
125 AREA FILL
126
127 hoehe=256
128 breite=640
129 tiefe=4
130 wndwbase=WINDOW(7):scrnbase=PEEKL(wndwbase+46)
131 scrnbase=scrnbase+184:bitplanes=scrnbase+8
132 scrnvprt=scrnbase+44:viewmodes=PEEKW(scrnvprt+32)
133 colors=PEEKL(scrnbase+48)
134 memsize=40000&
135
136 CALL routine3(file,hoehe,breite,tiefe,bitplanes,colors,viewmodes,memsize)
137 SCREEN CLOSE 2
138
139 END
```

```
*****
*          AMIGA DOS INFO
*  Programmname : Unpack5.Source
*  Funktion      : Entpacker von Dateien
*  Sprache       : SEKA-Assembler
*  Besonderheiten :
*****

100 Listing : Unpack5.Source
101
102
103 Execbase=4
104 AllocMem=-198
105 FreeMem=-210
106
107 run:
108     movem.l a0/d0-d1,-(sp)
109
110     move.l Execbase,a6           ;Speicher für Variablen
111     move.l #44,d0               ;reservieren
112     move.l #$10000,d1           ;Mem_Clear
113     jsr AllocMem(a6)
114     move.l d0,a4                ;Basis d. Variablen
115     beq return
116
117 ; Eingangparameter sichern
118
119     move.l 16(sp),a4             ;memory
120     move.l 20(sp),8(a4)         ;Breite in Bytes: anzahl
121     move.l 24(sp),4(a4)         ;höhe*breite*tiefe: 1
122     move.l 28(sp),a5            ;Bitplanes
123     move.l 32(sp),12(a4)        ;Anz. d. Bitplanes: planes
124     move.l 36(sp),40(a4)        ;Memsize für File
125
126     lea 16(a4),a1               ;bitplns kopieren
127     move.l 12(a4),d0           ;planes
128     subq.l #1,d0               ;wegen dbf
129 loop1:
130     move.l (a5)+(a1)+          ;kopieren
131     dbf d0,loop1
132
133     lea 16(a4),a5              ;bitplns
134
135 ; ----- nach 'BODY' suchen
136
137     move.l (a4),a3             ;memory
138 loop2:
139     cmp.b #"B",(a3)+
140     bne.s loop2
141     cmp.b #"O",(a3)+
142     bne.s loop2
```

```
143     cmp.b #"D",(a3)+
144     bne.s loop2
145     cmp.b #"Y",(a3)+
146     bne.s loop2
147
148     move.l (a3)+,d2             ;Anzahl der Bytes in d2
149
150     move.l (a5),a1             ;Bitplane0 Zeiger
151     moveq #0,d5                ;Gesamtlänge
152     moveq #1,d6                ;aktuelle Bitplane ist 0
153     moveq #0,d4                ;Anzahl
154
155 ; ----- Unpacken
156
157 loop3:
158     moveq #0,d3                ;Dekodier-Routine
159     move.b (a3)+,d3            ;Datenbyte holen
160     cmp.l #128,d3              ;<=128?
161     bts.s einzel               ;Ja, Einzelbyte
162
163     move.l #257,d1
164     sub.l d3,d1                ;d1: Anzahl der folg. Bytes
165     subq.l #1,d1               ;wegen dbf
166
167
168 loop4:
169     move.b (a3),(a1)+
170     addq.l #1,d4
171     cmp.l 8(a4),d4             ;anzahl erreicht?
172     bne.s go_on1
173     bsr.s chbitplane           ;Ja, Plane wechseln
174 go_on1:
175     addq.l #1,d5
176     cmp.l 4(a4),d5             ;länge erreicht?
177     beq.s return              ;Ja, PRG beenden
178     dbf d1,loop4              ;nächstes Byte ausgeben
179
180     addq.l #1,a3
181 weiter:
182     dbf d2,loop3              ;nächstes Datenbyte
183
184 einzel:
185     move.b (a3)+(a1)+
186     addq.l #1,d4
187     cmp.l 8(a4),d4             ;anzahl erreicht?
188     bne.s go_on2
189     bsr.s chbitplane           ;Ja, Plane wechseln
190 go_on2:
191     addq.l #1,d5
192     cmp.l 4(a4),d5             ;länge erreicht?
193     beq.s return              ;Ja, PRG beenden
194     dbf d3,einzel             ;nächstes Einzelbyte
195     bra weiter
196
197
198 chbitplane:
199     move.l d6,d4               ;bestimmt die neue
200     subq.l #1,d4               ;Adresse der neuen
201     lsl #2,d4                  ;Bitplane
202     move.l a1,(a5,d4)          ;PlaneNr*4+a5=Planeadr
203
204     addq.l #1,d6
205     move.l 12(a4),d7           ;planes
206     addq.l #1,d7
207     cmp.l d6,d7                ;Bitplane=maxBitplane?
208     bne.s go_on3              ;Nein, weiter
209     move.l #1,d6               ;Ja, Bitplane=1
210 go_on3:
211     move.l d6,d4               ;NeuePlaneNr*4+a5=
212     subq.l #1,d4               ;Planeadr
213     lsl #2,d4
214     move.l (a5,d4),a1          ;Zähler v. neuer Plane
215     moveq #0,d4                ;holen, d4 wieder 0
216     rts
217
218 return:
219     move.l execbase,a6         ;Speicher
220     move.l (a4),a1             ;für File
221     move.l 40(a4),d0           ;wieder
222     jsr FreeMem(a6)            ;freigeben
223
224     move.l execbase,a6         ;Speicher
225     move.l a4,a1               ;für Variablen
226     move.l #44,d0              ;wieder
227     jsr FreeMem(a6)            ;freigeben
228
229     movem.l (sp)+,a0/d0-d1     ;Register zurück
230     rts
231 ende:
```

```
*****
*          AMIGA DOS INFO
*  Programmname : Examine5.1.Source
*  Funktion      : Lader
*  Sprache       : SEKA-Assembler
*  Besonderheiten :
*****
```



```

100 Listing :      Examine5.1.Source
101
102
103 execbase=4          ;Die Variablen werden
104 openlib=-408        ;über a4 indirekt mit
105 closelibrary=-414   ;Offset adressiert !!
106 lock=-84
107 examine=-102
108 allocmem=-198
109 open=-30
110 close=-36
111 read=-42
112
113 run:
114     movem.l a0/d0-d1,-(sp)      ;Register retten
115
116     move.l  execbase,a6         ;Speicher für
117     move.l  #$10002,d1         ;Variablen
118     move.l  #300,d0            ;anfordern
119     jsr     allocmem(a6)
120     move.l  d0,a4              ;Basis in a4
121     beq     fehler
122
123 ; Übergangsparameter retten
124
125     move.l  16(sp),20(a4)       ;filename
126     move.l  20(sp),8(a4)       ;speicher
127
128     bsr     opendos            ;File-Länge ermitteln
129     move.l  (a4),a6            ;dosbase
130     move.l  20(a4),d1          ;filename
131     move.l  #-2,d2             ;Access_Read
132     jsr     lock(a6)
133     move.l  d0,d1              ;Zeiger auf Lock
134     move.l  (a4),a6            ;dosbase
135     move.l  a4,d2              ;#infoblock
136     add.l   #24,d2             ;#infoblock
137     jsr     examine(a6)
138     lea     24(a4),a0          ;#infoblock
139     move.l  124(a0),d0         ;Länge des Files
140     move.l  d0,d4
141
142     move.l  execbase,a6         ;Speicher reservieren
143     move.l  #$10002,d1         ;Chpmem + Clear
144     jsr     allocmem(a6)
145     move.l  d0,4(a4)           ;memory
146     beq     fehler            ;nicht verfügbar
147

```

```

148     move.l  24(sp),a0          ;size
149     move.l  d4,(a0)            ;Länge in Basic-Var.
150     move.l  8(a4),a0           ;speicher
151     move.l  4(a4),(a0)         ;memory in Basic-Var.
152
153     lea     24(a4),a0          ;#infoblock
154     move.l  124(a0),16(a4)     ;laenge
155
156 ; File laden
157
158     move.l  (a4),a6            ;dosbase
159     move.l  20(a4),d1          ;filename
160     move.l  #1005,d2           ;Mode_old
161     jsr     open(a6)
162     move.l  d0,12(a4)          ;filehd
163     beq     fehler            ;File not found?
164     move.l  (a4),a6            ;dosbase
165     move.l  d0,d1              ;filehd nach d1
166     move.l  4(a4),d2           ;memory nach d2
167     move.l  16(a4),d3          ;laenge nach d3
168     jsr     read(a6)           ;lesen
169
170     move.l  (a4),a6            ;dosbase
171     move.l  12(a4),d1          ;filehd
172     jsr     close(a6)
173
174 ; ----- Rücksprung
175 fehler:
176     bsr     closedos
177     movem.l (sp)+,a0/d0-d1     ;Register zurück
178     rts
179
180 ; -----
181 opendos:
182     move.l  execbase,a6
183     lea     dosname(pc),a1
184     jsr     openlib(a6)
185     move.l  d0,(a4)            ;dosbase
186     rts
187
188 closedos:
189     move.l  execbase,a6
190     move.l  (a4),a1            ;dosbase
191     jsr     closelibrary(a6)
192     rts
193
194 dosname:  "dos.library",0
195 ende:

```

WETTERBILDEMPFANG MIT AMIGA

FAX-Signale, die von umlaufenden Satelliten und vom Meteosat sowie auf Langwelle und Kurzwelle von Wetterdiensten, Pressediensten, Behörden und Funkamateuren ausgestrahlt werden, können ausgewertet werden. Verbinden Sie Ihren Empfänger und Ihren Amiga mit unserem Interface.

Sie besitzen keinen Empfänger? Unser Langwellenempfänger mit eingebautem Interface eröffnet Ihnen die FAX-Welt. Sie installieren lediglich eine Langdrahtantenne. Geräte fertig aufgebaut, im Gehäuse, anschlussfertig mit Steckernetzteil

Preise: Interface, Programm, Kabel 345,- DM
Interface, LW-Empfänger, Programm 585,- DM
Info: Gratis und unverbindlich
Demodisk: bitte großen Rückumschlag, mit 2,40 DM frankieren, DM 5,- (auch in Briefmarken möglich), beilegen.

Karl J. Ebensberger, Fällhofstr. 11, 8068 Pfaffenhofen, Tel. 084 41/61 45, BTX und FAX: 0844172213

Bitte beachten Sie bei Inbetriebnahme des LW-Empfängers die Postvorschriften!

512k RAM für A500 – abschaltbar – mit Akku-Uhr – Einbau intern 219,00	2MB RAM für A2000 – bis zu 8MB – Autokonfig – 0 Waitstates 999,00	ANIMATE-TURBO-BOARD III – MIT MC68020 UND MC68881, 16MHZ – BOARD MIT MC68882 20MHZ AB LAGER – BESTÜCKBAR MIT 32BIT-RAM 256K-1MB – 32BIT BUS HERAUSGEFÜHRT ab 1450,-
3,5" DRIVE EXTERN – AMIGAFARBEN – BUSDURCHFÜHRUNG – ABSCHALTBAR – LANGES KABEL – NEC 1037A 219,00	HURRICANE-BOARDS – MIT 68020, 68881, 16MHZ – UMSCHALTBAR AUF 68000 – SCHNELL SIEHE TEST 8.89 – ALLE BOARDS IM PROGRAMM ab 1798,-	** AKTUELL ** AKTUELL ** ANIMATE-TURBO-BOARD-IIA – MIT MC68020 UND MC68881, 16MHZ – BOARD MIT MC68882, 20MHZ, AB LAGER – FÜR ALLE AMIGAS – UMSCHALTBAR ab 849,-
HARMS * SCHOMBURG COMPUTER-SYSTEME		ANNA-SEGHERS-STR. 99 2800 BREMEN 61 Tel. 04 21/83 38 64 Tel. 04 21/38 86 67

Dipl. Phys. Dr. Peter Kittel

BASIC in allen Gassen

BASIC ist wohl die am weitesten verbreitete Programmiersprache mit oft verkannter Leistungsfähigkeit. Und AmigaBASIC macht hier keine Ausnahme. Um es noch breiter einsetzen zu können, wird hier gezeigt, wie man AmigaBASIC-Programme auf andere Rechner portieren kann.

Die Sprache BASIC wird von arrivierten Fachleuten gern mit Naserümpfen übergangen, sie hat einen schlechten Ruf. Dabei wird meist übersehen, daß andere Sprachen genauso oder noch schlimmer zu schlechtem Programmierstil verführen, man denke nur an C oder gar Assembler.

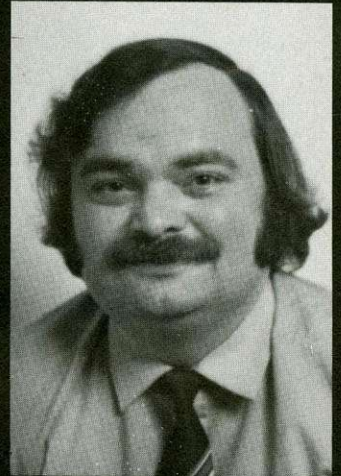
BASIC hat aber auf der anderen Seite einen ganz entscheidenden Vorteil: Es ist in der Regel kostenlos, es liegt jedenfalls allen Commodore-Computern kostenlos bei oder ist in ihre ROMs eingebaut. Es ist auch so leicht zu erlernen oder wenigstens zu verstehen, daß jedermann aus BASIC-Programmen Lehren ziehen kann, die er später immer noch auch in anderen Sprachen weiterverwenden kann. Wenn wir uns hier in einer Amiga-Zeitschrift befinden, muß die Rede natürlich vor allem vom AmigaBASIC sein. Es ist nicht als Geschwindigkeitsweltmeister berüchtigt, hat aber so viele fortschrittliche Eigenschaften, daß man es schnell schätzen lernt. In diesem Artikel soll aber auch die Welt um uns herum beachtet werden, und zwar unter dem Gesichtspunkt, wie man AmigaBASIC-Programme auch auf anderen Rechnern, vor allem PCs, zum Laufen bringen kann. AmigaBASIC kommt aus dem Hause Microsoft, genauso wie die Interpreter, die in älteren Commodore-Rechnern eingebaut und bei MS-DOS-PCs mitgeliefert werden. Der Dialekt ist damit schon einmal sehr verwandt, über die kleinen Unterschiede soll hier ausführlich berichtet werden. In die Betrachtung miteinbeziehen wollen wir das QuickBasic

von Microsoft. Dieser Compiler für PCs gehört zwar bei keinem Rechner zum Lieferumfang, ist aber ein interessantes Produkt.

Es gibt nun mehrere Fälle zu unterscheiden: Soll ein Programm vom Amiga weg oder zu ihm hin übertragen werden? Ist das andere BASIC das eines C-64, eines CBM, der PC-GW-BASIC-Interpreter oder

Zum Autor:

Dr. Peter Kittel, aufgewachsen in Hamburg, Studium der Physik an der TU Braunschweig, programmiert dort seit 1970, seit 1979 Benutzer von Commodore-Geräten, angefangen mit einem PET 2001, nach Promotion Anfang 1985 direkt zu Commodore Frankfurt in den Sales Support, dort heute dienstlich mit Dokumentation befaßt, hobbymäßig eingefleischter Amiga-Fanatiker mit bevorzugter Programmiersprache AmigaBASIC.



C64-Avenue

Amiga-Platz



gar die Compilerversion, das QuickBasic? Die erste zu bedenkende Stufe ist die rein technische. Kann ich mein Programm, zum Beispiel in ASCII-Form, direkt zu dem anderen Rechner hinüberbringen, oder muß ich es dort womöglich komplett neu eintippen?

Der Weg über Disketten ist meist mühselig wegen der verschiedenen Formate. Allerdings kann hier in einigen Fällen durch spezielle Hard- und/oder Software schnell geholfen werden. Der Austausch mit MS-DOS kann am elegantesten über eine Brückenkarte (XT oder AT) in einem Amiga 2000 erfolgen. Steht diese nicht zur Verfügung, gibt es Public-Domain-Software oder auch kommerzielle Produkte (wie Dos2Dos), die MS-DOS-kompatible Disketten auch direkt am Amiga beschreiben und lesen können; man muß dann nur noch die Diskettengröße

einhalten. Genauso ermöglichen es die gängigen C-64-Emulatoren, über ein einfaches Adapterkabel und entsprechende Treibersoftware die 1541-Floppy direkt am Amiga zu betreiben. Und da manche Laufwerke der ganz alten CBM-Serie (z. B. 4040, 2031, 4031) auch zur 1541-kompatible Diskettenformate aufweisen, kann auf diesem Weg auch mit ihnen ein Austausch realisiert werden.

Wenn es mit Disketten doch nicht klappt, bleibt einem noch der Weg über die serielle Schnittstelle. Wenn man für solche meist einmaligen Vorgänge keine zu hohen Geschwindigkeitsansprüche stellt, kommt man mit einer Dreidraht- oder gar Zweidrahtverbindung aus, die man sich auch schnell selbst zusammenbasteln kann. Listing 1 zeigt ein Sendeprogramm, Listing 2 ein Empfangsprogramm. Das Dateienende wird wie bei MS-DOS durch das Steuerzeichen Ctrl-Z (ASCII 26) angezeigt und erkannt. In den Listings ist ebenfalls eingetragen, was beim C-64 anders zu schreiben ist. (Da die CBM-Rechner über keinerlei serielle Schnittstelle verfügen, müssen sie hier leider zusehen.)

Wie bekommt man nun den Programmtext in ASCII-Form

Listing 1: Sendeprogramm

```
10 INPUT "Name der zu sendenden Datei = ";FS$
20 REM "AmigaBASIC, GWBASIC, QuickBasic:"
21 OPEN FS$ FOR INPUT AS 1
22 REM "CBM, C 64:"
23 REM OPEN 1,8,3,FS$+",S,R":NS=CHR$(0)
30 REM "AmigaBASIC:"
31 OPEN "COM1:300,N,8,1" FOR OUTPUT AS 2
32 REM "GWBASIC, QuickBasic:"
33 REM OPEN "COM1:300,N,8,1,RS,CSO,DSO,CDO" AS 2
34 REM "C 64:"
35 REM OPEN 2,2,0,CHR$(6)+CHR$(0)+CHR$(0)+CHR$(0)
40 GS=""
50 REM "AmigaBASIC, GWBASIC, QuickBasic:"
51 GS=INPUT$(1,1):GC=EOF(1)
52 REM "CBM, C 64:"
53 REM GET #1,GS:GG=ST:GS=LEFT$(NS+GS,1)
60 IF GG=0 THEN PRINT #2,GS;:GOTO 40
70 PRINT #2,CHR$(26);:CLOSE 1:CLOSE 2
80 END
```

Listing 1: Das Programm zum Senden von ASCII-Dateien, darunter können natürlich auch Programme fallen

auf die Diskette? Beim AmigaBASIC und GW-BASIC braucht man an den SAVE-Befehl nur die Option "A" anzuhängen. Bei CBM und C-64 muß man mit OPEN, CMD und LIST arbeiten, siehe die dortigen Handbücher. Weitaus trickreicher muß man hier vorgehen, wenn man eine ASCII-Datei wieder in ein ausführbares Programm umwandeln will. Das würde hier zu weit führen, es muß bei einem Verweis auf einschlägige Veröffentlichungen bleiben.

Ganz zwanglos sind wir mit den Listings 1 und 2 schon bei einfachen Programmen gelandet, die unverändert oder zumindest nur wenig geändert, in mehreren Welten funktionieren. Das liegt daran, daß sie im Urdialekt von BASIC geschrieben sind, mit Zeilennummern, höchstens zweistelligen Variablennamen und einfachen GOTO-Befehlen. In AmigaBASIC könnte man dies alles wesentlich eleganter, strukturierter und, ja, länger gestalten, so daß es nachher übersichtlicher ist. Nur, ein C-64 kann dann damit rein gar nichts mehr anfangen.

Dies ist eine wichtige Erkenntnis: Je einfacher die Sprachelemente sind, die benutzt werden, desto leichter fällt die Umsetzung. Man einigt sich sozusagen auf den kleinsten gemeinsamen Nenner. (Übrigens haben die Mathematiker unter den Lesern schon einmal bemerkt, wie falsch diese Redensart vom "kleinsten gemeinsamen Nenner" ist? Dieser ist nämlich trivialerweise immer ein-

fach die Eins. Gemeint ist vielmehr der "größte gemeinsame Teiler", und die Redensart will bedauernd herausstellen, daß dieser größte Teiler in diesen besonderen Fällen leider ziemlich klein ist. So kommt es zu dieser mathematisch unkorrekten Formulierung.) Nun wäre es aber schade, wenn man die mächtigen neuen Möglichkeiten der aktuellen BASIC-Dialekte nie mehr benutzen dürfte. Sehen wir uns daher einmal die Verwandtschaften dieser Varianten an:

1. Die einfachste Version ist das BASIC von C-64 und CBM. Daß beim BASIC 4.0 oder 7.0 ein paar Befehle mehr als bei der Version 2.0 vom C-64 existieren, braucht hier nicht zu interessieren.
2. Die nächste Stufe bildet das GWBASIC für PCs. Die Fortschritte sind allerdings relativ bescheiden, lediglich die WHILE-WEND-Struktur und eine etwas freiere Formulierung des OPEN-Befehls sind dazugekommen. Die gravierendsten Unterschiede zur ersten Stufe bestehen darin, daß zwischen den Befehlswörtern und Variablennamen Leerzeichen zwingend sind und statt Cursor-Steuerzeichen innerhalb PRINT-Anweisungen nun die Befehle LOCATE und CLS zu benutzen sind.
3. Die bisher höchste Stufe bilden der QuickBasic-Compiler für PCs und das AmigaBASIC. Hier sind keine Zeilennummern mehr nötig, Blockanweisungen mit IF-ELSE-END IF sind möglich, es gibt echte Subroutinen mit lokalen Variablen und formalen Parametern. Beim Quick-

PC-Allee



Listing 2: Empfangsprogramm

```

10 INPUT "Name der zu empfangenden Datei = ";F$
20 REM "AmigaBASIC, GWBASIC, QuickBasic:
21 OPEN F$ FOR OUTPUT AS 1
22 REM "CBM, C 64:
23 REM OPEN 1,8,3,F$+","S,W":N$=CHR$(0)
30 REM "AmigaBASIC:
31 OPEN "COM1:300,N,8,1" FOR INPUT AS 2
32 REM "GWBASIC, QuickBasic:
33 REM OPEN "COM1:300,N,8,1,RS,CSO,DSO,CDO" AS 2
34 REM "C 64:
35 REM OPEN 2,2,0,CHR$(6)+CHR$(0)+CHR$(0)+CHR$(0)
40 GG$=CHR$(26)
50 REM "AmigaBASIC, GWBASIC, QuickBasic:
51 G$=INPUT$(1,2)
52 REM "CBM, C 64:
53 REM GET #2,G$:G$=LEFT$(N$+G$,1)
60 IF G$ < > GG$ THEN PRINT #1,G$;:GOTO 50
70 CLOSE 1:CLOSE 2
80 END

```

Listing 2: Das dazugehörige Empfangsprogramm zum Einlesen der ASCII-Datei

Basic geht diese Entwicklung noch etwas weiter als beim AmigaBASIC, indem die Subroutinen auch rekursiv aufgerufen werden können. Dabei sind beide Dialekte aber aufwärtskompatibel zu ihren Vorgängern geblieben, sie verstehen also auch die alten Programmtexte, die mit Zeilennummern geschrieben sind.

Wie man sich leicht denken kann, ist der Übergang innerhalb einer Leistungsstufe oder zu einer höheren Leistungsstufe aufgrund der Aufwärtskompatibilität immer der einfachere. Schwierig wird es, wenn man von einem höheren Dialekt auf eine niedrigere Stufe hinabsteigen will, weil dort einige der evtl. verwendeten Sprachelemente einfach nicht existieren. Im folgenden wollen wir uns konkret mit dem AmigaBASIC als Ausgangsdialekt beschäftigen, sozusagen mit dem schwierigsten Fall. Die anderen Fälle kann man sich aus den zugehörigen Erörterungen wohl leicht selbst erarbeiten. Eine letzte allgemeine Betrachtung vorweg muß den Grafikeigenschaften gelten. Auf PCs gibt es ab dem GWBASIC auch schon den kompletten Satz an Grafikbefehlen mit nur leicht abweichenden Bedeutungen, aber PCs haben nun mal von Haus aus ganz andere Grafikmodi als ein Amiga. Vor allem die Fenstertechnik wird im Gegensatz zum AmigaBASIC von diesen Dialekten überhaupt nicht unterstützt. Hier gibt es lediglich einen SCREEN-Befehl, der einen bestimmten Grafikmodus an-

wählt. Wenn vorher in einem anderen Modus (z. B. dem einfachen Textmodus) schon ein Bildschirminhalt bestand, ist nicht sicher, daß dieser nach Zurückschalten auf den Textmodus wieder sichtbar wird. Vielmehr werden die Videoinhalte schon leicht überschrieben. Zusätzlich sind bei den PC-Grafikmodi zahlreiche individuelle Eigenheiten zu beachten, so haben der COLOR- und der PALETTE-Befehl im CGA-Modus ganz andere Wirkungen als im EGA-Modus! Man sollte sich also merken, daß man aus SCREEN- und WINDOW-Befehlen des Amigas jeweils einen einzelnen SCREEN-Befehl auf dem PC machen muß. Will man darüber hinaus mehrere Fenster (Windows) auf einem Schirm darstellen, muß man sich die nötigen Routinen gefälligst zunächst selbst schreiben (es wäre in etwa die Layers-Library des Amigas nötig, was wohl kaum jemand aus dem Ärmel schütteln wird). Also, am besten Finger weg von solchen Fällen bzw. kräftige Umdisposition. Daß auch die Sound-Fähigkeiten des PCs nicht an die eines Amigas heranreichen, braucht wohl nur nebenbei erwähnt zu werden. Konkret wollen wir die Übertragung am Beispiel eines Programms von der ExtrasD-Diskette durchspielen, wir nehmen das Programm CLogo aus der Schublade BasicDemos. Dieses ist von noch überschaubarer Länge, zeigt aber andererseits gerade einige Differenzen der beteiligten Basic-Dialekte auf. Das Listing der Amiga-Version er-

sparen wir uns hier, jeder kann es sich direkt auf seinem Amiga ansehen. Um die Sache nicht zu sehr zu verkomplizieren, nehmen wir einfach einmal an, jeder hätte einen Amiga 2000 mit einer Brückenkarte (egal ob PC oder AT). Diese kann CGA-Grafik direkt emulieren, weshalb wir diesen Grafikmodus zunächst ansteuern. Um das Programm auf die MS-DOS-Seite hinüberzubringen, speichert man es zunächst als ASCII-Datei ab. Dazu ruft man es ganz einfach auf, z. B. per Maus und bricht es nach den ersten paar Durchläufen mit der q-Taste ab. Im Output-Fenster befiehlt man dann im Direktmodus:

SAVE "RAM:CLogo",A

Mit der A-Option sorgt man bekanntlich für die Abspeicherung in ASCII-Form. Als Ziellaufwerk ist hier die RAM-Disk gewählt, weil wir noch ein bißchen daran herumwerkeln und dabei das Original nicht zerstören wollen. Wir wollen hier gleich den schwierigen Fall besprechen, daß das Programm unter dem GW-BASIC-Interpreter zum Laufen gebracht werden soll. Was bei der QuickBasic-Version einfacher wäre, wird am Schluß verraten. GWBASIC benötigt vor allem Zeilennummern, die im Amiga-Programm überhaupt nicht vorkommen. Da hilft nichts, die fügt man am besten von Hand dazu. (Wenn man viele solcher Programme zu übertragen hat, lohnt sich ein kleines Programm nach Art von Listing 3, um die Zeilennummern automatisch zu erzeugen.)

Wir benutzen für die ersten Manipulationen den Editor ED auf Amiga-Seite und rufen ihn auf mit **ED RAM:CLogo**. Bevor wir allerdings die Zeilennummern dazufügen, löschen wir (per Ctrl-B) gleich ein paar typische Amiga-Befehle, mit denen ein PC sowieso nichts anfangen kann. Am Anfang ist das gleich die Zeile mit dem Befehl "WINDOW 2,...", dann ab zirka Zeile 18 die vier Zeilen "PALETTE" und den folgenden "COLOR"-Befehl auch gleich mit (letzteren gibt es wohl auch auf dem PC, aber bei CGA-Grafik in ganz anderer Bedeutung). Am Ende entfernen wir den Befehl "COLOR 1", die Zeile mit vielen "AREA-", "WINDOW CLOSE"- und "PALETTE"-Befehlen. Alle Leerzeilen eliminiert man bei der Gelegenheit auch gleich. Den Rest der Zeilen nummerieren wir irgendwie sinnvoll, aber unbedingt in **aufsteigender Reihenfolge in Zehnerschritten** durch. Was dabei herauskommt, ist fast egal, weil keine der Zeilennummern mit GOTO oder GOSUB angesprungen wird. Diese Datei speichern wir dann mit ESC X ab. Da wir auf einem Amiga 2000 arbeiten, wird jetzt einfach ein PC-Fenster geöffnet und die Datei herübergeholt:

AREAD RAM:CLogo CLOGO.BAS

Bei der Janus-2.0-Software werden dabei auch gleich automatisch die Zeichensätze konvertiert, was man sonst mit den paar Umlauten und zu Fuß im Editor nachholen

Listing 3: Automatisches Numerieren einer Programmdatei

```

10 REM "Programm für die Amiga- Seite
20 PRINT "Dieses Programm versieht eine vorhandene
30 PRINT "Basic-Programmdatei automatisch mit Zeilennummern.
40 PRINT PRINT "Das Programm muß schon in ASCII-Form vorliegen.
50 PRINT
60 INPUT "Name der bestehenden ASCII-Programmdatei = ";fe$
70 INPUT "Anfangs-Zeilenummer = ";na
80 INPUT "Zeilennummern-Schrittweite = ";nd
90 INPUT "Name der zu erzeugenden Programmdatei = ";fo$
100 OPEN fe$ FOR INPUT AS 1
110 OPEN fo$ FOR OUTPUT AS 2
120 WHILE NOT EOF(1)
130 LINE INPUT #1,a$
140 REM "Kein Leerzeichen am Zeilenanfang erlaubt!
150 PRINT #2,MID$(STR$(na),2);" ";a$
160 na=na+nd
170 WEND
180 CLOSE 2:CLOSE 1
190 END

```

Listing 3 erzeugt nach dem Start Zeilennummern für eine Programmdatei, die als ASCII abgespeichert wurde

muß. Wer keine Brückenkarte hat, muß an dieser Stelle die oben erwähnten Transferprogramme oder serielle Übertragungen verwenden. Wir haben die Programmdatei nun glücklich auf der PC-Seite, können GW-BASIC aufrufen und das Programm gleich laden. Dabei dürften keine Fehler auftreten, man sollte das Programm allerdings noch nicht starten! Spätestens beim ersten "AREA" (so noch nicht entfernt!) würde uns ein Syntax-Error gemeldet, sowas kennt man bei PCs leider nicht. Aber der Reihe nach.

Im Editor des GW-BASIC gehen wir nun also an die Feinarbeit. Sie können ab jetzt auch schon in das Listing 4 schielen, wo Sie das Endergebnis finden. (Bitte sich nicht zu sehr über die dortigen Zeilennummern wundern. So glatt wie hier beschrieben lief die Umsetzung natürlich nicht, es wurde schon ziemlich herumprobiert, wobei die Zeilennummern auch krumme Werte abkamen.) Zunächst müssen wir vor dem CLS eine Zeile ergänzen, die den SCREEN- und WINDOW-Befehlen beim Amiga entspricht. Da wir CGA-Grafik machen wollen, lautet es hier SCREEN 1,0,0. In derselben Zeile sollte man mit KEY OFF auch die unterste Zeile für die Grafik zugänglich machen.

Der erste zu ändernde Befehl ist der PAINT-Befehl, mit dem der Bildschirm gelöscht wird. Hier wird übrigens PAINT statt CLS genommen, weil der Bildschirm nicht auf Hintergrundfarbe, sondern auf Weiß gesetzt werden soll. Dies soll zukünftig nur noch einmal vor der WHILE-Schleife geschehen. Wir ersetzen mit diesem PAINT-Befehl also den alten CLS-Befehl am Anfang und müssen noch die Farbnummer in 3 ändern. Innerhalb der Schleife nehmen wir anstelle des PAINT-Befehls lieber einen LINE-Befehl mit BF (BlockFill)-Option, der ein gefülltes Rechteck zeichnet. Wir füllen damit nur den veränderlichen Teil des Schirms, also unterhalb der Titelzeile (s. u.) und etwas rechts vom äußersten linken Rand:

LINE (10,19)-(319,199),3,BF

Daran sieht man gleich noch einiges: Die Farbnummer für Weiß ist bei CGA die 3, und die maximalen Koordinaten entsprechen der Amiga-

NTSC-LowRes 320x200. Die nächsten drei Zeilen des Originalprogramms mit LOCATE und zweimal PRINT für die Titelzeile transferiert man am besten vor die WHILE-Schleife, den Text muß man wegen der halbierten Schirmbreite kürzen. Hinter LOCATE ergänzt man einen Befehl COLOR, 1. Damit wird bei CGA-Grafik diejenige Farbpalette ausgewählt, in der Blau und Rot vorkommen (genauer Kobalt und Lila), die ja im Commodore-Firmenzeichen benötigt werden. Das Ergebnis ist in Zeilen 50 bis 52 von Listing 4 zu bewundern.

In der folgenden Zeile des Originals sind die Maßstabsfaktoren UY und UX den CGA-Verhältnissen anzupassen, siehe Zeile 130 von Listing 4. Dem Kommentar zu dieser Zeile ist übrigens zu entnehmen, daß dieses Programm vor langer Zeit schon einmal eine Umwandlung mitgemacht hat. Es ist nämlich gar nicht ursprünglich auf dem Amiga entstanden, sondern noch früher auf einem CBM 8032 mit einer 512x512-Grafikkarte, damals auch noch mit Zeilennummern, die beim Amiga weggefallen waren.

In der Zeile, die mit "x2 = " beginnt, ist für die GW-BASIC-Version noch eine Zuweisung CC = 1 zu ergänzen, das ist die CGA-Farbnummer für Blau, genauer gesagt Himmelblau, etwas Besseres gibt es hier nicht, siehe Zeile 162 in Listing 4. Diese Farbnummer CC ist dann in den LINE-Anweisungen zu ergänzen und in den CIRCLE- und PAINT-Befehlen statt der 0 einzutragen, siehe Zeilen 166 bis 174 in Listing 4.

Statt der COLOR-Anweisung vor den AREA-Befehlen schreiben wir eine Zuweisung für die rote Farbnummer bei CGA: CU = 2, siehe Zeile 182 in Listing 4.

Mit den folgenden AREA-Befehlen wird es ernst: Hier muß man sich etwas Neues ausdenken, GW-BASIC kennt diese Befehle nicht, man kann lediglich mit PAINT eine umrahmte Fläche nachträglich ausfüllen. Zur Umrahmung der Fähnchen benutzen wir den LINE-Befehl. Und jetzt wird auch klar, warum die AREA-Befehle bisher nicht gelöscht wurden: Meistens braucht man nur das Wort AREA durch LINE zu überschreiben. Die beiden ersten AREA-Befehle muß man al-

Listing 4: CGA-Version des Programms CLogo

```
1  REM save"clogoc",a:rem PC-CGA-Adaptierung 29.9.89,
   25.10.89
10  ' Commodore-Logo C= 4.7.86 Ki, Aktualisierung 5.12.86
25  SCREEN 1,0:KEY OFF
30  PAINT (1,1),3 ' Schirm löschen
50  LOCATE 2,2:COLOR,1
52  PRINT " Größe zufällig, Abbruch mit q-Taste "
100 WHILE INKEY$ < > "q" ' Endlosschleife bis Abbruch mit q-Taste
110 F=.5+.5*RND(1) ' Größe zufällig zwischen 0.5 und 1
120 LINE (10,19)-(319,199),3,BF
130 UY=F*180/512:UX=UY*1.2 ' Umrechnung 8032-Highspeed in CGA-Koord
132 IR=107:AR=200:RR=61:FL=170:FK=97:FM=5 ' Logo-Abmessung en
134 XM=300/F:YM=256/F ' Mittelpunkt
136 R1=IR*IR:R2=AR*AR
160 X=XM+RR ' Das C zeichnen
162 X2=X-XM:X2=X2*X2:CC=1 ' CC ist C-Farbe blau
164 Y1=SQR(R1-X2):Y2=SQR(R2-X2)
166 LINE (UX*X,UY*(YM+Y1))-(UX*X,UY*(YM+Y2)),CC
168 LINE (UX*X,UY*(YM-Y1))-(UX*X,UY*(YM-Y2)),CC
170 PH=RR/IR:PH=3.14159/2-ATN(PH/SQR(1-PH*PH)):CIRCLE
   (UX*XM,UY*YM),UX*IR,CC,PH,2*3.14159-PH
172 PH=RR/AR:PH=3.14159/2-ATN(PH/SQR(1-PH*PH)):CIRCLE
   (UX*XM,UY*YM),UX*AR,CC,PH,2*3.14159-PH
174 PAINT (UX*(XM-IR)-3,UY*YM),CC ' Das C ausmalen
180 Y2=SQR(R1-RR*RR)-FM
182 CU=2:REM COLOR 3 ' Unteres Fähnchen rot
184 REM AREA (UX*(XM+RR),UY*(YM+FM))
186 LINE (UX*(XM+RR),UY*(YM+FM))-(UX*(XM+RR+FK),UY*(YM+FM)),CU
188 LINE -(UX*(XM+RR+FL),UY*(YM+Y2)),CU
190 LINE -(UX*(XM+RR),UY*(YM+Y2)),CU
192 LINE -(UX*(XM+RR),UY*(YM+FM)),CU
194 PAINT (UX*(XM+RR+FL*.5),UY*(YM+.5*(Y2+FM))),CU
204 LINE (UX*(XM+RR),UY*(YM-FM))-(UX*(XM+RR+FK),UY*(YM-FM)),CC
206 LINE -(UX*(XM+RR+FL),UY*(YM-Y2)),CC
208 LINE -(UX*(XM+RR),UY*(YM-Y2)),CC
210 LINE -(UX*(XM+RR),UY*(YM-FM)),CC
212 PAINT (UX*(XM+RR+FL*.5),UY*(YM-.5*(Y2+FM))),CC
220 T=VAL(RIGHT$(TIMES$,1))+4:IF T>9 THEN T=T-10
222 WHILE T<>VAL(RIGHT$(TIMES$,1)):WEND
240 WEND 265 SCREEN 2,0,0 270 SYSTEM
```

Listing 4: Die fertige PC-Version des Programms CLogo für die CGA-Karte

erdings zu einem einzigen LINE-Befehl mit Anfangs- und Endpunkt zusammenfassen, und in den anderen ehemaligen AREA-Befehlen muß vor die erste Klammer noch ein Minuszeichen gesetzt werden, damit die Linie vom letzten Endpunkt zum angegebenen neuen Endpunkt weitergezogen wird. Der AREAFILL-Befehl wird durch eine PAINT-Anweisung ersetzt, siehe Zeile 194 in Listing 4. Außerdem ist in allen LINE-Befehlen und auch im folgenden PAINT-Befehl die Farbnummer CU anzugeben, siehe Zeilen 186 bis 194 in Listing 4.

Die alte Anweisung COLOR 0 löschen wir einfach. Mit den folgenden AREA- und PAINT-Anweisungen machen wir das gleiche wie oben, nur daß hier immer die Farbnummer

CC für Blau eingesetzt wird, vergleiche Zeilen 204 bis 212 in Listing 4. Die PAINT-Anweisung ist bis auf ein Minuszeichen identisch mit der in Zeile 194, man kann sie mit dem Editor leicht aus dieser erzeugen.

Die Warteschleife mit der leeren 9000er-FOR-Schleife wollen wir als gewissenhafte Programmierer gleich in eine echte Zeitschleife umwandeln, sonst muß man den Schleifenendwert immer an die verschiedenen Prozessorgeschwindigkeiten anpassen. Zeilen 220 bis 222 in Listing 4 zeigen eine Lösung für eine Zeit von zirka 4 Sekunden.

Vor dem Programmende fügt man am besten noch einen Befehl ein, der die CGA-Darstellung noch auf 80-Zeichen-Darstellung umschaltet. Das ist dann zwar einfarbig, man

Listing 5: Unterschiede gegenüber Listing 4 bei EGA- Grafik

```

1  REM save"clogoe",a:rem PC-EGA-Adaptierung 29.9.89,
   25.10.89
25  SCREEN 9,0:KEY OFF
30  PAINT (1,1),15 ' Schirm löschen
50  LOCATE 2,6:COLOR 15,0
52  PRINT "Größe des Firmenzeichens ist zufallsgesteuert, "
   ;
54  PRINT "Abbruch mit q-Taste "
120 LINE (10,29)-(600,349),15,BF
130 UY=F*339/512:UX=UY*1.36...
134 XM=300/F:YM=256/F
162 .....:CC=9...
182 CU=12:...
265 SCREEN 0,0,0

```

Listing 5 zeigt die Unterschiede gegenüber Listing 4 bei einer PC- EGA-Grafikkarte

kann aber sehr viel besser im Programm edieren: SCREEN 2,0,0. Und statt END kann man das Programm auch mit SYSTEM beschließen, um dann gleich zur DOS-Ebene zurückzukehren.

Wenn Sie Ihr Programm so weit haben: erstmal abspeichern! Damit das etwas bequemer geht, kann man sich den SAVE-Befehl schon, wie in Zeile 1 gezeigt, in einer REM-Zeile bereitstellen. Man holt sich dann nur Zeile 1 auf den Schirm, geht mit dem Cursor an deren Anfang, überschreibt die 1 und das REM mit der Leertaste und schickt den SAVE-Befehl ab. Das ist umso bequemer, je öfter man zwischendurch zur Sicherheit abspeichert.

Ist Ihnen eigentlich schon aufgefallen, daß in GW-BASIC alle Variablenamen durchgängig in Großschrift erscheinen, selbst wenn sie per Datei in Kleinschrift hereingekommen sind? Auch so ein kleiner Punkt, wo das AmigaBASIC

das modernere Entstehungsdatum erkennen läßt.

Bevor es jetzt endgültig losgeht, muß man zumindest beim Amiga 2000 darauf achten, daß man auch das PC-Color-Fenster geöffnet hat. Wenn nicht: Auf MS-DOS-Ebene den Befehl MODE CO80 geben, das Mono-Fenster schließen und das Color-Fenster öffnen.

Wenn Sie das Programm dann starten, werden Sie in der Titelzeile in der Regel statt Umlauten und ß nur Chaos sehen. Eine CGA-Karte hat von Haus aus nur einen 7-Bit-Zeichensatz! Es gibt aber den MS-DOS-Befehl GRAFTABL, mit dem man den Rest des Zeichensatzes resident ins normale RAM lädt. Wenn man dann wieder GW-BASIC CLOGOC aufruft, müßte es eigentlich klappen.

So, das wäre geschafft, aber es waren ja auch noch Tips für eine EGA-Version versprochen. In Listing 5 finden Sie bei den entsprechenden Zei-

lennummern passend zu Listing 4 die wenigen Änderungen. In der Hauptsache geht es um größeren Koordinaten-Spielraum und andere Farbnummern. Die Zeile 130 mit den Maßstabsfaktoren hat es übrigens in sich: Setzt man hier falsche Werte ein, wird beim Füllen der C-Kreisbögen der ganze Schirm gefüllt. Das passiert, wenn die beiden senkrechten Abschlußstriche des Cs nicht auf der richtigen Höhe liegen. Und diese ist äußerst empfindlich vom Verhältnis UY/UX abhängig. Wenn nun Ihre EGA-Karte eine andere Auflösung als die des Autors hat, muß am Faktor für UX solange probiert werden, bis die senkrechten Striche korrekt sitzen. Eine Verkleinerung des Faktors für UX läßt die Striche relativ nach außen rutschen.

Und wenn wir gerade von Versionen sprechen: Für QuickBasic wären alle genannten Änderungen auch nötig gewesen, mit einer wichtigen Ausnahme, den Zeilennummern.

Im ganzen gesehen war an diesem Programm doch sehr viel zu ändern. Danach wurde es aber auch ausgesucht. Andere Programme, bei denen es mehr um Rechnen oder Datenorganisation geht, sind da wesentlich anspruchsloser. Eine spezielle AmigaBASIC-Eigenschaft kam in diesem Beispielprogramm allerdings nicht vor, die auch noch viel Arbeit verursachen kann: Die Anweisungsblöcke bei IF-THEN. QuickBasic kann diese direkt verkraften, sie brauchen nicht geändert zu werden. Bei GW-BASIC muß man jedoch einige Klimmzüge machen und GOTOs einbauen.

Hat man einen einfachen Block ohne ELSE-Block, muß man die logische Bedingung bei IF invertieren (z. B. auf Gleich statt Ungleich abfragen) und hinter dem THEN eine Zeilennummer angeben, die die Zeile hinter dem Ende des Anweisungsblocks anspringt. Mit ELSE wird es entsprechend noch komplizierter. Eine weitere Einschränkung des GW-BASIC ist, daß es die langen Integervariablen des AmigaBASICs (Anhängsel &) nicht kennt, QuickBasic erst seit der neuesten Version. Aber wem das alles zu undurchdringlich erscheint, dem sei noch einmal versichert, daß wir hier die Stolpersteine extra konzentriert haben, in der täglichen Praxis kommt man bequemer davon.

(Dr. Peter Kittel/jb)

AMIGA DOS - INFO

Der vorliegende Artikel richtet sich vornehmlich an diejenigen, die neben einem Amiga auch mit einem PC oder der zum Amiga 2000 erhältlichen XT- oder AT-Karte arbeiten und die auf PCs existierenden BASIC-Dialekte und -Sprachen benutzen. Aber auch Um- oder Aufsteiger von anderen Systemen können anhand des Textes und der Beispiel-Listings Informationen über Umsetzungsschwierigkeiten bei BASIC-Programmen bekommen. Schauen Sie ruhig mal in die Programmzeilen rein, Sie werden bemerken, daß sich das eine oder andere Spezielle in einem Programm meistens sehr gut umsetzen läßt.

Das ist Ihre Chance...

Schon eine Kleinanzeige bringt oftmals großen Erfolg und hilft, neue Kontakte zu knüpfen. Nutzen Sie unser Angebot und profitieren Sie von der Tatsache, daß unsere Zeitschrift

»AMIGA DOS«

jeden Monat von zigttausend Computer-Interessierten gelesen wird.

Möchten Sie etwas verkaufen, tauschen, oder suchen Sie das »Tüpfelchen auf dem i« Ihre Annonce erscheint dann in der nächsterreichbaren Ausgabe.

Wir möchten ausdrücklich darauf hinweisen, daß wir keine Anzeigen veröffentlichen, aus denen ersichtlich ist, daß es sich hierbei um Veräußerungen von Raubkopien handelt.

Des weiteren machen wir darauf aufmerksam, daß indizierte Computerspiele nicht in Form von Anzeigen beworben werden dürfen.

Natürlich besitzt eine Erstausgabe noch keine Leserbriefe – es sei denn, einige Leser wären mit einer prophetischen Gabe ausgestattet, oder eingetragene Mitglieder im Hellseherverein. Deshalb wollen wir Ihnen an dieser Stelle den Leserservice des AMIGA-DOS-Redaktionsteams vorstellen.



SIE FRAGEN WIR ANTWORTEN

Der AMIGA-DOS-Leserservice

Wer einen Computer kauft, weiß noch gar nicht, was später an Problemen auf ihn zukommt – dies gilt auch für den Amiga. Sind es nun nichtlaufende Programme, fehlende Druckertreiber, nichtverstandene CLI-Kommandos oder Hardware-Defekte, irgendwann hat man als Amiga-Besitzer die eine oder andere Frage. Wir von der AMIGA DOS wollen eine Art 'Rettungsanker' für Sie sein, denn nicht immer hat man die Antwort zur Frage schnell parat.

Fragen Sie ruhig, denn Fragen kostet höchstens eine Briefmarke oder einen Telefonanruf

Die Redaktion der AMIGA DOS steht Ihnen in zweierlei Hinsicht zur Verfügung:

1.) Sie schreiben uns einen ausführlichen **Brief mit beigefügtem Rückumschlag** über Ihr Problem, und wir versuchen Ihnen so schnell wie möglich zu helfen. So schnell wie möglich heißt, daß wir entweder selbst die Antwort kennen und Sie Ihnen umgehend zusenden, oder (vor allem, wenn es um Anfragen zu

in der AMIGA DOS erschienenen Programmen geht) wir leiten die Frage an den zuständigen freien Mitarbeiter weiter. Dies dauert in der Regel zwar etwas länger, dafür können Sie (ebenfalls in der Regel) mit einer kompetenten und ausführlichen Antwort rechnen.

Leserbriefe, die Probleme schildern, die auch weitere Amiga-Besitzer interessieren können, werden auf den Leserbriefseiten mit einer ausführlichen Antwort abgedruckt, ebenso Briefe, deren Frage bei uns nicht direkt zu klären ist. Und deshalb sind die Leserbriefseiten **Ihre** Seiten, denn Sie sollen ein Forum für alle AMIGA-DOS-Leser und die, die es werden wollen, sein, sozusagen ein Treffpunkt der Amiganer.

2.) Die zweite Möglichkeit, die Hilfe des Redaktionsteams in Anspruch zu nehmen, besteht dienstags zwischen 17 und 20 Uhr bei der **AMIGA DOS HOTLINE**. Hier stehen Ihnen die Redakteure Rede und Antwort – wenn es um Fragen zum Heft oder rund um den Amiga geht. Dazu gehören auch Probleme, die bei einer der Lieblingsbeschäftigungen mit dem Amiga entstehen können – den Spielen. Denn **nicht nur zu Soft- und Hardwareproble-**

men wollen wir helfen, auch Probleme beim einen oder anderen Spiel können hier behoben werden.

Noch ein paar Hinweise zu beiden Methoden:

Wenn Sie einen Brief schreiben, bitte denken Sie daran, daß Sie **Ihre vollständige Adresse noch einmal in den Briefkopf schreiben**, somit ist gewährleistet, daß wir die Antwort auch ohne Briefumschlag noch an Sie richten können.

Bei telefonischen Anfragen, die Artikel oder Programme des Heftes betreffen, ist es sinnvoll, das entsprechende Heft neben sich zu legen. Angaben wie "Sie hatten da mal ein Programm, das funktioniert nicht!" helfen weder dem Leser noch dem Redakteur, und der Frust wird wohl auf beiden Seiten voll zuschlagen.

"So, der Worte sind genug gewechselt, nun laßt uns endlich Fragen hören!", könnte man jetzt sagen. Wir sagen allerdings noch eines: Scheuen Sie sich nicht, Fragen zu stellen, auch Anfänger haben eine reelle Chance, ihr Problem mit der AMIGA DOS zur Lösung zu bringen.

Wir freuen uns auf Ihren Anruf.

Ihre AMIGA-DOS-Redaktion



Garry Glendown

Die Jagd geht weiter...

Waren bis vor einiger Zeit die Bootblock-Viren die 'Geißel der Amiga-Welt', so sind es nun die Link-Viren. Es handelt sich dabei um Viren, die sich nicht über den Bootblock verbreiten, sondern über Programme. Viele Virenchecker sind nicht in der Lage, diese zu erkennen.

Im Gegensatz zu den Bootblock-Viren sind die Link-Viren auch eine Gefahr für Harddisk-User, da diese Programme vor keiner Harddisk haltmachen. Dazu kommt noch, daß Harddisks in den wenigsten Fällen schreibgeschützt sind, so sind Viren Tür und Tor geöffnet.

Um dies zu verhindern, muß man ständig alle wichtigen, im Speicher befindlichen Werte im Auge behalten. Damit dies nicht zu stressig wird, haben wir für Sie das Programm 'VirTest' entwickelt.

Zunächst ist es für den normalen Systemablauf wichtig, daß das Programm sich "ordentlich" vom CLI löst. Dies wird durch die Verwendung der Aztec-'Detach'-Library erreicht. Zu beachten ist dabei lediglich, daß einige Variablen definiert werden müssen (im Block zwischen '#ifdef DETACH' und '#endif'); außerdem steht kein Output-Kanal zur Verfügung, so können Funktionen wie 'printf' oder 'puts' nicht aufgerufen werden.

Letzteres wird durch Verwendung der Routine 'PrintWarning()' umgangen. Diese gibt einen Text in einem eigenen Console-Window aus und holt eine Eingabe.

Das Hauptprogramm besteht nur aus einer Endlos-Schleife, in der abwechselnd die Routinen 'CheckVirus' und 'Delay' aufgerufen werden, letztere mit dem Parameter 500, der etwa zehn Sekunden Wartezeit entspricht. Delay wurde deshalb verwendet, weil er praktisch keine Rechenzeit kostet und unabhän-

gig vom verwendeten Prozessor gleich lange wartet.

Außerdem spart man sich die umständliche Verwendung des Timer-Device.

Die Funktion 'CheckVirus' kann mit drei verschiedenen Parametern aufgerufen werden: Bei '0' werden die Vektoren überprüft, und bei Veränderungen wird eine Warnung ausgegeben.

Bei '1' werden alle geänderten Vektoren wieder zurückgestellt, so daß Viren deaktiviert werden (zumindest jedoch deren Reset-Routinen).

Der Parameter '2' sorgt dafür, daß die neuen Werte in den Puffer übernommen werden, etwa wenn eine reset-feste RAM-Disk installiert wurde. Doch wie reagiert VirTest auf die Link-Viren? Zu diesem Zweck haben wir die zwei uns zur Verfügung stehenden getestet: Der IRQ-Virus wird durch Änderungen des Kick-Tag-Pointers sowie der Exec-Vektoren gemeldet, das zweite Exemplar meldet sich nur durch den Kick-Tag-Pointer.

Sollte VirTest während des Betriebs einen 'Eindringling' melden, und sollten Sie sich sicher sein, daß Sie kein Programm (wie etwa Gomp) gestartet haben, so dürfte sich ein Virus im System befinden. In dem Fall bestätigten Sie den Requester mit 'j' sowie Return. Schließen Sie alle Arbeiten ab, auch die RAM-Disks nicht vergessen, und starten Sie Ihren Rechner neu (am besten ausschalten). Danach fahren Sie den Amiga von Hand hoch und starten als erstes VirTest. Als Parameter geben Sie 'i' an, womit die Wartezeit im Programm auf eine Sekunde reduziert wird.

Nun können Sie nacheinander die virusverdächtigen Befehle ausprobieren. Da die Wartezeit von VirTest reduziert ist, kann man recht schnell nach dem Aufruf schon genau wissen, ob der Befehl virulent ist oder nicht. Als Parameter in der Startup-Sequence empfiehlt sich 'L'. Damit werden alle internen Buffer auf die aktuellen Werte vordefiniert, damit zum

Beispiel resette RAM-Disks etc. nicht gemeldet werden.

Um VirTest aus dem System zu entfernen, muß man sich eines Debuggers bemächtigen. Dort sucht man sich den Task mit dem Namen 'VirTest...' und der Priorität '10' heraus. Danach geht man das Programm so lange in Einzelschritten durch, bis man den Aufruf der Delay-Routine ge-

funden hat (zu erkennen an der etwas längeren Wartezeit sowie dem BRA.S als einer der nächsten Befehle). Nun setzt man den Rechner auf den Befehl hinter dem BRA.S und läßt das Programm weiterlaufen. Nun ist VirTest entfernt; der Speicher ist auch freigegeben. Im Aztec-DB würde diese Aktion folgendermaßen aussehen:

```
> as
< tasknummer von VirTest
  eingeben >
> t
> 60T (eventuell 2x)
> t bis JSR <irgendwo> =
  Delay(500)
> t => Ausgabe: "addq.w
  #4,a7"
> t => Ausgabe: "bra.s <ir-
  gendwo>"
> r pc=pc+2
> ar
```

Listings

```
*****
*                               *
*   AMIGA DOS INFO             *
*   Programmname : VIRTEST     *
*   Funktion    : Helfer bei Virenbefall *
*   Sprache     : AZTEC-C      *
*   Besonderheiten : ----- *
*****

1: /*****\
2: *      VirTest *
3: *****/
4: *      Written 1989 by G. Glendown *
5: \*****/
6:
7: ULONG found,new;
8: #define COLD (long)(1<<0)
9: #define COOL (long)(1<<1)
10: #define KTPT (long)(1<<2)
11: #define VBI (long)(1<<3)
12: #define DOIOW (long)(1<<4)
13: #define EXECV (long)(1<<5)
14:
15: long Cool=0,Cold=0,KtPtr,DoIOW;
16:
17: UBYTE *EXECBUF,*AllocMem(),*ebpnt;
18:
19: long ebsize;
20:
21: #ifdef DETACH
22: long _stack=10000;
23: long _priority=-10;
24: long _BackGroundIO=0;
25: char *_procname="VirTest V1.01";
26: #endif
27:
28: struct FileHandle *out,*Open();
29: long Read();
30:
31: extern struct ExecBase *SysBase;
32:
33: int cnt;
34:
35: char msg[512];
36:
37: main(argc,argv)
38: int argc;
39: char *argv[];
40: {
41: int t;
42: LONG delay;
43:
44: if ((argc==2)&&(toupper(argv[1][0])=='L'))
45: CheckVectors(2);
46: if ((argc==2)&&(toupper(argv[1][0])=='I'))
47: delay=50L;
48: else delay=500L;
49:
50: ebsize=(long)((SysBase->LibNode).lib_NegSize);
51: if (EXECBUF=AllocMem(ebsize,MEMF_PUBLIC|MEMF_CLEAR))
52: {
53: ebpnt=(long)(SysBase)-ebsize;
54: for (t=0;t<ebsize;t++) EXECBUF[t]=ebpnt[t];
55: }
56: for (;;) {
57: CheckVectors(0);
58: Delay(delay); /* warten */
59: }
60:
61: if (EXECBUF) FreeMem(EXECBUF,ebsize);
62: }
63:
64: print(c)
65: char *c;
66: {
67: Write(out,c,(long)strlen(c));
68: }
69:
70: PrintWarning(c)
71: char *c;
72: {
```

```
73: long t;
74: char buf;
75:
76: if (out=Open("con:0/0/640/150/VirTest Virus Message",
77: MODE_OLDFILE)) {
78: print("Achtung!\n\nIch habe folgendes gefunden:\n\n");
79: print(c);
80: print("\nEntfernen (J/N) : ");
81: t=0;
82: while (!t)
83: t=Read(out,&buf,1L);
84:
85: if (toupper(buf)=='J')
86: CheckVectors(1);
87: else
88: CheckVectors(2);
89: Close(out);
90: }
91: }
92:
93: CheckVectors(remove)
94: int remove;
95: {
96: struct IntVector *v;
97: int t;
98: APTR w;
99:
100: msg[0]='\0';
101:
102: switch (remove) {
103: case 0:
104: new=0L;
105:
106: if ((long)(SysBase->CoolCapture)!=Cool)
107: new!=COOL,
108: strcat(msg,"CoolCapture geändert\n");
109:
110: if ((long)(SysBase->ColdCapture)!=Cold)
111: new!=COLD,
112: strcat(msg,"ColdCapture geändert\n");
113:
114: v=&(SysBase->IntVects[5]);
115: if (v->iv_Code<0xfc0000)
116: if (!(found&VBI))
117: new!=VBI,
118: strcat(msg,"Vertical Blank (Entfernen=RESET!)\n");
119:
120: if ((long)SysBase->KickTagPtr!=KtPtr)
121: new!=KTPT,
122: strcat(msg,"KickTagPtr verändert\n");
123:
124: if (EXECBUF) {
125: cnt=0;
126: for (t=0;t<ebsize;t++)
127: if (EXECBUF[t]!=ebpnt[t]) cnt++;
128: if (cnt)
129: new!=EXECV,
130: strcat(msg,"Exec-Vektor(en)\n");
131: }
132:
133: if (new)
134: PrintWarning(msg),
135: found=new;
136: break;
137:
138: case 2:
139: Disable();
140: Cold=(long)(SysBase->ColdCapture);
141: Cool=(long)(SysBase->CoolCapture);
142: KtPtr=(long)(SysBase->KickTagPtr);
143: for (t=0;t<ebsize;t++) EXECBUF[t]=ebpnt[t];
144: Enable();
145: break;
146:
147: case 1:
148: Disable();
149: for (t=0;t<ebsize;t++) ebpnt[t]=EXECBUF[t];
150: SysBase->ColdCapture=Cold;
151: SysBase->CoolCapture=Cool;
152: SysBase->KickTagPtr=KtPtr;
153: Enable();
154: break;
155: }
156: }
```


Martin Woop

Bibliotheks- routinen im Überblick

Wer selbst schon einmal auf dem Amiga programmiert hat, der kennt die Situation: Statt selbst-geschriebene Routinen zu entwickeln, ist es oft vorteilhaft, auf die reichhaltigen Bibliotheksfunktionen zurückzugreifen. Um hier erfolgversprechend arbeiten zu können, ist eine genaue Kenntnis der Routinen unbedingt erforderlich. Grund genug für AMIGA DOS, diese Routinen einmal zusammenzufassen.



Routinen

clist.library

```
-030: InitCLPool(a0=clpool,d0=size)
-036: AllocCLList(a1=clpool)
-042: FreeCLList(a0=clist)
-048: FlushCLList(a0=clist)
-054: SizeCLList(a0=clist)
-060: PutCLChar(a0=clist,d0=byte)
-066: GetCLChar(a0=clist)
-072: UnGetCLChar(a0=clist,d0=byte)
-078: UnPutCLChar(a0=clist)
-084: PutCLWord(a0=clist,d0=word)
-090: GetCLWord(a0=clist)
-096: UnGetCLWord(a0=clist,d0=word)
-102: UnputCLWord(a0=clist)
-108: PutCLBuf(a0=clist,a1=buffer,d1=length)
-114: GetCLBuf(a0=clist,a1=buffer,d1=maxlength)
-120: MarkCLList(a0=clist,d0=offset)
-126: IncrCLMark(a0=clist)
-132: PeekCLMark(a0=clist)
-138: SplitCLList(a0=clist)
-144: CopyCLList(a0=clist)
-150: SubCLList(a0=clist,d0=index,d1=length)
-156: ConcatCLList(a0=sourceclist,a1=destclist)
```

console.library

```
-042: CDInputHandler(a0=events,a1=device)
-048: RawKeyConvert(a0=events,a1=buffer,
    d1=length,a2=keymap)
```

diskfont.library

```
-030: OpenDiskFont(a0=textattr)
-036: AvailFonts(a0=buffer,d0=bufbytes,d1=flags)
```

dos.library

```
-030: Open(d1=name,d2=accessmode)
-036: Close(d1=file)
-042: Read(d1=file,d2=buffer,d3=length)
```

```
-048: Write(d1=file,d2=buffer,d3=length)
-054: Input()
-060: Output()
-066: Seek(d1=file,d2=position,d3=offset)
-072: DeleteFile(d1=name)
-078: Rename(d1=oldname,d2=newname)
-084: Lock(d1=name,d2=type)
-090: UnLock(d1=lock)
-096: DupLock(d1=lock)
-101: Examine(d1=lock,d2=fileinfoblock)
-108: ExNext(d1=lock,d2=fileinfoblock)
-114: Info(d1=lock,d2=parameterblock)
-120: CreateDir(d1=name)
-126: CurrentDir(d1=lock)
-132: IoErr()
-138: CreateProc(d1=name,d2=pri,d3=seglist,d4=stacksize)
-144: Exit(d1=returncode)
-150: LoadSeg(d1=filename)
-156: UnLoadSeg(d1=segment)
-162: GetPacket(d1=wait)
-168: QueuePacket(d1=packet)
-174: DeviceProc(d1=name)
-180: SetComment(d1=name,d2=comment)
-186: SetProtection(d1=name,d2=mask)
-192: DateStamp(d1=date)
-198: Delay(d1=timeout)
-204: WaitForChar(d1=file,d2=timeout)
-210: ParentDir(d1=lock)
-216: IsInteractive(d1=file)
-222: Execute(d1=string,d2=file,d3=file)
```

exec.library

```
-030: Supervisor()
-036: ExitIntr()
-042: Schedule()
-048: Reschedule()
-054: Switch()
-060: Dispatch()
-066: Exception()
-072: InitCode(d0=startclass,d1=version)
-078: InitStruct(a1=inittable,a2=memory,d0=size)
-084: MakeLibrary(a0=funcinit,a1=structinit,a2=libinit,
    d0=datasize,d1=codesize)
```



```
-090: MakeFunctions(a0=target,a1=functionarray,a2=funcdispa
se)
-096: FindResident(a1=name)
-102: InitResident(a1=resident,d1=seglist)
-108: Alert(d7=alertnum,a5=parameters)
-114: Debug()
-120: Disable()
-126: Enable()
-132: Forbid()
-138: Permit()
-144: SetSR(d0=newsr,d1=mask)
-150: SuperState()
-156: UserState(d0=sysstack)
-162: SetIntVector(d0=intnumber,a1=interrupt)
-168: AddIntServer(d0=intnumber,a1=interrupt)
-174: RemIntServer(d0=intnumber,a1=interrupt)
-180: Cause(a1=interrupt)
-186: Allocate(a0=freelist,d0=bytesize)
-192: Deallocate(a0=freelist,a1=memoryblock,d0=bytesize)
-198: Allocmem(d0=bytesize,d1=requirements)
-204: AllocAbs(d0=bytesize,a1=location)
-210: FreeMem(a1=memoryblock,d0=bytesize)
-216: AvailMem(d1=requirements)
-222: AllocEntry(a0=entry)
-228: FreeEntry(a0=entry)
-234: Insert(a0=list,a1=node,a2=pred)
-240: AddHead(a0=list,a1=node)
-246: AddTail(a0=list,a1=node)
-252: Remove(a1=node)
-258: RemHead(a0=list)
-264: RemTail(a0=list)
-270: Enqueue(a0=list,a1=node)
-276: FindName(a0=list,a1=name)
-282: AddTask(a1=task,a2=initpc,a3=finalpc)
-288: RemTask(a1=task)
-294: FindTask(a1=name)
-300: SetTaskPri(a1=task,d0=priority)
-306: SetSignal(d0=newsignal,d1=signalset)
-312: SetExcept(d0=newsignal,d1=signalset)
-318: Wait(d0=signalset)
-324: Signal(a1=task,d0=signalset)
-330: AllocSignal(d0=signalnum)
-336: Freesignal(d0=signalnum)
-342: AllocTrap(d0=trapnum)
-348: FreeTrap(d0=trapnum)
-354: AddPort(a1=port)
-360: RemPort(a1=port)
-366: PutMsg(a0=port,a1=message)
-372: GetMsg(a0=port)
-378: ReplyMsg(a1=message)
-384: WaitPort(a0=port)
-390: FindPort(a1=name)
-396: AddLibrary(a1=library)
-402: RemLibrary(a1=library)
-408: OldOpenLibrary(a1=libname)
-414: CloseLibrary(a1=library)
-420: SetFunktion(a1=library,a0=funcoffset,d0=funcentry)
-426: SumLibrary(a1=library)
-432: AddDevice(a1=device)
-438: RemDevice(a1=device)
-444: OpenDevice(a0=devname,d0=unit,a1=iorequest,d1=flags)
-450: CloseDevice(a1=iorequest)
-456: DoIO(a1=iorequest)
-462: SendIO(a1=iorequest)
-468: CheckIO(a1=iorequest)
-474: WaitIO(a1=iorequest)
-480: AbortIO(a1=iorequest)
-486: AddResource(a1=resource)
-492: RemResource(a1=resource)
-498: OpenResource(a1=resname,d0=version)
-504: RawIOInit()
-510: RawMayGetChar()
-516: RawPutChar(d0=char)
-522: RawDoFmt(a0,a1,a2,a3)
-528: GetCC()
-534: TypeofMem(a1=address)
-540: Procedure(a0=semaphore,a1=bidmsg)
-546: Vacate(a0=semaphore)
-552: OpenLibrary(a1=libname,d0=version)
```

graphics.library

```
-030: BltBitmap(a0=srcbitmap,d0=srcx,d1=srcy,
a1=destbitmap,d2=destx,d3=desty,
a4=size,x,d5=size,y,d6=minterm,d7=mask,
a2=tempa)
-036: BitTemplate(a0=source,d0=srcx,d1=srcmod,
a1=destrastport,d2=destx,d3=desty,
d4=size,x,d5=size,y)
-042: ClearEOL(a1=rastport)
-048: ClearScreen(a1=rastport)
-054: TextLength(a1=rastport,a0=string,d0=count)
-060: Text(a1=rastport,a0=string,d0=count)
-066: SetFont(a1=rastportid,a0=textfont)
-072: OpenFont(a0=textattr)
-078: CloseFont(a1=textfont)
-084: AskSoftStyle(a1=rastport)
-090: SetSoftStyle(a1=rastport,d0=style,d1=enable)
-096: AddBob(a0=bob,a1=rastport)
-102: AddVSprite(a0=vsprite,a1=rastport)
-108: DoCollision(a1=rastport)
-114: DrawGList(a1=rastport,a0=viewports)
-120: InitGels(a0=dummyhead,a1=dummytail,a2=gelsinfo)
-126: InitMask(a0=vsprite)
-132: RemIBob(a0=bob,a1=rastport,a2=viewports)
-138: RemVSprite(a0=vsprite)
-144: SetCollision(d0=type,a0=routine,a1=gelsinfo)
-150: SortGList(a1=rastport)
-156: AddAnimObj(a0=obj,a1=animationkey,a2=rastport)
-162: Animate(a0=animationkey,a1=rastport)
-168: GetGBuffer(a0=animationobj,a1=rastport,
d0=doublebuffer)
-174: InitGMask(a0=animationobj)
-180: GelsFuncE()
-186: GelsFuncF()
-192: LoadRGB4(a0=viewports,a1=colors,d0=count)
-198: InitRastPort(a1=rastport)
-204: InitVPort(a0=viewports)
-210: MrgCop(a1=view)
-216: MakeVPort(a0=view,a1=viewports)
-222: LoadView(a1=view)
-228: WaitBlt()
-234: SetRast(a1=rastport,d0=color)
-240: Move(a1=rastport,d0=x,d1=y)
-246: Draw(a1=rastport,d0=x,d1=y)
-252: AreaMove(a1=rastport,d0=x,d1=y)
-258: AreaDraw(a1=rastport,d0=x,d1=y)
-264: AreaEnd(a1=rastport)
-270: WaitTOF()
-276: QBlt(a1=blit)
-282: InitArea(a0=areainfo,a1=vectortable,
d0=vectortablesizes)
-288: SetRGB4(a0=viewports,d0=index,d1=r,d2=g,d3=b)
-294: QBSBlt(a1=blit)
-300: BltClear(a1=memory,d0=size,d1=flags)
-306: RectFill(a1=rastport,d0=x1,d1=y1,d2=xu,d3=yu)
-312: BltPattern(a1=rastport,a0=ras,d0=x1,d1=y1,d2=maxx,
d3=maxy,d4=fillbytes)
-318: ReadPixel(a1=rastport,d0=x,d1=y)
-324: WritePixel(a1=rastport,d0=x,d1=y)
-330: Flood(a1=rastport,d2=mode,d0=x,d1=y)
-336: PolyDraw(a1=rastport,d0=count,a0=polytable)
-342: SetAPen(a1=rastport,d0=pen)
-348: SetBPen(a1=rastport,d0=pen)
-354: SetDrMd(a1=rastport,d0=drawmode)
-360: InitView(a1=view)
-366: CBump(a1=copperlist)
-372: CMove(a1=copperlist,d0=destination,d1=data)
-378: CWait(a1=copperlist,d0=x,d1=y)
-384: VBeamPos()
-390: InitBltMap(a0=bitmap,d0=depth,d1=width,d1=height)
-396: ScrollRaster(a1=rastport,d0=dx,d1=dy,d2=minx,d3=miny,
d4=maxx,d5=maxy)
-402: WaitBOVP(a1=viewports)
-408: GetSprite(a0=simplesprite,d0=num)
-414: FreeSprite(d0=num)
-420: ChangeSprite(a0=vp,a1=simplesprite,a1=data)
-426: MoveSprite(a0=viewports,a1=simplesprite,d0=x,d1=y)
-432: LockLayerRom(a5=layer)
```



```
-438: UnlockLayerRom(a5=layer)
-444: SynkSBitMap(a0=1)
-450: CopySBitMap(a0=11,a1=12)
-456: OwnBlitter()
-462: DisownBlitter()
-468: InitTmpRas(a0=tmpr,al=buff,d0=size)
-474: AskFont(al=rastport,a0=textattr)
-480: AddFont(al=textfont)
-486: RemFont(al=textfont)
-492: AllocRaster(d0=width,d1=height)
-498: PriRaster(a0=planepr,d0=width,d1=height)
-504: AndRectRegion(a0=rgn,a1=rect)
-510: OrRectRegion(a0=rgn,a1=rect)
-512: NewRegion()
-522: ** reserviert **
-528: ClearRegion(a0=rgn)
-534: DisposeRegion(a0=rgn)
-540: FreeVPortCopList(a0=viewport)
-546: FreeCopList(a0=coplist)
-552: ClipBlit(a0=srcrtp,d0=srcx,d1=srcy,al=destrp,
             d2=destx,d3=desty,d4=size,x,d5=size,y,
             d6=minterm)
-558: XorRectRegion(a0=rgn,a1=rect)
-564: FreeCprList(a0=cprlist)
-570: GetColorMap(d0=entries)
-576: FreeColorMap(a0=colormap)
-582: GetRGB4(a0=colormap,d0=entry)
-588: ScrollVPort(a0=vp)
-594: UCopperListInit(a0=copperlist,d0=num)
-600: FreeGBBuffer(a0=animationobj,al=rastport,
                 d0=doublebuffer)
-606: BltBitMapRastPort(a0=srcbm,d0=srcx,d1=srcy,al=destrp,
                       d2=destx,d3=desty,d4=size,x,
                       d5=size,y,d6=minter)
```

icon.library

```
-030: GetWBOObject(a0=name)
-036: PutWBOObject(a0=name,a1=object)
-042: GetIcon(a0=name,a1=icon,a2=freelist)
-048: PutIcon(a0=name,a1=icon)
-054: FreeFreeList(a0=freelist)
-060: FreeWBOObject(a0=wbobject)
-066: AllocWBOObject()
-072: AddFreeList(a0=freelist,a1=mem,a2=size)
-078: GetDiskObject(a0=name)
-084: PutDiskObject(a0=name,a1=diskobj)
-090: FreeDiskObj(a0=diskobj)
-096: FindToolType(a0=tooltypearray,a1=typename)
-102: MatchToolValue(a0=typename,a1=value)
-108: BumbRevision(a0=newname,a1=oldname)
```

intuition.library

```
-030: OpenIntuition()
-036: Intuition(a0=event)
-042: AddGadget(a0=addptr,a1=gadget,d0=position)
-048: ClearDMRequest(a0=window)
-054: ClearMenuStrip(a0=window)
-060: ClearPointer(a0=window)
-066: CloseScreen(a0=screen)
-072: CloseWindow(a0=window)
-078: CloseWorkBench()
-084: CurrentTime(a0=seconds,a1=micros)
-090: DisplayAlert(d0=alertnumber,a0=string,d1=height)
-096: DisplayBeep(a0=screen)
-102: DoubleClick(d0=sseconds,d1=smicros,d2=cseconds,
                 d3=cmicros)
-108: DrawBorder(a0=rport,a1=border,d0=leftoffset,
                 d1=topoffset)
-114: DrawImage(a0=rport,a1=image,d0=leftoffset,
                 d1=topoffset)
-120: EndRequest(a0=requester,a1=window)
-126: GetDefPrefs(a0=preferences,d0=size)
-138: InitRequest(a0=req)
-144: ItemAddress(a0=menustrip,d0=menunumber)
-150: ModifyIDCMP(a0=window,d0=flags)
-156: ModifyProp(a0=gadget,a1=ptr,a2=reg,d0=flags,
                 d1=hpos,d2=vpos,d3=hbody,d4=vbody)
```

```
-162: MoveScreen(a0=screen,d0=dx,d1=dy)
-168: MoveWindow(a0=window,d0=dx,d1=dy)
-174: OffGadget(a0=gadget,a1=ptr,a2=req)
-180: OffMenu(a0=window,d0=menunumber)
-186: OnGadget(a0=gadget,a1=ptr,a2=req)
-192: OnMenu(a0=window,d0=menunumber)
-198: OpenScreen(a0=osargs)
-204: OpenWindow(a0=owargs)
-210: OpenWorkBench()
-216: PrintIText(a0=rp,a1=itext,d0=left,d1=top)
-222: RefreshGadgets(a0=gadgets,a1=ptr,a2=req)
-228: RemoveGadgets(a0=rempr,a1=gadget)
-234: ReportMouse(a0=window,d0=boolean)
-240: Request(a0=requester,a1=window)
-246: ScreenToBack(a0=screen)
-252: ScreenToFront(a0=screen)
-258: SetDMRequest(a0=window,a1=req)
-264: SetMenuStrip(a0=window,a1=menu)
-270: SetPointer(a0=window,a1=pointer,d0=height,d1=width,
                 d2=xoffset,d3=yoffset)
-276: SetWindowTitles(a0=window,a1=windowtitle,
                       a2=screentitle)
-282: ShowTitle(a0=screen,d0=showit)
-288: SizeWindow(a0=window,d0=dx,d1=dy)
-294: ViewAddress()
-300: ViewPortAddress(a0=window)
-306: WindowToBack(a0=window)
-312: WindowToFront(a0=window)
-318: WindowLimits(a0=window,d0=minwidth,d1=minheight,
                  d2=maxwidth,d3=maxheight)
-324: SetPrefs(a0=preferences,d0=size,d1=flag)
-330: IntuiTextLength(a0=itext)
-336: WBenchToBack()
-342: WBenchToFront()
-348: AutoRequest(a0=window,a1=body,a2=ptext,a3=ntext,
                 d0=pflag,d1=nflag,d2=w,d3=h)
-354: BeginRefresh(a0=window)
-360: BuildSysRequest(a0=window,a1=body,a2=postext,
                     a3=negtext,d0=flags,d1=w,d2=h)
-366: EndRefresh(a0=window,d0=complete)
-372: FreeSysRequest(a0=window)
-378: MakeScreen(a0=screen)
-384: RemakeDisplay()
-390: RethinkDisplay()
-396: AllocRemember(a0=rememberkey,d0=size,d1=flags)
-402: AlohaWorkbench(a0=wbport)
-408: FreeRemember(a0=rememberkey,d0=reallyforget)
-414: LockIBase(d0=dontknow)
-420: UnlockIBase(a0=iblock)
```

layers.library

```
-030: InitLayers(a0=li)
-036: CreateUpfrontLayer(a0=li,a1=bm,d0=x0,d1=y0,d2=x1,
                        d3=y1,d4=flags,a2=bm2)
-042: CreateBehindLayer(a0=li,a1=bm,d0=x0,d1=y0,d2=x1,
                        d3=y1,d4=flags,a2=bm2)
-048: UpfrontLayer(a0=li,a1=layer)
-054: BehindLayer(a0=li,a1=layer)
-060: MoveLayer(a0=li,a1=layer,d0=dx,d1=dy)
-066: SizeLayer(a0=li,a1=layer,d0=dx,d1=dy)
-072: ScrollLayer(a0=li,a1=layer,d0=dx,d1=dy)
-078: BeginUpdate(a0=layer)
-084: EndUpdate(a0=layer)
-090: DeleteLayer(a0=li,a1=layer)
-096: LockLayer(a0=li,a1=layer)
-102: UnlockLayer(a0=li,a1=layer)
-108: LockLayers(a0=li)
-114: UnlockLayer(a0=li)
-120: LockLayerInfo(a0=li)
-126: SwapBitsRestPortClipRect(a0=rp,a1=cr)
-132: WhichLayer(a0=li,d0=x,d1=y)
-138: UnlockLayerInfo(a0=li)
-144: NewLayerInfo()
-150: DisposeLayerInfo(a0=li)
-156: FattenLayerInfo(a0=li)
-162: ThinLayerInfo(a0=li)
-168: MoveLayerInFrontOf(a0=layer_to_move,
                        a1=layer_to_be_in_front_of)
```


mathffp.library

```
-030: SPFix(d0=float)
-036: SPFlt(d0=integer)
-042: SPCmp(d1=leftfloat,d0=rightfloat)
-048: SPTst(d1=float)
-054: SPAbs(d0=float)
-060: SPNeg(d0=float)
-066: SPAdd(d1=leftfloat,d0=rightfloat)
-072: SPSub(d1=leftfloat,d0=rightfloat)
-078: SPMul(d1=leftfloat,d0=rightfloat)
-084: SPDIV(d1=leftfloat,d0=rightfloat)
```

mathieeedoubbas.library

```
-030: IEEEEDPFix(d0=integer,d1=integer)
-036: IEEEEDPFlt(d0=integer)
-042: IEEEEDPCmp(d0=integer,d1=integer,d2=integer
d3=integer)
-048: IEEEEDPTst(d0=integer,d1=integer)
-054: IEEEEDPAbs(d0=integer,d1=integer)
-060: IEEEEDPNeg(d0=integer,d1=integer)
-066: IEEEEDPAdd(d0=integer,d1=integer,d2=integer
d3=integer)
-072: IEEEEDPSub(d0=integer,d1=integer,d2=integer
d3=integer)
-078: IEEEEDPMul(d0=integer,d1=integer,d2=integer
d3=integer)
-084: IEEEEDPDiv(d0=integer,d1=integer,d2=integer
d3=integer)
```

mathtrans.library

```
-030: SPAtan(d0=float)
```

```
-036: SPSin(d0=float)
-042: SPCos(d0=float)
-048: SPTan(d0=float)
-054: SPSincos(d1=leftfloat,d0=rightfloat)
-060: SPSinh(d0=float)
-066: SPCosh(d0=float)
-072: SPTanh(d0=float)
-078: SPExp(d0=float)
-084: SPLoc(d0=float)
-090: SPPow(d1=leftfloat,d0=rightfloat)
-096: SPSqrt(d0=float)
-102: SPTieee(d0=float)
-108: SPFieee(d0=float)
-114: SPASin(d0=float)
-120: SPACos(d0=float)
-126: SPLog10(d0=float)
```

potgo.library

```
-006: AllocPotBits(d0=bits)
-012: FreePotBits(d0=bits)
-018: WritePotgo(d0=word,d1=mask)
```

timer.library

```
-042: AddTime(a0=dest,a1=src)
-048: SubTime(a0=dest,a1=src)
-054: CmpTime(a0=dest,a1=src)
```

translator.library

```
-030: Translat(a0=inputstring,d0=inputlength,
a1=outputbuffer,d1=bufferize)
```

Softwareautoren für die Amiga-Computer gesucht.

Haben Sie nicht auch schon einmal daran gedacht, ein gutes Programm, das Sie selbst geschrieben haben, zu veröffentlichen?

Warum sollten nicht auch andere Leser in den Genuß Ihrer Mini-Dateiverwaltung, Grafikerweiterung, Tips, Tricks, Tools, Utilities, Simulationen, Games usw. kommen?

Wirklich gute Software, die den Anforderungen unserer Leser genügt, wird von uns entsprechend honoriert.

Sie sollten jedoch bei der Einsendung Ihres Programms ein paar Punkte beachten.

Wenn Sie Nachstehendes befolgen, wird Ihre Post zügig und ohne große Rückfragen und Verzögerungen bearbeitet:

Senden Sie uns Ihr Programm mit

- (a) allen benötigten Files auf der mit dem Programmnamen bezeichneten Diskette,
- (b) den kompletten Ausdrucken/Listings aller Files der Diskette,
- (c) einer Beschreibung Ihres Programms und
- (d) einer genauen Bedienungsanleitung.

Die Bedienungsanleitung und die Beschreibung sollten als Textdatei mit auf der Programm-Diskette enthalten sein. Wichtig für uns zu wissen wäre noch, mit welcher Konfiguration Sie arbeiten, welchen Drucker Sie benutzen, ob Sie ein zweites Laufwerk angeschlossen haben usw.

Wenn Sie der Meinung sind, ein solches Programm geschrieben zu haben, dann nichts wie einschicken an den

DMV-Verlag · Red. AMIGA DOS
Postfach 250 · 3440 Eschwege

Gewußt wie!

Die Seite für Ratsuchende

Wer Computer hat, der hat auch Sorgen, könnte man in Abwandlung eines alten Sprichwortes sagen. Beim Amiga treten diese Sorgen zum Beispiel in Gestalt von Hardwareproblemen, Programmierschwierigkeiten und Fehlern beim Arbeiten im DOS auf, ohne daß hierfür sofort Gründe erkennbar sind. Meistens reicht nur ein kleiner Denkanstoß, und schon liegt die Lösung des Problems vor Ihren Augen. Zu diesem Zweck gibt es in der AMIGA DOS die "Gewußt wie!"-Seiten.

Hier werden wir Ihnen kurze Tips und Kniffe zeigen, die Ihnen beim einen oder anderen Problem weiterhelfen sollen. Eigentlich ist die "Gewußt wie!"-Seite Ihre Seite, denn hier sollen Hilfen und Tips von Lesern für Leser erscheinen. Sie alle sind daher aufgefordert, sich zu beteiligen. Als Anreiz haben wir uns ein kleines Tauschgeschäft namens **Tausche Tip gegen DATABOX** ausgedacht. Näheres dazu finden Sie am Ende dieser Seite in der Info-Box. Doch nun zu den Tips:

Dateinamen in Variablen

Will man in Basic vernünftig mit Dateien arbeiten, so ist es unerlässlich, ab und zu den Anwender nach Dateinamen zu fragen. Doch wie kann man mit den gängigen Befehlen Dateien ansprechen, deren Name in einer String-Variablen abgelegt ist? Ganz einfach! Man gibt nach dem entsprechenden Befehl den Bezeichner der String-Variablen in Klammern (..) an. Ein Beispiel zum Ändern des aktuellen Directories sieht dann so aus:

```
REM chdir.bas
INPUT "Directory: ",a$
CHDIR (a$)
```

Farbumrechnung

In ILBM-Grafikdateien werden die Farben in einem besonderen Format abgespeichert. So ist für jeden der drei Farbanteile ein Byte reserviert, welches aber noch auf einen Wert zwischen null und eins umgerechnet werden muß, damit es mit Hilfe des

PALETTE-Befehls in ein Farbregister geschrieben werden kann. Wenn man nun davon ausgeht, daß sich eine ILBM-Grafikdatei im Speicher befindet und in der Variablen 'cmap' die Adresse liegt, an der die Farbdaten beginnen, so kann man mit der folgenden Routine alle Farbregister setzen:

```
REM color.bas
FOR i=0 to 31
  r=PEEK(cmap+i*3)/(16*15)
  g=PEEK(cmap+i*3+1)/(16*15)
  b=PEEK(cmap+i*3+2)/(16*15)
  PALETTE i,r,g,b
NEXT i
```

Frequenzen errechnen

Der BASIC-Befehl "Sound" verlangt unter anderem als Parameter die Frequenz des gewünschten Tones in Hertz. Die im Handbuch angegebene Tabelle ist nicht nur unzureichend, sondern auch noch in ihren Bezeichnungen falsch. Ausgehend vom Kammerton "a" ist es jedoch ein Kinderspiel, sich die Frequenzen selbst zu berechnen. Der Kammerton a hat eine Frequenz von 440,0 Hertz. Das Verhältnis zweier Halbtöne zueinander beträgt $2^{1/2}$. Will ich nun die Frequenz des Tones "c" berechnen, so zähle ich als erstes die Halbtonschritte. Diese betragen hier drei. Dann bestimme ich die Frequenz mit folgender Rechnung:

```
440 Hz*(12/2)*3 =
440 Hz*4 1/2 =
523.25 Hz
```

Entsprechend läßt sich jeder Ton ausrechnen. Eine Oktave nach oben erreicht man, indem man die Frequenz mit

zwei multipliziert, nach unten, indem man sie durch zwei dividiert.

Window-Strukturen unter Intuition

Jeder BASIC-Programmierer, der schon einmal versucht hat, Routinen der INTUITION-Library aufzurufen, wird festgestellt haben, daß diese unzählige Parameter irgendwelcher Strukturen benötigen. Entweder kennt man diese Strukturen gar nicht, oder man weiß nicht, woher man ihre Adresse bekommen soll. Doch dem Frust kann diese Tabelle abhelfen:

```
Das Fenster:
w&=WINDOW(7)
y-Koordinate der Maus
relativ zum Fenster:
PEEKW(w&+12)
x-Koordinate:
PEEKW(w&+14)
IDCPM-Flags:
PEEKL(w&+24)
Screen-Zeiger:
PEEKL(w&+46)
Rastport-Zeiger:
PEEKL(w&+50)
Zeiger auf Sprite-Daten für
Pointer:
PEEKL(w&+74)
Der Screen:
s&=PEEKL(WINDOW(7)+46)
Viewport:
s&+44
Rastport:
s&+84
Bitmap:
s&+184
Der Viewport:
v&=PEEKL(WINDOW(7)+46)+44
Zeiger auf ColorMap:
PEEKL(v&+4)
Viewport-Modi:
PEEKW(v&+32)
Der Rastport:
r&=PEEKL(WINDOW(7)+46)+84
Zeiger auf Zeichengenerator:
PEEKL(r&+52)
Die Bitmap:
b&=PEEKL(WINDOW(7)+46)+184
Zeiger auf
1. Bitplane: PEEKL(b&+8)
2. Bitplane: PEEKL(b&+12)
3. Bitplane: PEEKL(b&+16)
4. Bitplane: PEEKL(b&+20)
5. Bitplane: PEEKL(b&+24)
6. Bitplane: PEEKL(b&+28)
```

(Bernd Rudolf/jb)

Dateien verschieben

Der RENAME-Befehl des CLI dient normalerweise zum Umbenennen von Dateien. Wußten Sie, daß man mit ihm aber auch Dateien innerhalb von Verzeichnissen verschieben

kann? COPY erzeugt ja bekanntlich ein direktes Abbild der Originaldatei, ohne diese zu löschen. Meistens hat dies seinen Sinn, jedoch möchte man ab und zu die eine oder andere 'Karteileiche' aus einem Verzeichnis entfernen, ohne sie (vorsorglich) zu löschen. Dazu kopiert man die Datei nicht, sondern verschiebt sie. Wie? Zum Beispiel so:

Angenommen, im Verzeichnis DATEIEN existiert eine ALT_DATEI, und wir sind uns nicht im klaren, ob wir sie jemals wieder benötigen. In solchen Fällen bietet sich TRASHCAN (Mülleimer) an, ein Unterverzeichnis, welches mit Vorliebe Datenmüll aufnimmt, ohne ihn komplett zu entfernen. Anstatt nun mit COPY die Datei ins neue Verzeichnis zu kopieren und danach mit DELETE aus dem alten Verzeichnis zu entfernen, geben wir im CLI folgendes ein:

```
1> RENAME DATEIEN/ALT_DATEI T
O SYS:TRASHCAN/ALT_DATEI
```

Wenn wir jetzt im Verzeichnis DATEIEN nachschauen, finden wir die Datei ALT_DATEI nicht mehr, dafür aber im 'Mülleimer'.

CLI-'Fensterln'

Nach dem Start der Workbench finden wir die Disk-Icons, deren Inhalt wir uns mit der Maus ansehen können, indem wir die Icons anklicken. Für Workbench-Fans ist dies ein Muß. Es soll aber auch Amiga-Besitzer geben, die gerne ab und zu einen Abstecher ins CLI machen (mit gutem Grund, wie Sie anhand unseres CLI-Kurses sehen können). Nun wäre dies kein Problem, wenn man sich nach dem Start der Workbench-Diskette nicht noch dauernd durch sämtliche Icons durchsuchen müßte, bis sich endlich ein CLI-ähnliches Viereck zeigt (bei der WB-Version 1.3 gibt's zwar schon nach einem Versuch ein 'Shell-Icon', doch auch hier heißt es durchforsten, durchforsten...).

Wesentlich einfacher wäre es, das CLI-Window sofort präsent zu haben. Aber wie?

Dies läßt sich durch eine kleine Änderung der STARTUP-SEQUENCE im S-Verzeichnis erreichen. Dazu sollten Sie diese Stapeldatei mit dem Texteditor ED (oder einem anderen Texteditor) laden:

SPIELEN SIE GERNE?

Lieben Sie es, in finsternen und unheimlichen Gewölben den letzten Geheimnissen auf den Grund zu gehen? Zeichnen Sie gerne Karten von Gewölben und Spiellandschaften?

Dann sind Sie der richtige Partner für uns.

Denn für unsere AMIGA-DOS-Spieletips suchen wir ständig Tips, Tricks, Karten, Cheatmodes und alles andere, was das Spielerherz höher schlagen läßt. Wenn Sie interessante Informationen, Karten oder Lösungsvorschläge für Computerspiele herausgefunden haben, sind Sie unser(e) Mann/Frau. Durch Ihre Information können Sie vielen Spielern weiterhelfen, die ohne Ihre Hilfe unter Umständen monatelang an der gleichen Problematik "festhängen". Oft ist es nur ein Wort oder ein kleiner Hinweis, der dem Spielgeschehen neues Leben einhaucht.

Aus diesen Gründen möchten wir Sie um Ihre Mitarbeit bitten. Wenn Sie also über Spieletips - gleich welcher Art - verfügen, würden wir uns freuen, von Ihnen Post zu bekommen.

Und hier nun die Adresse, an die Sie Ihre Spieletips schicken können:

**DMV Verlag
Redaktion AMIGA DOS
Stichwort: Spieletips
Postfach 250
3440 Eschwege**

PS: Unter allen Einsendungen verlosen wir jeden Monat fünf Softwaretitel für die besten Spieletips.

Welt- hunger. Ernte- dank.

Die Deutsche Welthungerhilfe unterstützt Selbsthilfe-Projekte von Bauern der Dritten Welt, damit für sie Ernährung aus eigener Kraft möglich wird. Und sie hilft den Bauern, Natur und Umwelt als Lebensgrundlage zu erhalten, damit Entwicklung auch Zukunft hat.



**DEUTSCHE
WELTHUNGERHILFE**
Spendenkonto Sparkasse Bonn: 111

Adenauerallee 134 · 5300 Bonn 1 · Tel.: 02 28/22 88 0

3-fach Kickstart

Umschaltplatine für das vorhandene ORIGINAL-ROM und Platz für zwei zusätzliche Betriebssysteme mit Umschalter. Bestückt mit einem Betriebssystem Ihrer Wahl auf 4 Eproms. (leichter Einbau) DM 149,00
Zusätzliches Betriebssystem auf 4 Eproms. (Programmierservice) DM 100,00

Amiga DOS 1.3

Erweiterungspaket mit Kickstart-, Workbench- und Extras-Diskette sowie Benutzerhandbuch. DM 79,00

Amiga-DOS 1.3 Kit mit Kickstart-Umschaltplatine DM 220,00

Farbband-Kassetten

Commodore MPS-1200 schwarz	DM 12,60
Commodore MPS-1200 farbig	DM 16,95
Commodore MPS-1224 farbig	DM 20,75
Commodore MPS-1500 schwarz	DM 27,60
Commodore MPS-1500 Color	DM 29,90
Commodore MPS-2000 schwarz	DM 12,70
Commodore MPS-2000 Color	DM 31,80
NEC P-2200 schwarz	DM 14,10
NEC P-2200 farbig	DM 16,50
NEC P6+ schwarz	DM 14,90
NEC P6+ Color	DM 36,80
Star LC-10 schwarz	DM 10,80
Star LC-10 farbig	DM 13,25
Star LC-10 Color	DM 18,45
Star NL-10 schwarz	DM 13,30
Star NL-10 farbig	DM 18,40



Appel & Grywatz OHG
Werwolf 54
5650 Solingen 1
Telefon : 0212 / 13084
Telefax : 0212 / 10647
BTX : 0212 / 13083

M2AMIGA MODULA-2

DAS MEISTVERKAUFTE, SCHNELLSTE, BEWÄHRTESTE UND KOMFORTABELSTE MODULA-2 SYSTEM AUF DEM AMIGA GIBT ES IN EINER NEUEN VERSION MIT VIELEN PRAKTISCHEN ERWEITERUNGEN.

COMPILER	SFr.	DM
M2AMIGA VERSION 3.3	270.00	342.00
DEMOSKETTE (FISH-DISK 113)	10.00	10.00
UPDATE-PREIS	10.00	10.00
(ZUM UPDATE ORIGINALSKETTE EINSCHICKEN)		

ZU M2AMIGA IST EINE LANGE LISTE VON WERKZEUGEN UND BIBLIOTHEKEN ERHÄLTICH:

WERKZEUGE	SFr.	DM
SOURCE-LEVEL-DEBUGGER	180.00	228.00
AUTOMATISCHES MAKE	80.00	108.30
M2APSE ENVIRONMENT	80.00	108.30
M2DECODER	80.00	108.30
OBJEKTCONVERTER	80.00	108.30
SOURCECODE + RTS	80.00	108.30
IFF-BIBLIOTHEK	80.00	108.30
SPEED-EDITOR	80.00	108.30

TREASURES-BIBLIOTHEKEN	SFr.	DM
AMIGA TREASURES	158.00	201.78
FILE TREASURES	158.00	201.78
MODULA TREASURES	78.00	102.60
MATH TREASURES	78.00	102.60
TREASURES-DEMOSKETTE	10.00	10.00

REPORT-BIBLIOTHEKEN	SFr.	DM
INTUITION REPORT	80.00	108.30
DEVICE REPORT	80.00	108.30
GRAPHIC REPORT	80.00	108.30
REPORT-DEMOSKETTE	10.00	10.00

PD-DISKETTEN	SFr.	DM
AMOK PD-DISK, JE	10.00	10.00
TREASURES-PD-DISKETTE, JE	10.00	10.00

DIE GENANNTEN PREISE SIND UNVERBINDLICH

DIE MODULA-2 LEUTE:

BUNDESREPUBLIK DEUTSCHLAND:	
- H.J. MIELE-DATENTECHNIK GmbH	02983/8307
- SOS SOFTWARE SERVICE GmbH	0821/571081
- SW-DATENTECHNIK GmbH	04106/3998
- WILKEN & SABELBERG	0531/42689
- INTERPLAN-MUHLERT GmbH	089/1234066
- ADVANCED APPLICATIONS GmbH	0721/700912
SCHWEIZ:	
- FREI-ELEKTRONIK	01/945 54 32
ÖSTERREICH:	
- ICA ELEKTRONISCHE GERÄTE GmbH	0222/4545010

BEZUG AUCH BEI IHREM NÄCHSTEN COMPUTER- ODER BUCHHÄNDLER

GENERALVERTRIEB FÜR EUROPA:



A+L AG
DÄDERIZ 61
CH-2540 GRENCHEN/SO
TEL. (0041/0)65/52 03 11

ED SYS:S/STARTUP-SEQUENCE

Nach dem 'LoadWb'-Befehl geben Sie folgende Zeile ein:
NEWCLI CON:0/200/640/56/CLI-WINDOW

Nach einem anschließenden Druck auf die ESC-Taste und der Eingabe von 'X' wird die veränderte Datei abgespeichert.

Nach erneutem Boot-Vorgang der WorkBench finden wir im unteren Bildbereich ein neues CLI-Window mit eben diesem Namen.

Voraussetzung allerdings ist, daß Sie in den Preferences den 80-Zeichen-Modus eingestellt haben.

Was macht die NEWCLI-Anweisung nun? NEWCLI CON: richtet auf dem aktuellen Bildschirm ein neues Fenster (Window) ein. Dazu benötigen wir noch ein paar andere Angaben, nämlich den Standort der linken oberen Ecke des Windows ($x=0/y=200$), die Breite des Windows ($B=640$) und die Größe ($G=56$). Bei einem Screen von 640×256 Pixeln haben wir also den Bereich von 640×56 Pixeln von Vertikal 200 an belegt. Als letzte Angabe muß noch der Name des neuen Windows eingegeben werden.

Alles klar? Dann experimentieren Sie ruhig einmal.

Noch ein Tip für CLI-Enthusiasten: Man kann die Startup-Sequence im CLI enden lassen und dazu auch den vollen Bildschirm nutzen, indem wir die Datei wieder mit ED laden und die volle Bildschirmgröße eingeben:

NEWCLI CON:0/0/640/256/CLI

Beim Ablauf der Startup-Sequence wird der CLI-Task 1 gestartet, danach wird durch die NEWCLI-Anweisung TASK 2 gesetzt, während TASK 1 durch ENDCLI beendet wird. Am Ende haben wir ein CLI-Window zur Verfügung, welches die volle Bildschirmbreite nutzt.

Keine Wartezeiten beim Drucken

Unter dem CLI ist es recht einfach, Dateien auf den Drucker umzuleiten. Mit TYPE lassen sich dabei ganz gute Ergebnisse erzielen:

TYPE DATEI TO PRT:

sendet die Datei DATEI zum Ausgabegerät PRT: - eben dem Printer. Leider ist unser guter alter Amiga damit ein Weilchen beschäftigt, jeden-

falls dann, wenn des Druckers Speicher nicht allzu groß ist. Wie gut, daß unser guter, neuer Amiga multitaskingfähig ist. Was dies bedeutet? Ganz einfach, lassen Sie die Druckerausgabe im Hintergrund ablaufen. Dazu nimmt man einfach den CLI-Befehl RUN. Er bewirkt, daß andere CLI-Befehle nicht im Direktmodus funktionieren, sondern in den Hintergrund gelegt werden:

RUN TYPE DATEI TO PRT:

gibt zwar immer noch die DATEI auf den Drucker aus, jedoch steht Ihnen der Amiga sofort wieder zur Verfügung. Mit RUN lassen sich auch mehrere Befehle des CLI hintereinander abrufen:

RUN LIST+
INFO+
DIR

funktioniert auch. Hier muß nach jedem durch RUN aufgerufenen Befehl ein Pluszeichen '+' folgen, danach wird die <-d9- Taste gedrückt. Der Amiga arbeitet den ersten Befehl noch nicht ab, erst nachdem nach einem CLI-Kommando kein '+' mehr gefunden wurde, werden alle 'verketteten' Befehle abgearbeitet.

Klein, kleiner...

Kennen Sie Fraktal-Generatoren? Das sind Programme, die mathematische Berechnungen in Bilder umsetzen, mit meist fantastischen Möglichkeiten. Ein extrem kurzes Beispiel hierfür ist folgendes Programm in AmigaBASIC:

```
Variable:
  nts=20:a=0:b=0:c=1:d=1:m=
  256:r=640
Start:
  SCREEN 2,640,256,4,2:WIND
  OW 2,"Fraktale",2
  CLS
  FOR p=1 TO m
    FOR q=1 TO m
      x=a+(c-a)*p/m:y=b+(
d-b)*q/m
      FOR n=1 TO nts
        IF y>.5 THEN x=2
*x:y=2*y-1
        IF x>.5 THEN x=2
*x-1:y=2*y
        x=2*x:y=2*y
        IF x*x+y*y>r THE
N PSET(p,q),n:n=nts
      NEXT n
    NEXT q
  NEXT p
```

Es dauert eine Weile, bis das Bild fertig ist, aber experimentieren Sie ruhig einmal mit dem Programm. Das Programm entstammt übrigens

dem Buch 'Fractals everywhere' von M. Barnsley, erschienen bei Academic press, Inc. (jb)

Drucker automatisch einbinden

Folgender Tip dürfte für die interessant sein, die in ihren Amiga eine XT- oder AT-Karte eingebaut haben:

Möchte man den Drucker, der am Amiga angeschlossen ist, auch auf der PC-Seite nutzen, so muß im PC-Fenster das Icon LPT1 angeklickt werden. Dieses ist aber auf die Dauer doch recht mühsam.

Den PC-Teil des Amiga kann man automatisch hochfahren, indem man in der Startup-Sequence der PC-Workbench anstatt ENDCLI > NIL: den Befehl PCWINDOW COLOR einsetzt und auf der PC-Seite in die AUTOEXEC.BAT direkt an den Anfang MODE CO80 schreibt.

Wenn man jetzt vor dem PCWINDOW COLOR den Befehl RUN LPT1 setzt, wird im Hintergrund der Befehl LPT1 gestartet. Im Hintergrund durch RUN deswegen, weil sonst der Amiga nicht mehr zurückfindet aus dieser Befehlssequenz.

Es erscheint ein Fenster mit der Meldung "Parallel Port allocated to Emulator LPT1 V1.3". Anklicken braucht man dieses Fenster nicht. Nun steht der Benutzung des Druckers von der PC-Seite nichts mehr entgegen.

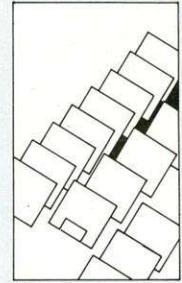
(Hermann-Josef Cuylen/jb)

Verstecktes Formatieren

Kennen Sie da auch? Man befindet sich im CLI, will schnell eine Startdiskette formatieren und benötigt die Kopie einer anderen Diskette. Nun ist das Format-Programm nicht gerade schnell, viel Zeit vergeht bis die Diskette fertig ist. Was tun? Man nimmt ganz einfach den RUN-Befehl zum formatieren und schreibt sich eine kleine Befehls-Datei, die folgende Befehle enthält:

```
NEWCLI CON:0/0/640/256/NEW-
FORMAT
RUN FORMAT DRIVE DF1: NAME DISK
NOICONS
INSTALL DF1:
ENDCLI
```

Die Diskette in DFA: wird nun formatiert, ohne den Benutzer zum Unterbrechen seiner Arbeit zu zwingen. (jb)



Tausche Tip gegen DATABOX

'Wir haben alle mal klein angefangen!' - Hinter dieser Überschrift verbirgt sich im Grunde nichts anderes, als der Appell, Verständnis für die aufzubringen, die ihren Amiga erst seit kurzer Zeit haben, und nun vor unüberwindlichen Problemen stehen, die so manchem Profi nur ein müdes Lächeln abringen.

Deshalb unser Aufruf an Fortgeschrittene: Wenn Sie einen guten Tip oder einen Griff in die Trickkiste wissen, lassen Sie von sich hören. Dabei ist es egal, ob es um BASIC, CLI, Assembler oder einfach nur Hinweise zum Arbeiten mit diesem Computer geht, nur eines ist Bedingung: Es muß interessant für Einsteiger sein, also nicht so: "Der COPY-Befehl wird zum Kopieren benutzt..."

Sie sollen sich aber nicht ohne Anreiz an die Arbeit machen:

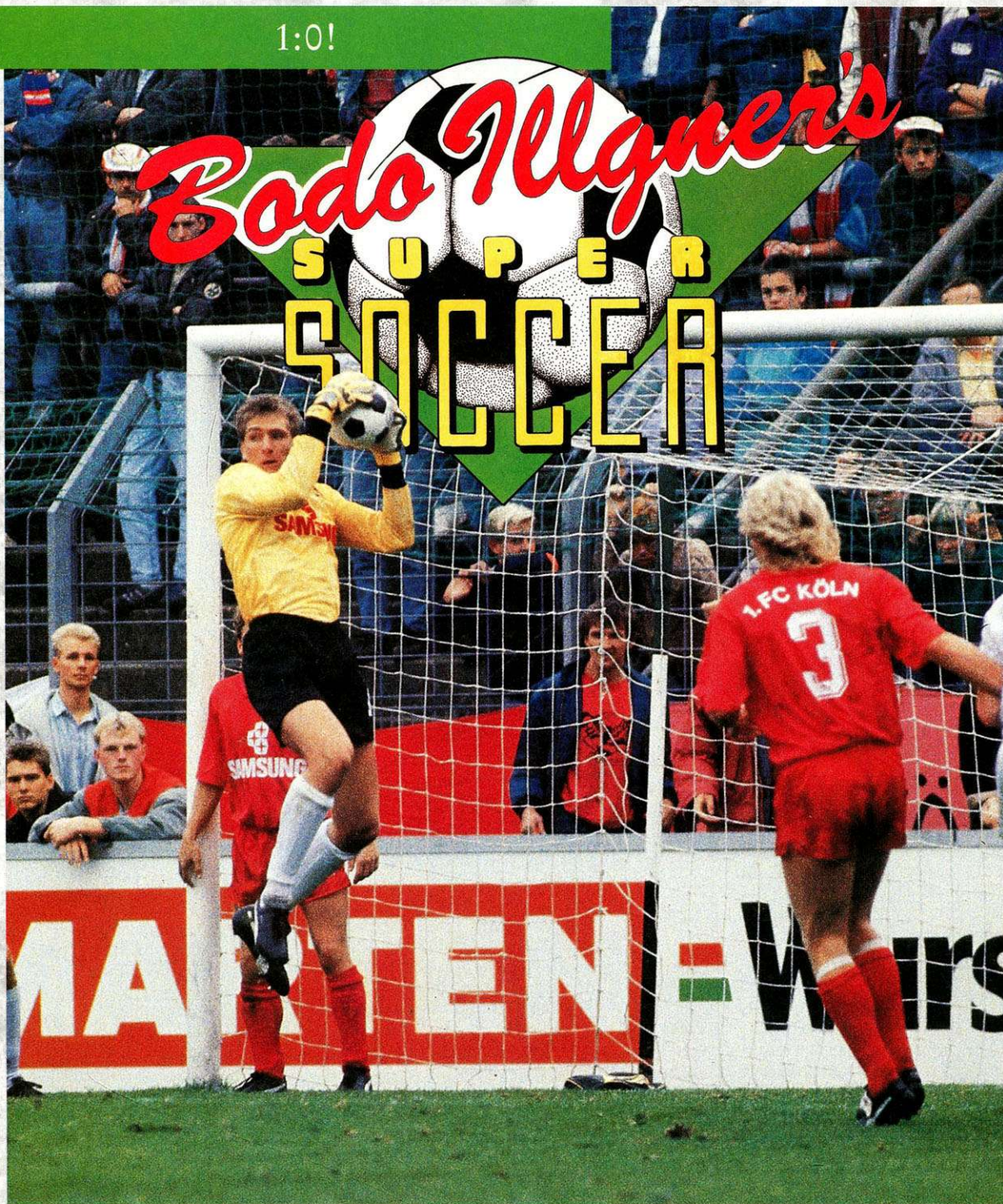
Wir bieten für jede Einsendung, die von uns für gut befunden und im Heft veröffentlicht wird, eine DATABOX der AMIGA DOS mit allen im Heft veröffentlichten Programmen.

Also ran an Tastatur und Maus und überlegt, was es zu verbessern oder zu erklären gibt. Oder Kurz-Tips schicken Sie bitte als Textfile auf Diskette an:

Redaktion AMIGA DOS
Kennwort: Tausche Tip
gegen DATABOX
DMV-Verlag
Postfach 250
3440 Eschwege

Ihre Diskette wird Ihnen nach Ansicht wieder zurückgeschickt, diejenigen, deren Tip angenommen wurde, finden gleichzeitig die neue DATABOX beiliegend. Viel Spaß beim Tipsuchen, und denken Sie dran: Neueinsteiger werden es Ihnen danken.

1:0!



Kennen Sie den Namen des Fußballspiels, bei dem man jeden Spieler voll kontrollieren kann — auch den Torwart? Bei dem man immer vollen Überblick über alle Spielzüge hat und keinen Ball ins Leere schießt? Bei dem man wirklich alles selbst bestimmen kann; die Länge einer Halbzeit, die Mannschaftszusammensetzung, alle Eigenschaften der Spieler bis hin zur Haarfarbe? Na?

Na klar! BODO ILLGNER's SUPER SOCCER!
ST, AMIGA, COMMODORE 64, AMSTRAD CPC



Ariola Soft
Das Programm

Auf der Diskette AOA von Digital Marketing befindet sich neben fünf Utilities, auf die wir später etwas näher eingehen wollen, eine Demoversion, die erwähnenswert ist. Der Name Stupid Demo läßt es wahrscheinlich schon vermuten, daß es sich hierbei um eine nicht ganz sinnvolle, aber wie gesagt, dennoch bemerkenswerte Sache handelt. Haben Sie schon einmal El Condor Pasa auf einem Diskettenlaufwerk gehört? Genau das macht nämlich dieses Programm. Nachdem man es gestartet hat, wird man aufgefordert, die Diskette herauszunehmen, dies ist eine Präventivmaßnahme, da nach Betätigen der Return-Taste das Laufwerk wirklich anfängt, diese bekannte Melodie zu spielen. Wie schädlich dies jedoch für das Laufwerk ist, kann man sich ja wohl denken. Glücklicherweise wird man aber vorher darauf hingewiesen, daß der Schrittmotor diese Spielerei nicht lange aushält...

Doch sehen wir uns lieber die nützlichen Programme auf dieser Diskette an, als da wären die Demoversion eines Texteditors, zwei Kopierprogramme namens A.H.A. Copy und AmigaCopy, einem Directory-Utility und einer Kombination aus Textverarbeitung und Datenbank. Aber immer schön der Reihe nach. Beginnen wir mit dem Texteditor TxEd.

Das hier vorliegende Programm ist eine Demoversion des Microsmiths Texteditor TxEd V1.1, von dem mittlerweile ein Update auf Version 1.3 zum Preis von 59.95 US\$ erhältlich ist. Dieses Update soll wesentlich schneller sein, als die ohnehin schon relativ schnelle Demoversion. Obwohl das Programm schon einige Jährchen auf dem Buckel hat (es stammt aus dem Jahre 1986) ist es doch recht leistungsfähig. Die Demoversion ermöglicht zudem nur die Erstellung von Textdateien, die weniger als 10 kByte lang sind. Ein weiterer Unterschied liegt darin, daß die Such- und Ersetzen-Funktion nicht in diesem Programm enthalten ist.

Was macht man eigentlich mit einem Texteditor, was man mit einer Textverarbeitung nicht kann? Diese Frage sollte wohl erst noch geklärt werden. Nun, ein Texteditor

Die Public-Domain-Werkzeugkiste

Möchten Sie für wenig Geld möglichst viel Software? Dann ist Public Domain das Schlagwort, das Sie sich merken sollten. Wir haben für Sie in der PD-Szene etwas herumgestöbert und einige Programme gefunden, deren Erwerb lohnenswert ist.

ist hauptsächlich dazu da, um Quelltexte für Compiler und Assembler möglichst schnell zu erstellen. Mit einem separaten Texteditor geht das meist schneller und komfortabler als mit den eingebauten, oft wenig benutzerfreundlichen Editoren der Compiler bzw. Assembler. Eine Textverarbeitung dagegen ist für einen ganz anderen Bereich zuständig und dementsprechend mit wesentlich mehr Funktionen ausgestattet.

Aber warum braucht man bei einem Editor auch Optionen, die beispielsweise Brief- oder Seriendrucke ermöglichen. Ein Editor ist eben nur eine Eingabehilfe, die das Erstellen von Quellprogrammen übernehmen soll. Klar, man könnte dies auch mit einer Textverarbeitung machen, aber wer will schon unnötig Geld ausgeben?

Die Menüleiste von TxEd ist mit den Einträgen Project, Edit, Cursor, Search und Random versehen. Im Menü Project kann man den Arbeitsspeicher leeren, Dateien laden oder speichern und das

Programm auch wieder verlassen. Das Menü Edit ermöglicht, Blockoperationen durchzuführen sowie Zeilen ganz oder teilweise zu löschen, aber auch versehentlich gelöschte Zeilen wiederherzustellen. Im Menü Cursor kann man beliebig im Text hin- und herspringen und das Fenster auf- und abscrollen lassen.

In der Option Random findet der Benutzer schließlich nützliche kleine Helferlein, wie z.B. das Anzeigen des Wertes des ASCII-Zeichens, auf dem sich der Cursor momentan befindet, einen direkten Aufruf eines neuen CLI-Fensters (hierbei muß jedoch erst der Befehl NewCLI zur Verfügung stehen). Weiterhin läßt sich hier auch die Darstellung des Bildschirmhintergrundes farblich umgestalten. Fast alle Funktionen sind auch über die Tastatur in Verbindung mit den Amigatasten aufrufbar. Alles in allem steht mit dieser Demoversion schon ein Werkzeug zur Verfügung, das den täglichen Umgang mit ASCII-Dateien zu einem Kinderspiel werden läßt.

Die nächsten beiden Programme sind Kopierprogramme, die es in sich haben. Bei A.H.A.Copy kann man sowohl für das Quell- als auch für das Ziellaufwerk die Gerätenummern df0: bis df3: festlegen und den Kopiermodus einstellen (Normal-, Nibble-, Parameter- oder IBM-Format-Copy). Des weiteren läßt sich die Start- bzw. Endtracknummer angeben sowie die Einstellung wählen, ob man ein- oder zweiseitig kopieren möchte.

Das zweite Kopierprogramm, AmigaCopy, besitzt ähnliche Funktionen, nur die Art des Kopiervorgangs läßt sich anders einstellen.

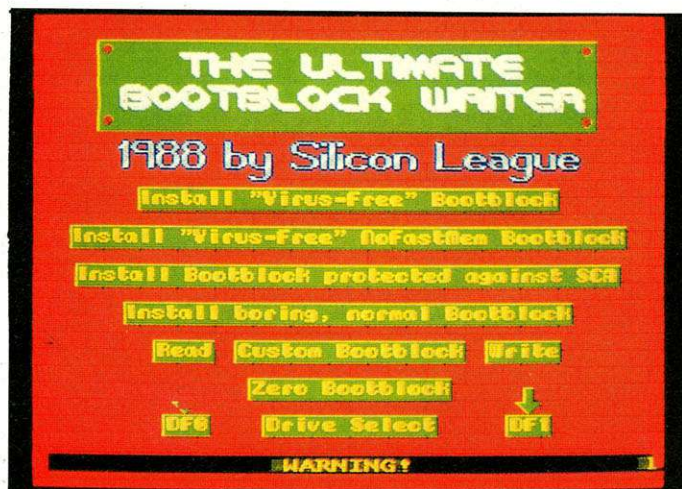
Die Optionen für die Kopiermodi heißen hier Fastcopy, Deepcopy1 bzw. Deepcopy2, wobei bei den Deepcopies das Diskettenformat keine Rolle spielt (es lassen sich also auch Fremdformate kopieren). Des weiteren kann man die Anzahl der Lese- und Schreibversuche einstellen, Verify ein- und ausschalten und überprüfen, ob sich das Duplikat in irgendeiner Weise von dem Original unterscheidet.

Das Directory-Utility ist ein sehr nützliches Heinzelmännchen, das das Bearbeiten von Disketten zum Kinderspiel macht.

Das Programmfenster ist in zwei Teile aufgeteilt worden, einem Unterfenster zum Auflisten der Filenamen und deren dazugehörigen Längen und einem Kommandofenster, das hauptsächlich aus anklickbaren Befehlen besteht. Aus Platzgründen können wir hier aber nur die wichtigsten Befehle aufzählen. Hier gibt es Befehle zum Einstellen des aktuellen Laufwerks sowie der RAM-Disk bzw. der Festplatte. Es gibt Funktionen zum Einstellen, Anzeigen, Herstellen und Löschen von Directories, weiterhin zum Auswählen, Anzeigen, Kopieren, Löschen, Editieren, Bewegen, ARCen und DE-ARCen, Ausdrucken und Ausführen von Files, zum Öffnen von CLI-Fenstern und vieles mehr.

Das vorletzte Programmfile auf dieser Diskette nennt sich TextKid II und ist eine kombinierte Text- und Adreßverarbeitung.

Für 10,- DM kann man vom Autor das neueste Update und eine Anleitung erhalten, mit der das Programm we-



Viel Programm für wenig Geld ist auf den Disketten von Digital Marketing der Fall: Nützliche Utilities wie der Bootblockwriter und diverse Virenjäger sind auf der DM A09 vorhanden

sentlich leichter zu handhaben ist.

Nach dem Laden meldet sich TextKid II mit einigen einleitenden Worten, in denen sich der Autor Michael Winterberg verewigt hat. Als nächstes erscheint das Hauptmenü, das sich in elf Unterbereiche gliedert. Hier können die einzelnen Programmkomponenten direkt angesprochen oder aber das Programm den eigenen Wünschen gemäß angepaßt werden.

Der erste Punkt, die Adreßverwaltung, kann nur angesprochen werden, wenn vorher schon eine Adreßmaske erstellt worden ist, beziehungsweise sich eine solche zusammen mit abgespeicherten Datensätzen auf der Datendiskette befindet.

In der Adreßverwaltung kann man ganz nach Belieben in erstellten Datenblättern wählen, neue Blätter erstellen, alte löschen, oder bereits bestehende auf dem Drucker ausgeben. Natürlich gibt es auch eine Funktion zum Speichern und Laden von Blättern, dazu muß man vorher allerdings eine Datei auf der Datendiskette angelegt haben. Mehr dazu jedoch später. Will man wissen, wieviel Datensätze verfügbar sind, braucht man nur den Menüpunkt "Info" anzuwählen, schon herrscht in dieser Beziehung keine Unklarheit mehr.

Funktionen zum Suchen nach bestimmten Kriterien und Sortieren fehlen leider, aber bei kleineren Datenmengen sind diese auch nicht unbedingt erforderlich.

Von der Adreßverwaltung kann man über das Menü Briefeditor in den Texteditor gelangen, um einen Brief zu schreiben. Vor dem Umschalten wird man gefragt, ob der Computer eine Briefmaske laden soll. Bejaht man diese Frage, so kann man einen schon geschriebenen Brief editieren. Der Briefeditor selbst entspricht in Aufbau und den Funktionen dem Texteditor. Die Menüleiste beinhaltet die Einträge Projekt, Editierung und Schriftart. Im Menü Projekt kann man den Text laden, speichern, drucken und die Anzahl der maximal verfügbaren Textseiten feststellen.

Mit Editierung kann man Blöcke markieren, löschen kopieren, zum Blockanfang bzw. -ende springen, und die Textseite direkt anwählen, die man editieren möchte.

TextKid II stellt verschiedene Schriftarten wie Normal, Italic, Unterstrichen und Fett zur Verfügung. Die Texte werden auf dem Bildschirm aber nicht nach dem WYSIWYG-Prinzip (What You See Is What You Get) dargestellt, sondern die Attribute werden durch Sonderzeichen angezeigt und erscheinen erst auf dem Drucker in der gewünschten Form. Das Menü Schriftart enthält außerdem noch den Punkt Format, mit dem man den Text blockweise oder gesamt links- oder rechtsbündig formatieren kann.

In die Briefe kann man mit verschiedenen Doppelkreuzanweisungen (#) Daten aus der Adreßverwaltung oder mit der Anweisung: "#Datum" das aktuelle Datum beim Ausdrucken einfügen. Auf diese Weise ist Mailmerging, also das Drucken von Serienbriefen, möglich.

Der nächste Punkt im Hauptmenü nennt sich Textaufbereitung. Hier können die Anzahl der Zeilen pro Seite sowie der linke und der rechte Rand festgelegt werden.

Man kann den Texteditor auch direkt vom Hauptmenü anspringen, hier entfällt dann auch die Frage nach der Briefmaske.

Der Hauptmenüpunkt Diskformat formatiert nicht, wie man vermuten könnte, eine neue Diskette, sondern legt auf einer leeren Diskette eine Datensatzdatei an, in der man seine mit der Adreßverwaltung erstellten Blätter speichern kann.

Im Menü Laufwerkseinstellung kann man festlegen, wo der Computer Daten speichern soll. Als Auswahlmöglichkeiten hat man hier nur die beiden Floppybezeichner DF0: und DF1:.

Ein weiterer Punkt ermöglicht die Übernahme der Systemkonfiguration seiner eigenen Workbench.

Schließlich verfügt TextKid II noch über ein Logbuch, in dem es, falls es aktiviert ist, alle Aktionen und Fehlermeldungen mit Zeit und Datum ablegt. So kann man feststellen, wann man welche Dateien bereits ausgedruckt hat oder welcher Ladeversuch nicht geklappt hat. Außerdem kann man eigene Meldungen in das Buch eintragen, der den Benutzer an bestimmte Termine erinnern soll.

Bleibt noch zu erwähnen, daß man fast alle Funktionen auch

über die Tastatur in Verbindung mit den Amigatasten erreichen kann.

Abschließend ist zu sagen, daß sich auf dieser Diskette noch ein Virus-Checker namens Virustest befindet, der beim Booten geladen wird und den Speicher nach diversen Viren (SCA, Byte Warrior, Byte Bandit etc.) überprüft. Alles in allem eine Diskette, deren Anschaffung sich wirklich lohnt!

Name: DM AOA
Vertrieb: Digital Marketing
Preis: 7,- DM

(Robert Marz/Jürgen Seibel
/br)

Anwendungsprogramme massiv

Bei dieser Public-Domain-Diskette spielt besonders die Zahl 18 eine große Rolle, denn es befinden sich sage und schreibe 18 Anwenderprogramme auf der Diskette.

Die Diskette ist selbstbootend, und nach einem kurzen Intro erscheint eine Menüleiste, aus der alle Programme mit der Maus angewählt werden können.

Vom ersten Programm namens TMN-FDD-System behaupten die Programmierer, da es eine ätzendere Wirkung als ein Desinfektionsmittel auf die Disketten ausübt. Na ja, im gewissen Sinne haben sie ja gar nicht mal so Unrecht, denn es handelt sich hier um ein Fast-Formatierungsprogramm, das die Disketten innerhalb weniger Sekunden formatiert. Der Vorteil dieses Programms liegt darin, daß es alle angeschlossenen Laufwerke anspricht, so daß ein gleichzeitiges Formatieren mehrerer Disketten möglich ist.

Das zweite Programm namens INM-Coder V1.0 ermöglicht das Codieren von Disketten. Dabei wird eine DOS-Diskette in einem speziellen Format codiert. Um eine codierte Diskette zu booten, benötigt man eine weitere Diskette, die die Code-Diskette entschlüsselt. So können ihre Daten vor den Klauen Dritter geschützt werden, vorausgesetzt daß jene Übeltäter nicht über beide Disketten verfügen. Leider ist hier die Voraussetzung, daß der Besitzer mindestens zwei

Diskettenlaufwerke besitzen muß.

Diejenigen, die schon lang nach Boot-Programmen gesucht haben, um ihr selbst erstelltes Programm in einem schönen Kleid zu präsentieren, sind mit dem Ultima Bootblockwriter gut bedient. Wie das Wort Ultima schon verrät, steht uns eine Vielzahl von "Bootern" zur Verfügung. Da wäre zuerst der "Virus-Free"-Booter, der lediglich seine Message auf den Bildschirm "zaubert" und den normalen Bootvorgang fortführt. Der "Virus-Free-NoFastMem" ist da schon ein wenig komfortabler. Denn das "NoFastMem" steht für NO Fast Memory. Das bedeutet, daß die möglichen Speichererweiterungen nicht vom Rechner erkannt werden. Dies ist besonders bei manchen Spieleprogrammen von Bedeutung. Ein weiterer Booter schützt gegen den allzu bekannten SCA-Virus, der schon so manchem Amigabesitzer Angst und Schrecken einjagte.

In den zumeist vom Chaos geprägten Räumen der Computerfreaks konnte man schon oft einen verzweiferten Hilferuf wahrnehmen. Die Ursache birgt meist zwei Möglichkeiten: Entweder ist der Computer nach der Eingabe von ellenlangen Listings abgestürzt, oder es haben sich unbarmherzige Viren auf den Disketten und dem Computer eingefunden. Spätestens jetzt ist es notwendig, seine gesamten Disketten zu "desinfizieren". Solche lästigen und vor allem zeitaufwendigen Arbeiten übernimmt der Virus-Killer Blizard-Protector 1.0. Dabei durchforstet er sowohl den Speicher als auch die Disketten nach dem bekannten SCA- und dem Byte-Bandit-Virus. Zusätzlich kann ein neuer Bootblock auf der Diskette installiert werden, der auch über den Viruskiller verfügt. So mancher Computerbesitzer, der sich in seiner Freizeit beim Spiel entspannen wollte, mußte erschreckend feststellen, daß das soeben erstandene Computerspiel sich beim Einladen weigert, dem Computer sinnvolle Daten weiterzugeben. In der Computersprache bezeichnet man solche Erscheinungen als ERROR. Um sich nun von vornherein abzusichern, ist diese Ansammlung von Kopierprogrammen sehr sinnvoll, da sie so manchen

Kopierschutz umgeht und so gleich eine Sicherheitskopie der Diskette erstellt. Dieses "Kopierpaket" umfaßt eine Ansammlung von drei Kopierprogrammen. Ein großes Manko ist jedoch, daß für alle drei Kopierprogramme ein zweites Laufwerk benötigt wird. Zudem kann auch die RAM-Disk nicht genutzt werden, so daß das Kopieren für Amigabenutzer, die nicht über ein zweites Laufwerk verfügen, mit diesen Kopierprogrammen ins Wasser fällt. Das Programm HP-10C ist eine Simulation eines Taschenrechners. Es verfügt über die wichtigsten mathematischen Funktionen. Hier werden zuerst die Grundrechenarten Addition, Multiplikation, Division und Subtraktion berücksichtigt. Zudem sind unter anderem trigonometrische und Logarithmus-Funktionen ermöglicht. Ein Zwischenspeichern von bestimmten Werten kann außerdem vorgenommen werden. Falls Sie sich einmal an den von Commodore mitgelieferten Zeichensätzen sattgesehen haben sollten, steht einer Umgestaltung des Fonts nichts im Wege. Mit dem Programm Font-Editor können Sie nun eigene Zeichensätze erstellen oder bereits vorhandene modifizieren. Zusätzlich ist es möglich, die erstellten Zeichen als ASCII-Codes abzulegen, so daß die Tastaturbelegung geändert werden kann.

AMCAT bringt Ordnung in Ihre Programmsammlung. Denn wem geht es nicht so, daß er ellenlange Directories auf seinen Disketten sieht, aber in der Vielzahl der Programme das gesuchte nicht ausfindig machen kann? AMCAT ermöglicht Ihnen, Ihre Computerprogramme zu erfassen und zu katalogisieren. Nach dem Einlegen der Diskette liest der Computer das Directory ein, worauf es als Programmliste sortiert und auf einer separaten Diskette abgespeichert werden kann. Die Besitzer eines Druckers können sich bei diesem Programm besonders glücklich schätzen, da ein Ausdrucken der sortierten Daten problemlos abläuft und man somit immer einen aktuellen Ausdruck der auf den Disketten befindlichen Programme hat. Um direkt zu sehen, was auf Disketten so vor sich geht, benötigt man einen sogenannten Monitor. Hierbei handelt



Des Amiganers Leid ist immer wieder die Unübersichtlichkeit der einzelnen Directories. Hier hilft DirUtil III Plus von der DM A04

es sich jedoch nicht um einen eigens für den Computer konstruierten Bildschirm, der eine höhere Frequenz als ein normaler Fernseher besitzt, sondern um ein Programm, das einen direkten Einblick in den momentanen Speicher des Computers ermöglicht. Auch solch ein Werkzeug, das mehr für den fortgeschrittenen Programmierer gedacht ist, befindet sich auf dieser Diskette. Nachteil der ganzen Sache ist jedoch, daß keine Anleitung, die die Befehlssätze erklärt, zu diesem Utility vorhanden ist. Man muß sich also direkt an den Programmierer wenden.

Das folgende Programm bringt ein bißchen Schwung in die anwendergeprägte Atmosphäre. Eine dynamische Melodie ist der Ausgangspunkt in das Programm HQC-Intro-Remover. Dabei speichert es ein Intro auf Diskette, das ein bestimmtes Programm dann automatisch bootet.

Da sitzt man nun entnervt vor seinem Amiga und schaut fassungslos auf den Bildschirm, in der Hoffnung, daß der Bootvorgang alsbald beendet ist. Aber das verzerrte Gesicht kann sich durch einen Bootblock in ein glückliches Lächeln wandeln lassen, vor allem, wenn der Bootblock von eigener Hand erstellt wurde. Dies ermöglicht der Jumping Bootwriter. Der Vorteil gegenüber einem einfachen Bootblock-Writer besteht darin, daß man bei dem Bootwriter einen eigenen Text (Message) eingeben kann. Der Jumping Bootwriter ist dabei sehr bedienungsfreundlich, da er über nur wenige Optionen verfügt. Durch diese Op-

tionen kann eine Laufschrift eingegeben, und/oder die Geschwindigkeit und Richtung des Scrollings variiert werden.

Jede Diskette ist in einzelne Bereiche aufgeteilt, die Informationen, bzw. Werte, enthalten. Durch einen Diskettenmonitor lassen sich diese Werte direkt nach dem Anwählen des Sektors auslesen. Hierbei handelt es sich um den Diskettenmonitor Disk-Zap, der das Einlesen der Diskette und das Editieren ermöglicht. Die Daten können nun direkt in hexadezimaler Schreibweise geändert und neu auf der Diskette abgespeichert werden. So können auch einzelne Tracks von defekten Disketten "repariert" werden.

Den krönenden Abschluß der Diskette setzt ein Texteditor, der den hohen Wert der bisherigen Programme nochmals unterstreicht.

Name: DM A09
Vertrieb: Digital Marketing
Preis: 7,- DM

(Jürgen Seibel/
Robert Marz/br)

Kein Herz für Viren

Wie alle PD-Disketten von Digital Marketing, besitzt auch die DM A04 eine benutzerfreundliche Menüführung, aus dem alle Programme, die sich auf der Diskette befinden, gestartet werden können. Auf der DM A04 sind das insgesamt zwölf Anwender-

programme und fünf Demos. Fangen wir mit dem ersten Programm an:

Der System Z VirusProtector ist ein Virenjäger vom Typ "Kein Herz für Bootblock-Viren". Auf dem Menüscreen hat man die Wahl zwischen vier Optionen: Help, Quit, Write und Read. "Help" bringt eine Hilfsseite hervor, die die einzelnen Funktionen des Protectors erläutert. Mit "Quit" kann man das Programm verlassen und mit "Write" einen speziellen Bootblock auf eine Diskette schreiben. Dieser Bootblock bleibt dann resident im Speicher und überlebt auch einen Reset. Nach einem solchen "Warmstart" liest dieses, sich im RAM befindliche Programm, den Bootblock der in df0: befindlichen Diskette, bevor das AmigaDOS auch nur die Chance dazu hat, und gibt eine Warnung aus, falls sich Viren auf der Diskette befinden. Wenn ein Virus gefunden wurde, besteht die Möglichkeit, ihn zu zerstören und den Virusjäger auf der Diskette zu installieren. Mit "Read" kann man den Bootblock einer beliebigen Diskette einlesen. Folgende Bootblockarten werden anschließend angezeigt:

1. Normal DOS Disk
2. Bootblock in Use
3. SCA-Virus
4. Bytebandit Virus.

Sollte eine der beiden letzten Meldungen erscheinen, so hat man die Möglichkeit, den Virus zu zerstören.

Der TLC-Memory Viruskiller checkt nach seinem Aufruf das gesamte RAM nach eventuell vorhandenen Viren durch und gibt eine entsprechende Meldung aus.

Bootfighter ist ein Virenschutz-Utility, das zehn Funktionen besitzt, deren Namen wohl alle für sich selbst sprechen. So ist es möglich, den Byte-Bandit bzw. den SCA-Virus auf eine Diskette zu kopieren, die sich im Laufwerk befindliche Diskette zu installieren, ihren Bootblock in den Puffer einzulesen, den SCA-Protector bzw. das Programm NOFASTMEM auf den Bootblock der Diskette zu schreiben und die Drive-Nummer zu ändern.

DirUtil III ist eine Weiterentwicklung des bereits auf Diskette A0A besprochenen DirUtility. Die inhaltlichen Änderungen sind geringfügig, das Programm macht jetzt aber einen professionelleren

Eindruck. Die herausragendste Neuerung ist wohl Diskcopy, dessen Name wohl für sich selbst spricht: Es können ganze Disketten kopiert werden.

The Ultimate Space Writer ist ein Programm um Briefe besonderer Art zu schreiben, die man auch als Programm-Intros verwenden kann. Diese Briefe sind allerdings nur dazu gedacht, auf Diskette weitergegeben zu werden. Gibt man also einen Text ein, so merkt sich das Programm jede Cursorbewegung (auch eventuelle Tippfehler und deren Korrektur) und gibt sie später originalgetreu vor einem animierten Hintergrund mit wahlweise einstellbarem Farbblau wieder. Außerdem kann der so erstellte Brief durch Musik unterlegt werden. Optional kann auch einer von zwei verschiedenen Equalizern im unteren Bildschirmviertel angezeigt werden. Dieses Programm ist gut dazu geeignet, um Freunde mit netten Briefen zu überraschen und vor allem zu beeindrucken.

Filecopy V3 sieht wieder einmal dem schon bekannten DirUtility sehr ähnlich, hat auch im großen und ganzen die selben Funktionen sowie einen ähnlichen Bildschirm-aufbau. (Wer hat da wohl von wem geklaut?) Hinzugekommen ist diesmal die Funktion Copy, mit der man einzelne Files oder auch File-Gruppen kopieren kann, nachdem man den Zielpfad angegeben hat.

Das nächste Programm ist ein Sprite-Editor, mit dem bequem Sprites erstellt werden können. Die Sprites können

mehrfarbig dargestellt werden, und es stehen diverse Zeichenfunktionen zur Verfügung. Da das Programm schon etwas älter ist (Baujahr 1986), sollte man auch keine allzu großen Ansprüche stellen, immerhin erfüllt es seinen Zweck zufriedenstellend. Mit dem Programm Show Fonts V1.0 kann man sich Zeichensätze direkt anschauen. In einem Fenster erscheinen verschiedene Icons und ein Beispielsatz, der in dem jeweiligen Font geschrieben ist. Dieser Satz gibt über Name des Fonts, seine Größe und die aktuelle Art seiner Darstellung Auskunft. Die ersten vier Icons sind zum Blättern in den Fonts auf der Diskette gedacht, das fünfte zum Verlassen des Programms und die letzten drei für die untereinander kombinierbaren Textattribute Bold, Italic, und Underline. Dieses kleine Programm kann sehr nützlich sein, wenn man während einer Textverarbeitung Experimente mit Schriften machen will, deren Aussehen man sich nicht merken kann.

Das Programm Zeige Libs listet nach seinem Start alle Libraries und deren Länge auf, die sich auf der Diskette befinden.

Mit dem Menümaker kann ein Disklademenü mit Scrolltexten erstellt werden. Nach Anwählen eines der bis zu acht möglichen Filenamen, führt der Computer dieses File dann auch aus.

Smart-Calc ist ein kleiner Taschenrechner, der die Grundrechenarten beherrscht und zwischen den Zahlensystemen Binär, Oktal, Dezimal

und Hexadezimal umrechnen, beziehungsweise mit ihnen rechnen kann.

Screenblanker 2 löscht den gesamten Bildschirm nach 90 Sekunden, wenn weder die Tastatur noch die Maus benutzt worden sind.

Weiterhin befinden sich auf dieser Diskette fünf Demoprogramme, die von Crackern erstellt worden sind. Alle Demos sind äußerst sehenswert, da sie auf eindrucksvolle Weise zeigen, was grafisch und musikalisch aus dem Amiga herauszuholen ist. Um einen Zeitgeist zu zitieren: "Hier zeigt sich wieder einmal, daß einige Cracker wesentlich mehr drauf haben, als so mancher kommerzielle Spieleprogrammierer".

Name: DM AO4
Vertrieb: Digital Marketing
Preis: 7,- DM

(Robert Marz/
Jürgen Seibel/br/

Ordnung ist das halbe Leben

Das oberste Gebot beim Diskettenhandling heißt Ordnung. Schwierig wird es nur dann, wenn sich im Laufe der Zeit mehrere Diskettenkästen (wenn man überhaupt welche besitzt) mit diversen Programmen gefüllt haben, da man doch relativ schnell die Übersicht verliert.

Hier hilft D-Sort. Es handelt sich dabei um ein Diskettenverwaltungsprogramm, das

den Inhalt einer Diskette einliest und weiterverarbeitet. Das Inhaltsverzeichnis der jeweils eingelegten Diskette wird in den Speicher geladen, worauf sich der Benutzer entscheiden kann, nach welchen Kriterien sortiert werden soll. Hier stehen zwei Sortiermöglichkeiten zur Auswahl, nämlich zum einen Unterhaltungs- und Anwendungssoftware und zum anderen das Sortieren nach alphabetischer Reihenfolge. Ein großer Nachteil dieses Programms liegt jedoch darin, daß das RAM, weitere Diskettenlaufwerke oder Harddisks nicht unterstützt werden. Das Druckmenü verfügt über eine besondere Funktion namens Etti. Etti bedeutet dabei nicht mehr oder weniger als Etikettendruck, denn mittels dieser Option können Diskettenlabels erstellt und anschließend auf dem Drucker ausgegeben werden. Das zu erstellende Etikett unterteilt sich dabei in drei Spalten:

Der erste Bereich ist für den Diskettennamen zuständig; hier können bis zu 12 Zeichen eingegeben werden. Als zweite Spalte folgt die Diskettennummer, deren Länge durch bis zu vier Zeichen dargestellt werden kann. In den dritten Bereich kann ein Kommentar von maximal 15 Zeichen gesetzt werden.

Name: RW 37 - D-Sort
Vertrieb: Wolf Computertechnik
Preis: 6,- DM

(Jürgen Seibel/
Robert Marz/br/

Eine kleine Übersicht der Vertreiber von Public Domain, Free- und Shareware sowie Prüf-vor-Kauf-Programmen (ohne Anspruch auf Vollständigkeit)

A.P.S. Electronic
Sonnenborstel 31
3071 Steimbke
Tel.: 05026/1700

BAVARIANSOFT
Postfach 72
8473 Pfreimd

BELLINGRATH, CHRISTIAN
Hans-Böckler-Str. 55
5860 Iserlohn
Tel.: 02371/24192

DIGITAL MARKETING D.
MÜCKTER
Krefelder Str. 16
5142 Hückelhoven-Baal
Tel.: 02435/2086,428, oder
1295

HIESKE, DIETER
Schillerstr. 36
6700 Ludwigshafen
Tel.: 0621/673105

KEIM, PETER
Vogelsanger Str. 34
5000 Köln 30
Tel.: 0221/520765

MAXON COMPUTER GMBH
Industriestr. 26
6236 Eschborn
Tel.: 06196/481811

TechnikSupport Verlag GmbH
Bundesallee 36-37
1000 Berlin 31
Tel.: 030/8621314

WOLF Computertechnik
Deipe Stegge 187
4420 Coesfeld
Tel.: 02541/2874

Österreich

KÜPPERS, BERND
Felberstr. 7
A-5730 Mittersill
Tel.: 06562/282

PETER RAUSCHERS
COMPUTERSHOP
A-1100 Wien
Weldengasse 41
Tel.: 0222/621535

Hack, Larn, Moria

Hack ist eines der am weitesten verbreiteten PD-Spiele. Es wurde schon auf diversen Computern gespielt, als es noch keinen Amiga gab. Es stand sogar einem kommerziellen Spielprogramm Pate, nämlich Rogue. Aber auch auf dem Amiga erlebte Hack eine ganz besondere Renaissance. Hack wurde adaptiert, und aus dem ursprünglichen Spiel entstanden Larn und Moria.

Hack - Fred Fish Disk 62
Larn - Fred Fish Disc 63
Larn - RPD 28
Moria - Oase 30.
Moria V3.0 - Fred Fish 194

Natürlich ähneln sich alle drei Spiele in ihrem grundsätzlichen Spielaufbau. Der Spieler muß versuchen, einen Krieger durch ein monsterverseuchtes unterirdisches Labyrinth von Gängen zu steuern und dabei alle Wertgegenstände, die dort versteckt sind, einzusammeln. Da es sich bei Hack, Larn und Moria um Rollenspiele handelt, werden die Fähigkeiten der Spielfigur in Form von Zahlenwerten dargestellt. Es gibt zum Beispiel Werte für Kraft und Geschicklichkeit. Je höher Kraft- und Geschicklichkeitswerte sind, desto besser kann die Spielfigur sich gegen angreifende Monster zur Wehr setzen.

Hack, das Ursprungsprogramm, beinhaltet zwar auch die eben genannten Features, präsentiert sich aber von der grafischen Seite sehr spartanisch. Hier werden die Verliese, die es zu erkunden gilt, durch neudefinierte Grafiken dargestellt. Die nächste Hack-Generation hört auf den Namen Larn. Hier findet der Spieler schon eine ganz possierliche Grafik vor. Auch gibt es bei Larn so etwas wie eine Hintergrundgeschichte, die der ganzen Handlung einen Sinn gibt. Ferner gibt es noch Geschäfte, in denen man die Ausrüstung aufstocken kann, eine Handelsstation, in der gefundene Gegenstände verkauft werden können, und eine Bank.

Moria ist wohl die am weitesten ausgearbeitete Hack-Version. Neben einer noch weiter verbesserten Spielgrafik gibt es hier noch zusätzlich Bilder, die das Spiel einleiten bzw. beenden. Während bei

Public-Domain-Spiele-Show

Wer sich auf seinem Amiga mit Public-Domain-Software beschäftigt, wird recht schnell feststellen, daß es hier auch ein weitgefächertes Angebot an Spielprogrammen unterschiedlichster Machart gibt.

Larn die einzelnen Stockwerke nur etwa einen Bildschirm groß sind, fallen die Dungeons bei Moria größer aus. Gelangt der Spieler bei der Erkundung eines Ganges an den Bildschirmrand, schiebt der Computer das Szenario entsprechend weiter. So etwas wie ein echtes Ziel, das es zu erreichen gilt, gibt es nur in Larn. Hier muß in den Verliesen ein Heilserum gegen eine schreckliche Krankheit gefunden werden. Ansonsten haben alle drei Spiele nur das

Ziel, dem Spieler so viel Kurzweil wie möglich zu beschaffen.

Tetrix und Steinschlag

Tetrix - Digital Marketing PD A8D
Steinschlag - Digital Marketing PD A8G

Nicht nur der Name erinnert an das erste Computerspiel aus Rußland. Die grundsätzli-

che Spielidee könnte einfacher nicht sein. In einem rechteckigen Spielfeld tauchen am oberen Ende Blöcke unterschiedlicher Form auf, die langsam Reihe um Reihe nach unten sinken. Während die Blöcke fallen, kann der Spieler sie drehen. Aus diesen Blöcken müssen an der unteren Seite des Spielfeldes geschlossene Reihen gebildet werden. Gelingt es, eine Reihe komplett zu schließen, verschwindet sie, und der Spieler erhält einen Punktebonus gutgeschrieben. Erreicht der Stapel der Blöcke, die nicht zu kompletten Reihen kombiniert wurden, das obere Ende des Spielfeldes, ist die Partie zu Ende.

Das eben beschriebene Spiel liegt in mehreren PD-Versionen vor. Wir haben uns Tetrix und Steinschlag einmal genauer angesehen.

Steinschlag präsentiert sich in einer gediegenen grafischen Gestaltung und wird per Joystick gesteuert. Der Spielablauf bei Tetrix hat seine Macken, dies sei unbestritten. So kann es durchaus vorkommen, daß eine bestimmte Blockform bis zu zehnmal hintereinander erscheint (beim Test tatsächlich passiert). Die grafische Aufmachung von Tetrix ist sehr einfach. Es werden zwei Windows geöffnet, in einem findet das Spielfeld Platz, im gleich nebenan stehenden Window Punktzahl und Level.

Tetrix kann nur über Tastatur gespielt werden.

Die optische Aufbereitung von Steinschlag ist um einiges besser als die Magergrafik von Tetrix. Das Spiel selbst folgt dem eingangs beschriebenen Spielprinzip. Tetrix befindet sich auf Digital Marketing PD A8D und Steinschlag auf Digital Marketing PD A8G.

Ein Spiel um Reichtum und Macht

Monopoly-Kickstart PD Nr. 08

Monopoly ist nach wie vor eines der beliebtesten Gesellschaftsspiele überhaupt. Um so erfreulicher, daß es dieses Spiel auch in einer respektablen Version unter Public Domain gibt. Das in BASIC geschriebene Programm hält

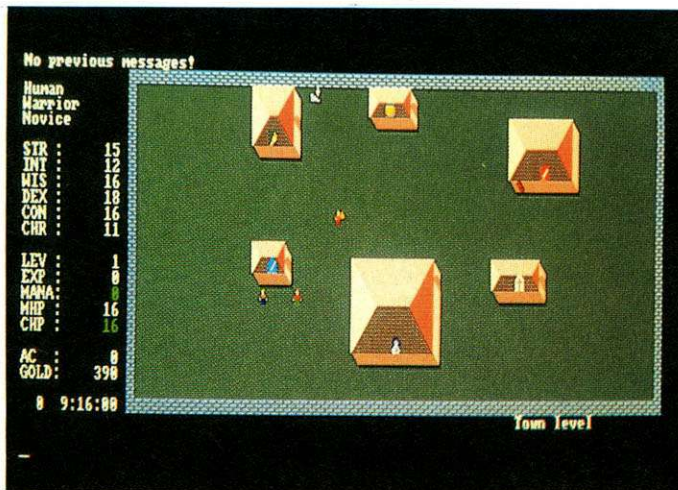


Bild 1. In den Verliesen von Moria erwartet den geneigten Spieler mehr als eine Überraschung



Bild 2. Sieht aus wie Tetris, spielt sich wie Tetris, ist aber kein Tetris. Steinschlag ist eine PD-Version des bekannten Kommerziels

sich eng an die bekannten Monopoly-Regeln.

Auch das Spielbrett, das grafisch recht ansprechend gestaltet ist, orientiert sich am Original. Am Ende jeder Spielrunde, also wenn jeder Spieler einmal gewürfelt hat, fragt das Programm nach, ob die Spieler untereinander Straßen verkaufen wollen. In dieser Phase können aber auch einfach Infos über bestimmte Straßen abgerufen werden. Zum Beispiel wie teuer die betreffende Straße ist und wieviel Mieteinnahmen zu erwarten sind. Wie beim Brettspiel auch, erhält jeder Spieler ein Gehalt, sobald er das 'Los'-Feld passiert.

Alles in allem ist das PD-Monopoly gut gelungen. Der einzige wirklich negative Aspekt ist der Umstand, daß immer nur ein menschlicher Spieler gegen drei Computerspieler antreten kann.

Tiles

Tiles - Digital Marketing PD A80

Es gibt einige Computerspiele, von denen behauptet wird, sie würden süchtig machen. Tiles gehört dazu. Wie bei vielen anderen PD Spielen auch, orientierten sich die Programmierer bei Tiles an einem äußerst erfolgreichen Kommerzprodukt namens Shanghai.

Zu Beginn des Spieles werden 144 quadratische Steine zu einer Pyramide aufgeschichtet. Jeder Stein trägt ein kleines Symbol, das jeweils viermal vorhanden ist. Es gilt nun, jeweils zwei Steine mit gleichen Symbolen aus der Pyramide zu entfernen. Jedoch können die Steine nicht wahllos entfernt werden. Die Steine müssen frei liegen, das heißt, kein anderer Stein darf darauf liegen. Jeder Stein, der aus dem Stapel genommen werden soll, muß, ohne andere Steine zu verschieben, nach links oder nach rechts aus der Pyramide gezogen werden können.

Im Idealfall endet eine Runde mit dem kompletten Abräumen der Pyramide. Es kann aber auch passieren, daß die Steine so ungünstig gemischt wurden, daß ein Rest der Steine übrig bleibt.

Tiles, das übrigens in Modula-2 geschrieben wurde, ist programmtechnisch gut um-

gesetzt. Alles in allem ist Tiles eine echte Bereicherung für jede Public-Domain-Sammlung.

Break Out in der dritten Dimension

3-D Break Out Kickstart PD A2

Das Spiel Break Out, immerhin eines der dienstältesten Spielprogramme überhaupt, präsentiert sich hier in einer neuen Form. Nach vielen kommerziellen Variationen des Break-Out-Spielgedankens ist es erstaunlich, daß eine 3D-Version des Spieles ausgerechnet im Rahmen einer Public-Domain-Reihe erscheint. Sie spielen 3D Break Out in einem quadratischem Raum, der auf dem Monitor dargestellt wird. Tatsächlich basiert dieses Spiel auf der Farbtrennmethode, was be-

deutet, daß Sie eine 3D-Brille benötigen, um den Tiefeneffekt wahrzunehmen. Diese 3D-Brillen haben zwei unterschiedlich eingefärbte Brillengläser bzw. Folien, links rot, rechts blau. Wird diese Brille getragen, nimmt das Auge, das durch das rote Glas blickt, wesentlich mehr Blau wahr, während Rot durch das Glas ausgefiltert wird. Entsprechend verhält es sich mit dem blauen Glas, nur daß hier Rot wahrgenommen und Blau gefiltert wird. Die Bildschirmdarstellung besteht aus dem grundsätzlichen Spielfeld, das zweifach dargestellt wird. Dadurch, daß beide Darstellungen seitlich zueinander versetzt sind, entsteht beim Betrachten mit der 3D-Brille eine Tiefenwirkung. Etwa so, als sei Ihr Monitor leer und an seinen Innenwänden das Spielfeld. Wenn Sie sich für das Spiel interessieren, finden Sie es auf der Pu-

blic-Domain-Disc Kickstart A2. Allein die 3D-Brille müssen Sie sich selbst besorgen.

Der Kampf um Eriador

Der Kampf um Eriador - Oase PD 2

Auf der fernen Welt Eglador bekämpfen sich die Mächte der Dunkelheit seit Urzeiten. Nachdem beide Seiten seit langem um die Vorherrschaft kämpften, kommt es nun zur entscheidenden Schlacht zwischen Gut und Böse. Die Heere der beiden Mächte stehen einander im Lande Eriador gegenüber, und wir stehen am Anfang der alles entscheidenden Schlacht. 'Kampf um Eriador' ist ein Fantasy-Strategiespiel für zwei Spieler. Jeder Spieler übernimmt die Steuerung eines der beiden Heere, die aus verschiedenen Einheiten mit unterschiedlichen Fähigkeiten bestehen. Da gibt es Bodentruppen wie die Axtmänner, die Schwertkämpfer und die Legionäre, und es gibt Luftstreitkräfte, die Adler. Neben diesen Truppenteilen hat jeder Spieler noch zwei spezielle Figuren, den Zauberer und den König. Der Zauberer ist die mächtigste Figur im Spiel, da er magische Fähigkeiten hat. So kann man beispielsweise mit einem Zauber ganze Berge ebnen.

Im großen und ganzen erinnert Kampf um Eriador an eine Kreuzung aus Schach und Sandkasten-Planspiel.

Die Spieler sind abwechselnd am Zug. Jedem Truppenteil steht pro Runde ein bestimmtes Kontingent an Bewegungspunkten zu. Gelangt ein Truppenteil während des Zuges in die Nähe einer gegnerischen Formation, kommt es zum Kampf, und der Computer ermittelt automatisch dessen Ergebnis.

Kampf um Eriador ist grafisch ein echter PD-Leckerbissen, wenn auch der Bildschirmaufbau etwas an Geschwindigkeit zu wünschen übrig läßt. Sogar einige kleine Soundeffekte haben die Programmierer eingebaut. Schade ist nur, daß das Spiel nur zu zweit gespielt werden kann. Sie wissen selbst, nicht immer ist jemand zugegen, der gerade dann Lust auf ein Spiel hat, wenn es Sie gerade reizt.

(hs)

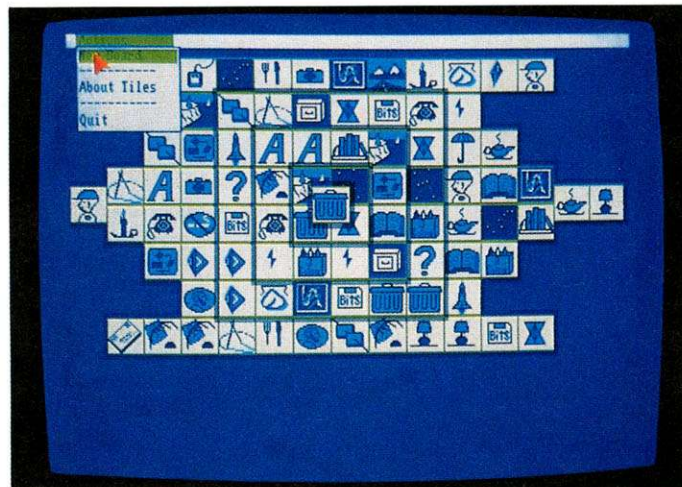


Bild 3. Eine Pyramide aus 144 Steinen, aus der nach einfachen Regeln Steine entfernt werden müssen. Tiles wird auch Sie fesseln



Bild 4. Der Kampf um Eriador entbrennt, an Ihnen ist es, die Schlacht für eine Seite zu entscheiden

SPIELE



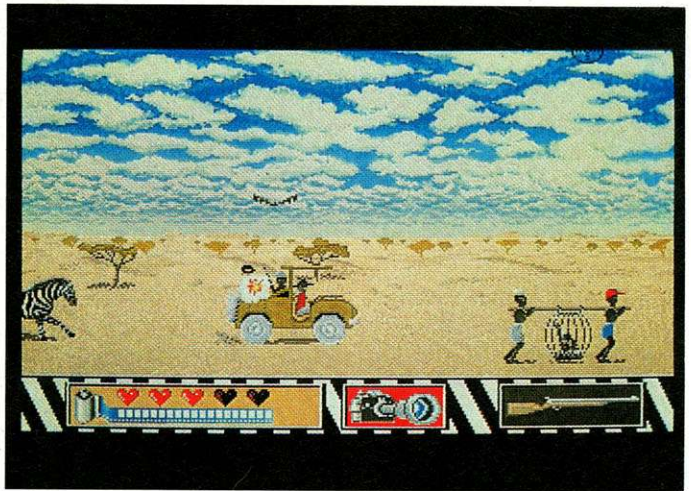
Auch wenn viele Leute, die einen Computer benutzen, immer steif und fest behaupten, ihre Maschinen ausschließlich seriös anzuwenden, sind die Computerspiele eines der am weitesten verbreiteten Software-Genres, die es überhaupt gibt. Ob nun als gekauftes Produkt oder als Raubkopie, die Computerspiele und ihre Themen sind Legion.

Aber obwohl die Spiele so weit verbreitet sind und so oft benutzt werden, ist gerade das Computerspiel ständig im Verdacht, der geistigen Gesundheit des Anwenders abträglich zu sein. Wer sich nicht mit Computern auskennt und unvoreingenommen an die Spielszene herangeht, kann sich dieses Eindrucks ob der großen Menge Unsinn, die hier produziert wird, kaum erwehren.

Es ist nicht Gold, was glänzt, aber auch nicht alles, was spontan einen mageren Eindruck macht, ist schlecht. Mitunter verbergen sich in den unscheinbarsten Verpackungen die feinsten Spiele. Ein solches Urteil läßt sich aber erst dann fällen, wenn das Spiel genauer betrachtet wurde.

Da es bisher keinen Softwarekauf auf Probe gibt, hat der Verbraucher auch nur in den seltensten Fällen die Gelegenheit, das Produkt, mit dem er liebäugelt, in Augenschein zu nehmen. Er kauft sozusagen die Katze im Sack. Ganz können wir Ihnen die Entscheidung, welches Spiel für Sie interessant ist, natürlich nicht abnehmen, aber wir können Ihnen helfen. Unsere Spieltests verraten Ihnen, ob der neue Fugsimulator auch wirklich einen Flug simuliert oder nur einen flachen Hüpfen. Hier erfahren Sie mehr über neue Amiga-Spiele. In diesem Sinne wünschen wir Ihnen gute Unterhaltung bei der Lektüre unserer Spiele-Tests.

(hs)



Fotosafari mit Ihrem Amiga. Wer bringt die besten Schnappschüsse mit nach Hause?

Safari Guns

Afrika und Urlaub, verbindet man diese beiden Komponenten miteinander, ergibt sich daraus fast zwangsläufig eine Safari.

Waren früher Urlauber und deren Jagdambitionen eine große Gefahr für wildlebende Tiere, so sind dies heute die Wilderer. Safari Guns verwandelt Ihren Amiga in einen Safarijeep. Sie als Spieler sind mit einem Fotoapparat und einem Gewehr ausgerüstet. Nun gilt es einerseits, den Film im Fotoapparat mit gelungenen Schnappschüssen zu füllen, andererseits mit dem Gewehr die immer wieder auftauchenden Wilderer zu vertreiben. Auf dem Monitor präsentiert sich das Ganze als horizontal scrollender Ausblick, gerade so, als würden Sie aus dem Seitenfenster eines fahrenden Jeeps blicken. Durch die vorbeigleitende Landschaft streifen wilde Tiere, Wildhüter und andere Safarigesellschaften, die fotografiert, aber nicht abgeschossen werden dürfen. Tut man's doch, gibt es Punktabzug. Für jedes gelungene Foto hingegen gibt es Punkte. An dieser Stelle kommen die Wilderer ins Spiel, die Ihre Aktivitäten nur ungern sehen und versuchen, Sie zu erschießen. Zumeist halten Sie sich hinter Steinen und Baumstümpfen versteckt und eröffnen das Feuer auf Sie, wenn Ihr Jeep an ihrem Versteck vorbeifährt. Dann muß der Spieler vom Fotoapparat zur Flinte umschalten und dem Wilderer mit einem gezielten Schuß zuvorkommen. Ist der ganze Film mit Bildern voll, erfolgt deren Bewertung

in Form einer nett gemachten Diaschau. Dann geht es zur nächsten Safari, die durch eine ganz andere Landschaft führt, als die erste. Die Aufgaben, die Sie erwarten, sind jedoch gleichgeblieben. Safari Guns ist ein nettes Spiel, welches das martialische Spielkonzept von 'Operation Wolf' gut variiert. Technisch mindestens ebensogut umgesetzt, ist Safari Guns ein nettes Ballerspiel, das ohne wüsten militärisches Säbelgerassel auskommt. (hs)

AMIGA DOS Blitzlicht

Name: Safari Guns
Hersteller: NO Productions
Quelle: Fachhandel
Preis: Stand bei Redaktionsschluß noch nicht fest.

Positiv:

- gute Grafik
- sauberes Scrolling
- nette Spielidee

Negativ:

- lange Ladezeiten

AMIGA DOS
URTEIL

GRAFIK SOUND MOTIVATION

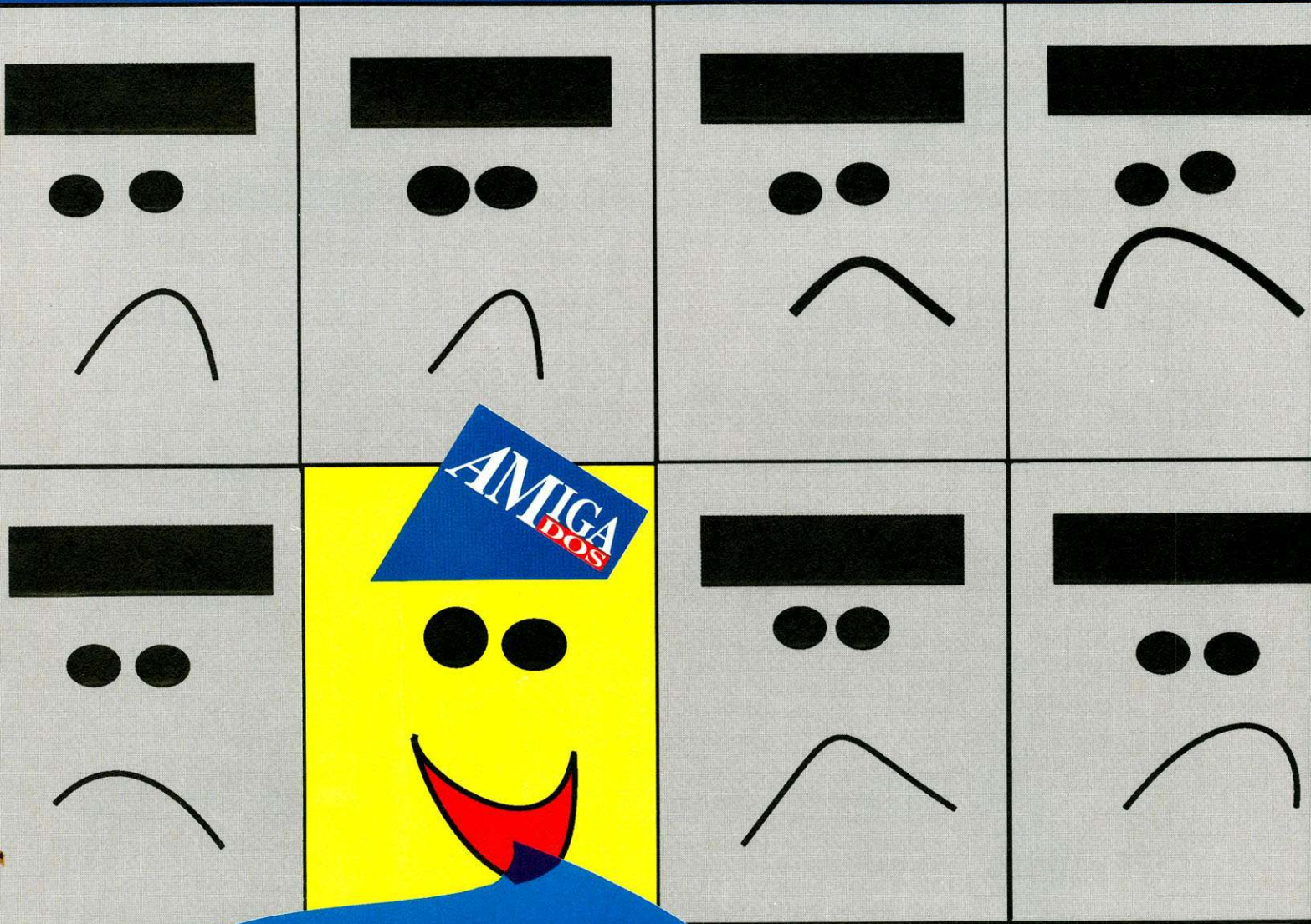
7

6

6



ABONNEMENT



**INFORMATIONEN AUS
ERSTER HAND**

**Ein Abonnement ist
praktisch und bequem.**

Widerrufsrecht

Jeder Abonnent hat das Recht, seine Bestellung innerhalb einer Woche beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich zu widerrufen. Die rechtzeitige Absendung des Widerrufschreibens genügt zur Fristwahrung.

**AMIGA DOS
kostet im Abonnement:**

Im Inland bzw. West-Berlin:

6 Ausg. = 35,- DM

12 Ausg. = 70,- DM

Im europäischen Ausland:

6 Ausg. = 50,- DM

12 Ausg. = 100,- DM

Im außereuropäischen Ausland:

6 Ausg. = 60,- DM

12 Ausg. = 120,- DM

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag • Postfach 250 • 3440 Eschwege





Steuern Sie die Laser Squad durch gefährliche Einsätze – strategisches Denken ist angesagt

Laser Squad

Die Laser Squad ist eine Eliteeinheit, die immer dann zum Einsatz kommt, wenn die Diplomatie versagt hat. Ob es sich um eine Entführung handelt oder die Suche nach einem intergalaktischen Schwerverbrecher, die Laser Squad holt die Kastien aus dem Feuer.

Was sich hier ein wenig nach einem Actionspiel anhört, ist tatsächlich ein Strategiespiel. Der Spieler übernimmt die Kontrolle jedes der einzelnen Lasersquad-Teammitglieder. Vor Antritt der Mission müssen die Teammitglieder erst einmal ausgerüstet werden. Zum Kauf von Waffen und Sprengstoff steht eine bestimmte Summe Geld zu Gebote. Sobald das erledigt ist, kann sich die Laser Squad ins Kampfgetümmel stürzen. Dies geschieht durch das Postieren der einzelnen Kämpfer auf dem Kampfgebiet. Laser Squad präsentiert dem Spieler fünf unterschiedliche Missionen, von denen jede in einem anderen Szenario spielt. Sind die Teammitglieder strategisch um den Zielort gruppiert worden, beginnt das eigentliche Spiel, das Zug um Zug abläuft. Während eines Zuges steht für jeden Kämpfer ein bestimmtes Kontingent von Actionpoints (Bewegungspunkte) zur Verfügung. Jede Aktion, die ein Kämpfer tätigen kann, kostet Actionpoints. Hat der Spieler alle seine Kämpfer die gewünschten Aktionen vollziehen lassen, folgt der Zug der gegnerischen Einheiten.

Sicher, hier werden Granaten, schwere Laser und anderes

martialisches Gerät eingesetzt, aber trotzdem bedarf es, um eine Partie Laser Squad erfolgreich zu beenden, eines scharfen strategischen Verstandes, schnelle Reaktionen sind fehl am Platze.

(hs)

AMIGA DOS Blitzlicht

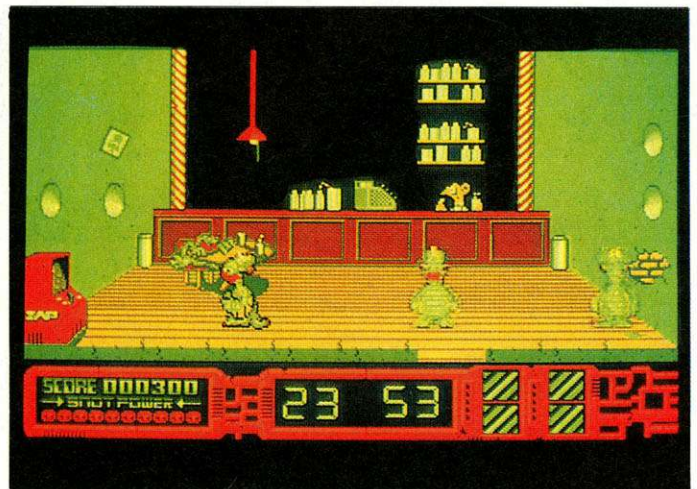
Name: Laser Squad
Hersteller: Blade
Quelle: Fachhandel
Preis: 69,95 DM

Positiv:

- verschiedene Szenarien
- ein oder zwei Spieler
- Option
- starker Computergegner

Negativ:

- kein Construction Set enthalten



Steuern Sie den Schleimkugeln schleudernden Gilbert auf der Suche nach den verlorenen Teilen seines Raumschiffs

Gilbert – Escape from Drill

Gilbert, ein niedliches grünes Monster, machte einen kleinen Ausflug durch die nähergelegenen Teile der Galaxis. Schon bald zeigte sich, daß er wohl besser kein Leihraumschiff der 'Wet Fish Taxi Corp' hätte nehmen sollen. Ein Maschinenschaden am Haupttriebwerk zwang Gilbert zu einer Notlandung auf dem entlegenen Planeten Drill.

Das Raumschiff wurde bei der unsanften Landung in fünf Teile zerschmettert. Diese Einzelteile muß Gilbert finden und wieder zusammensetzen, wenn er Drill wieder verlassen will. Daß die Bewohner des Planeten von kleinen grünen Monstern wenig halten, stellt Gilbert spätestens dann fest, wenn sich die unterschiedlichen Bewohner des Planeten in bester Kamikazemanier auf ihn stürzen. Glücklicherweise ist Gilbert nicht ganz wehrlos, er kann Schleimbällchen auf die Angreifer werfen, allerdings nur, solange sein Schleimvorrat reicht. Sind Gilberts Vorräte erschöpft, kann er eines der Geschäfte auf dem Planeten aufsuchen und dort ein Spielchen an einem der diversen Telespielautomaten wagen.

Gilbert ist mal wieder ein Such- und Sammelspiel äußerst konventioneller Machart. Der Spieler steuert Gilbert durch die unterschiedlichen Landschaften von Drill. So wandert er durch Städte, Sümpfe und Unterwasserzonen, wobei er ständig durch die Bewohner von Drill behindert wird. Einziger Lichtblick

sind die kleinen Subspiele, die Gilbert in den Geschäften der Stadt spielen kann. Ansonsten – Tiefschlafgefahr!

(hs)

AMIGA DOS Blitzlicht

Name: Gilbert - Escape from Drill
Hersteller: Enigma Variations
Quelle: Fachhandel
Preis: 79,95 DM

Positiv:

- Spiele im Spiel

Negativ:

- magere Grafik
- mäßige Animationen
- langweiliger Spielverlauf

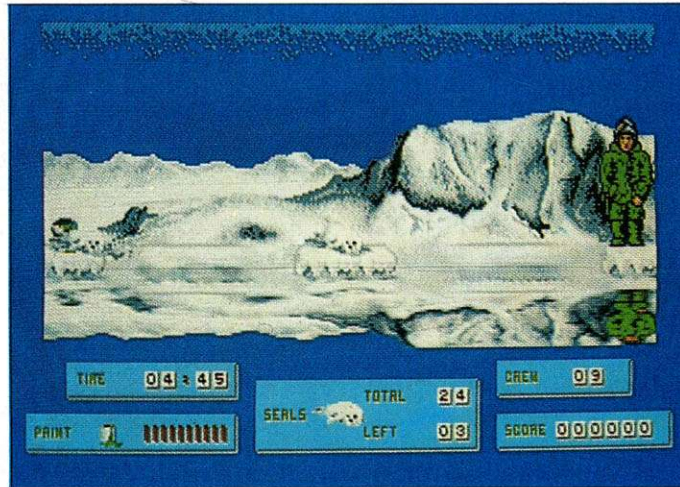


Ob die Greenpeace-Aktivisten nun Fabrikschlote erklettern und von dort Protesttransparente aufrollen oder ob sie Schiffe an der Säureverklappung im Meer hindern, Greenpeace-Aktionen sind fast immer spektakulär. Vor diesem Hintergrund spielt ein neues Computerspiel, in dem der Spieler die halsbrecherischen Aktionen nachvollziehen kann. Hier müssen keine intergalaktischen Bösewichter bekämpft werden, sondern Umweltsünder und deren hinterlassene Dreckhaufen.

Umweltschutz als Actionspiel

Rainbow Warrior besteht aus sechs verschiedenen Teilspielen, die jeweils einen gefährdeten Umweltbereich darstellen. Da muß der Spieler verhindern, daß im Meer Säure verklappt wird oder Abwasserleitungen verstopfen. Es gibt einen Spielteil, in dem es darum geht, an einem Fabrikschlot ein Transparent, das auf den sauren Regen aufmerksam macht, zu befestigen. Oder der Spieler muß mit Farbe die Felle junger Seehunde unbrauchbar machen, so daß Robbenjäger sie nicht mehr verwenden können und die Tiere am Leben lassen. Sogar eine Mission in der Antarktis erwartet den Spieler. Dort muß er gegen das stetig wachsende Ozonloch kämpfen. Jede dieser Aufgaben muß der Spieler mit dem Joystick in der Hand meistern.

In einem Auswahlbildschirm kann der Spieler entscheiden, welche der Missionen er in Angriff nehmen möchte. Beim Verstopfen von Abflüssen steuert der Spieler einen Delphin, der einem Taucher den Weg zu den versteckten Dreckrohren zeigen muß. Ist der Taucher bei einem Rohr angekommen, hat er eine Weile damit zu tun, es zu verstopfen. Währenddessen kann der Delphin wieder an die Oberfläche zurück, um dort nach Luft zu schnappen. Geht der Luftvorrat unter Wasser aus, stirbt der Delphin, und der Taucher findet den Weg zurück zum Schiff nicht. Auch wenn der Spieler verhindern soll, daß Dünnsäure verklappt wird, ist Joystick-Action angesagt. Man muß mit einem Schlauchboot neben das betreffende Schiff fahren und es



Mit einem Farbspray kann der Spieler die Felle junger Robben unbrauchbar machen. Umweltschutz als Telespiel

Rainbow Warrior

In einer Welt, in der es scheinbar keine Abenteuer mehr gibt, beginnt die größte Herausforderung vor unserer aller Haustür. Unsere gefährdete Umwelt braucht unsere Hilfe. Überall auf der Welt verschwenden wir bedenkenlos unser Kapital für die Zukunft. Das hier wirklich noch Abenteuer warten, beweist die Tätigkeit von Greenpeace.

entern. An Bord gilt es nur noch, die Kräne zu besetzen, die die Säurefässer ins Meer werfen.

Mal so, mal so...

Generell hat uns Rainbow Warrior, nicht zuletzt ob seines interessanten Themas, gut gefallen. Einige der gestellten Aufgaben sind auch dazu angetan, sowohl viel Spielspaß zu bereiten, als auch Verständnis für Greenpeace-Aktionen zu wecken. Andere Spielaufgaben allerdings sind so abstrakt dargestellt, daß das betreffende Teilspiel mit dem zugrunde liegenden Thema nur noch wenig gemein hat. Z.B. wenn es um das Ozonloch in der Antarktis geht. Hier muß der Spieler mit Schneebällen herumfliegende Spraydosen und andere FCKW-Erzeuger abschießen. Werden die Spraydosen nicht getroffen, fliegen Sie zum oberen Bildschirmrand und 'bohren' Löcher in die Ozonhülle. Durch diese Löcher kommen dann schädliche Strahlen, die dem Spieler,

läuft er zu lange darin herum, den Garaus machen. An diesen Stellen, von denen es im Spiel mehrere gibt, fragt man sich unwillkürlich, was das soll. Auch steht jedes Teilspiel für sich alleine. Der Spieler wählt in einem großen Menüscreen das Teilspiel aus, auf das er gerade Lust hat. Er kann also auf die schnelle ein paar Robben in der Arktis retten und wenige Minuten später auf einen englischen Fabrikschlot klettern. Es entsteht unwillkürlich der Eindruck, die Rainbow Warrior, das Greenpeace-Schiff, könne an allen Orten gleichzeitig sein.

An keiner Stelle des Spiels taucht der Gedanke auf, daß die vielen dargestellten Umweltprobleme nur Aspekte eines großen globalen Übels sind. Der hier dargestellte Umweltschutz erinnert eher an ein nettes Abenteuer, an dem man teilnimmt, solange es Spaß macht. Und macht es keinen Spaß mehr, soll es eben jemand anders machen.

Rundherum betrachtet, ist Rainbow Warrior ein ganz passables Spiel geworden.

Der Umweltgedanke, der hinter dem Spiel stehen soll, geht leider recht bald verloren. Zwar gibt das Handbuch detailliert Aufschluß über alle möglichen Greenpeace-Aktionen. Die Hintergründe, warum diese Aktionen überhaupt stattfanden, werden nur recht knapp angesprochen.

Denn eigentlich sind bei der Umweltproblematik nicht nur die Resultate interessant, sondern auch, wie es überhaupt dazu kommen konnte. Obwohl das Label Micro Style von seinen Initiatoren als eine Reihe konzipiert wurde, in deren Rahmen Software für Erwachsene realisiert werden soll, wird Rainbow Warrior auch von vielen Kindern und Jugendlichen gespielt werden. Das gerade hier mit einer Erläuterung ökologischer und wirtschaftlicher Zusammenhänge von Umweltdelikten viel Aufklärungsarbeit hätte geleistet werden können, ist klar. Aber leider blieb diese Möglichkeit ungenutzt.

Rainbow Warrior ist ein passables, technisch gut gemachtes Spiel. Weniger gut gefallen hat uns der Umstand, daß Umweltschutz hier zu einer Marketing-Idee verkommt.

Naja, irgendwie mußte ja auch Greenpeace sein neues Schiff bezahlen.

(hs)

AMIGA DOS Blitzlicht

Name: Rainbow Warrior
Hersteller: Micro Style
Quelle: Fachhandel
Preis: 89,- DM

Positiv:

- gute Grafik
- gute Animation
- reizvolle Spielatmosphäre

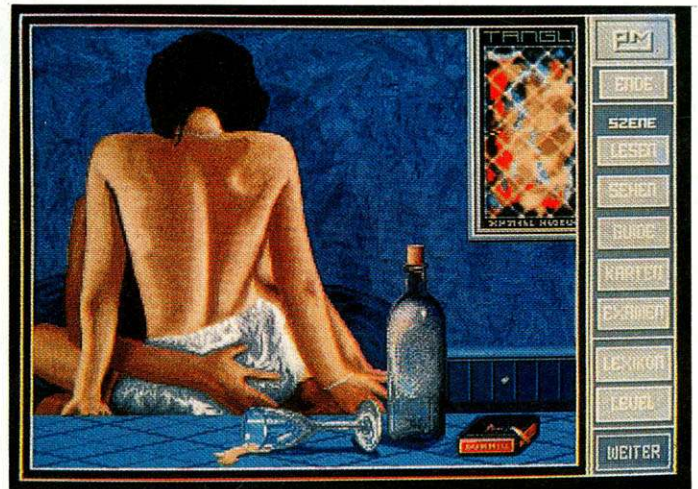
Negativ:

- teils zu abstrakt
- wird recht schnell langweilig





Ein neues Labyrinthspiel. Keine Monster, keine Aliens und trotzdem viel Spielspaß!



Trotz der liebevollen Handlung führt die Dame Böses im Schilde

Rock'n' Roll

Spätestens seit Pac Man sind Labyrinth und Irrgärten ein nicht mehr wegzudenkender Bestandteil von Computerspielen. In Rock'n' Roll ist der Irrgarten wesentliches Spielelement. Es gilt einen Ball zu lenken, ihn wohlbehalten durch ein Labyrinth zu steuern und dabei den vielen offenen und versteckten Gefahren zu entkommen.

Tatsächlich, keine Monster, keine Aliens und trotzdem ist Rock'n' Roll absolut kein langweiliges Spiel. Das Spielziel ist einfach: Einen kleinen Ball mittels der Maus vom Eingang bis zum Ausgang verschiedener Irrgärten zu steuern. Was sich einfach anhört, gestaltet sich schwierig, da der Parcours von Hindernissen unterschiedlichster Art nur so wimmelt. Da gibt es verschlossene Türen, die nur mit den richtigen Schlüsseln geöffnet werden können, Teleporter, die den Ball an eine andere Stelle des Irrgartens versetzen, Löcher im Boden, Eisflächen und vieles andere mehr. Doch ganz unbeholfen ist der Ball nicht, denn durch Einsammeln verstreut herumliegender Geldstücke kann ein kleines Vermögen aufgebaut werden, für das der Spieler seinen Ball mit Extras ausstatten kann. Da gibt es Spikes, mit denen auf Eisflächen besser manövriert werden kann, Bomben, die Wände niederreißen und so andere Teilstücke des Labyrinthes zugänglich machen, oder einfach Bodenplatten, die über Löcher gelegt werden können. Schon die Intro, in der man einer Rockband, bestehend aus lauter niedlichen grünen

Gnomen, bei der Arbeit zusehen kann, macht Spaß. Ich muß zwar zugeben, daß Labyrinthspiele auf mich einen besonderen Reiz ausüben, weswegen Rock'n'Roll mir auf Anhieb gefiel.

(hs)

AMIGA DOS Blitzlicht

Name: Rock'n' Roll
Hersteller: Rainbow Arts
Quelle: Fachhandel
Preis: ca. 80 DM

Positiv:
- absolut friedfertig
- langer Spielspaß
- gute Musik

Negativ:
- nur über Maus steuerbar



Die Stadt der Löwen

Singapur, die farbige und exotische Hauptstadt Malaysias ist nicht nur Ziel vieler Urlauber aus aller Welt. Mit Stadt der Löwen wurde Singapur der Schauplatz eines Abenteuerspiels.

Adventures leben im wesentlichen von ihrer Story, und die hat im vorliegenden Fall echte Thrillerqualitäten. Es geht um eine geheimnisvolle Verschwörung, deren Ränke sich fast um die ganze Welt spannen. Ein japanischer Industrieller wird in London tot aufgefunden. Todesursache angeblich Herzversagen. Die französische Reporterin Christine Monier beginnt in dem Fall zu recherchieren. Sie bringt in Erfahrung, daß das Herzversagen den Japaner während des Besuches einer Prostituierten ereilte. Die betreffende Dame reiste kurz darauf nach Singapur ab. Christine kann ihren Chefredakteur davon überzeugen, daß an der Geschichte mehr dran ist als ein einfaches Herzversagen, und reist ebenfalls nach Singapur...

und Mausclick gesteuert. Die Geschichte fängt etwas behäbig mit niedrigem Schwierigkeitsgrad an und steigert sich im Spielverlauf ständig.

(hs)

AMIGA DOS Blitzlicht

Name: Stadt der Löwen
Hersteller: Software 2000
Quelle: Fachhandel
Preis: 119,- DM

Positiv:
- gute Grafik
- präzise recherchiert
- interessante Story

Negativ:
- kein Sound



BEST – MODEMS

BEST 2400 PLUS

4 - Speed-vollduplex 300,1200, 1200/75 (BTX), 2400
V.21, V.22, V.22 bis, V.23 (BTX), Bell 103 & 212 A

4 4 7,-

BEST 1-2-3

Vollduplex 300, 1200, 1200/75 (BTX)
V.21, V.22, V.23 (BTX), Bell 103 & 212 A

3 2 9,-

BEST 1200 PLUS

Vollduplex 300, 1200
V.21, V.22, Bell 103 & 212 A

2 8 5,-

RS-232 Kabel: 20,-
Konverterstecker RJ-11 auf TAE-6: 12,-

Die Inbetriebnahme dieser Modems am öffentlichen Postnetz der
BRD einschl. Berlin-West ist verboten und unter Strafe gestellt.

Alfonso Sanz - Software & Beratung
Rabanus Maurus Str. 8 / 6415 Petersberg / (0661) 61748



zum Ortsstarif

2 MB-Rambox A1000 mit Bus 2 MB bestückt	DM 849,-
Profex 2-MB-Rambox vollbest. mit Bus A 500	DM 749,-
Amstrad LQ 3500 Letter-Quality-24-Nad.-Drucker	DM 599,-
Aztec C Professional System V3.6	DM 279,-
Digi View Gold V3.0 für A500/2000	DM 279,-
Golem 3.5-Zoll-Laufwerk	DM 249,-
Vortex 20 MB-Festplatte A500/1000	DM 899,-
Vortex 60 MB-Festplatte A500/1000	DM 1499,-
TDI-Modula Developers Version V3.01	DM 199,-
MCC-Assembler+Shell+Toolkit komplett	DM 99,-
Amiga 2000 8 MB-Karte mit 2 MB bestückt	DM 949,-
Amiga 500 1.8 MB Erweiterung intern/Uhr	DM 699,-
Balance of Power / The Pawn / Thexterje	DM 49,95
BTX/VTX-Manager V2.2/BTX-Decoder dt.	DM 219,-
Kostenlose Prospektete auch für ST und IBM von	



C W T G Joachim Tiede
Bergstr. 13 * * * 7109 Roigheim
Tel./BTX 0 62 98/30 98 von 17-19 Uhr

DONAU-SOFT

24 h-Schnellversand

Ihr Amiga-PD-Partner

● ab 2,50 DM ●

Alle gängigen Serien sind
lieferbar

Einzeldisk	4,50 DM
ab 10 Disk	4,— DM
ab 50 Disk	3,50 DM
ab 100 Disk	3,30 DM
ab 200 Disk	3,— DM
bei Serienabnahme: ab 2,50 DM	

Preise incl. 3,5" DD-Disks
— Mit Qualitätsgarantie —

Wir kopieren nur mit doppeltem Verify.

Alle Disks sind:

- 100 % Virus- und Error frei
- etikettiert.

Leerdisketten 3,5" 2 DD

NoName 100 %	ab 1,60 DM
Markendisk	ab 1,95 DM

3 ausführliche Katalogdisketten
mit Kurzbeschreibung aller
Programme gegen 10,— DM
(V-Scheck/Briefmarken) anfordern!

gratis zu unseren Katalogen:

Viruskiller, CLJ-Wizard + Turbo Backup

Das große Amiga-PD-Handbuch

Band I-IV + alle 42 Disks

+ 3 Katalogdisketten

(Einzelpreise erfragen)

325,-

Pakete für Einsteiger und Anwender
(jeweils 10 Disketten)

Einsteiger 1,2; Spiele 1,2,3;

Sound; Grafik; Modula II

jedes Einzelpaket

3 Pakete nach Wahl nur

35,— DM

99,— DM

Floppy 3 1/2" int.

Floppy 3 1/2" ext.

Floppy 5 1/4" ext.

abschaltbar
mit allen
Extras

180,— DM

235,— DM

269,— DM

+ DM 5,— bei Vorkasse, + DM 8,— bei Nachnahme

Ausland: + DM 10,— (nur Vorkasse)

MAIK HAUER

Postfach 1401, 8858 Neuburg Fax: 08431/49800

Tel.: 08431/49798 (bis 22 Uhr) BTX: *Donau-Soft #

TOOLBOX EDITION 68000

Wer seinen Computer oder Anwendungen programmieren will, findet hier Tips, Anregungen und vor allem Anwendungen erster Güte.

Das Projekt Fraktal wartet mit einem Weltrekord auf:
Das "Apfelmännchen" wird in 7 Sekunden auf den Bildschirm eines Amiga 2000 gezeichnet!! Da waren Hexenmeister am Werk.

Viele Reviews informieren über neue Programmierer-Software und helfen dem Leser zum Überblick über interessante Neuerungen.

Natürlich gibt es die Programme des Heftes auch in der DATABOX. DATABOX ist der Software-Service, der bei DMV Standard ist. Wem das Tippen zu fehlerträchtig ist, der bestellt die DATABOX zum Heft.

Aus dem Inhalt:

PROJEKT FRAKTAL — Portierung in M2Amiga
WELTREKORD — Apfelmännchen in 7 Sekunden
ACU — Mausgesteuerter Compiler
C-Compiler-Oberfläche
Universeller Multitasking-FileRequester in M2Amiga
Undercover Viruskiller mit eigenem Task
Do-Request — Trickreicher AutoRequester
AmigaDOS — Fenster mit Zwitterereigenschaften
M2Amiga-Inline-Code per Programm

DATABOX Amiga Best.-Nr. 3502 29,— DM*

TOOLBOX EDITION 68000 Best.-Nr. 3501
Sonderpreis für AMIGA-DOS-Leser 9,— DM*

* Unabhängig von der Anzahl der bestellten Produkte berechnen wir für das Inland 4,— DM bzw. für das Ausland 6,— DM Porto und Verpackung.

Bitte benutzen Sie die Bestellkarte

DMV-Verlag · Postfach 250 · 3440 Eschwege

Nr. 1/88-89

18,— DM 145 ÖS 18 sfr

DMV Verlag

EDITION

TOOLBOX
EDITION 68000

TOOLBOX

· Amiga · Atari · 68000 · Macintosh ·

AMIGA:

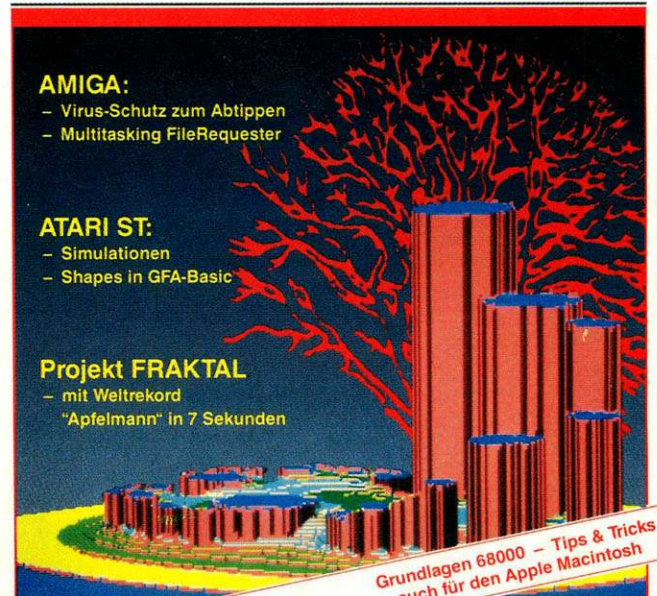
- Virus-Schutz zum Abtippen
- Multitasking FileRequester

ATARI ST:

- Simulationen
- Shapes in GFA-Basic

Projekt FRAKTAL

- mit Weltrekord
"Apfelmännchen" in 7 Sekunden



Grundlagen 68000 — Tips & Tricks
auch für den Apple Macintosh



Ein Actionspiel vom Allerfeinsten. Hier bekommen Sie Animationen zu sehen, wie sonst nur in der Spielhalle

The Shadow of the Beast

Vor vielen Jahren stahlen die Schergen des Beast Lords ein unschuldiges Kind aus seiner Wiege. Hinter den düsteren Mauern des mächtigen Tempels Necropolis wurde das Baby den Magiern des Beast Lords übergeben. Mit geheimen Elixiern und Salben veränderten sie das Kind. Aus dem kleinen Menschen wurde eine hirnlose Bestie, die dem Willen des Bösen Lords bedingungslos ergeben war.

Doch irgendwann dämmerte der Tag, an dem alles anders werden sollte. Die Erinnerungen der Kreatur, die einst ein Mensch gewesen war, kehrten zurück. Das Wesen erkannte, was es einst gewesen war, was es nie wieder würde sein können: ein Mensch. Und die Kreatur erkannte, wer ihr all dies angetan hatte. Wut und Haß erwachten und die Bestie hatte ein neues Ziel: Rache.

Aus dieser wilden und märchenhaften Hintergrundgeschichte entwickelten die Programmierer Martin Edmondson und Paul Howarth ein Actionspiel, das seinesgleichen sucht. Der Spieler steuert das Biest, das einst ein Mensch war, auf seinem Rachefeldzug. Die Suche nach dem Beast Lord beginnt irgendwo auf einer weiten Ebene. Schon dort treiben sich viele böse Wesen herum, die im Dienste des Lords stehen und nichts anderes im Sinne haben, als

die Rache des Biestes zu vereiteln. Dieses ist aber nicht wehrlos, denn als es noch in den Diensten des Lords stand, war Kämpfen seine einzige Aufgabe. Allein seine Fäuste sind tödliche Waffen. Auf der Ebene befinden sich auch die Eingänge zu verschiedenen, versteckten Schlupfwinkeln des bösen Lords. Ein Brunnen erweist sich als Eingang zu einem dieser Schlupfwinkel. Eine genauere Untersuchung zeigt jedoch, daß hier ein Schlüssel benötigt wird, um Zutritt zu erhalten. Wenige hundert Meter entfernt findet das Biest einen riesigen, verrottenden Baum, in dem sich eine Tür befindet, der Eingang zu einem weiteren Schlupfwinkel des Lords. Eine Treppe im Inneren des Baumes führt in die feuchte Dunkelheit hinunter. Baumwurzeln bedecken die Wände und winden sich über die vielen Ebenen der Höhle. Auch hier wimmelt es von den Un-

geheuern des Beast Lords, die mit Schwertern, Äxten und Feuerbällen bewaffnet, den Rachesuchenden schon erwarten. Doch das Biest weiß sich zu wehren und findet tief in der Kaverne einen goldenen Schlüssel.

Grafik - Sound - Spielwitz, alles nur vom Feinsten

Psygnosis gehört zu jenen Softwarehäusern, die zuerst den Trend hin zum Amiga erkannt haben und ihm Rechnung trugen. Dementsprechend liest sich auch die Liste der Spiele, die uns dieses Haus bisher bescherte: Deep Space, Barbarian, Obliterator und andere mehr. Auch wenn all diese Programme schon sehr gut aussahen, so setzt Shadow of the Beast neue Maßstäbe. Schon während des Ladevorgangs präsentiert sich Shadow of the Beast mit Bil-

dern, die ihresgleichen suchen. Durch schnelles, aber ruckelfreies Parallaxscrolling auf nicht weniger als elf verschiedenen Ebenen, entsteht der Effekt räumlicher Tiefe. 132 verschiedene Monster, die zum Teil größer als ein halber Bildschirm sind, sorgen für Abwechslung. An bestimmten Stellen des Spiels warten mächtige Supermonster, die auch das Biest mit seinen stählernen Fäusten nicht besiegen kann. Dann gilt es, eine Extrawaffe zu finden, die hier hilft.

Um an die heranzukommen, muß allerdings ein weiteres Großmonster, nämlich jenes, das die entsprechende Waffe bewacht, besiegt werden. Glücklicherweise bewegen sich die Wächter der Waffen nicht, sondern speien nur Feuer oder beschäftigen sich mit anderen, ähnlich destruktiven Praktiken.

Jedesmal, wenn das Biest eine neue Örtlichkeit betritt, wird während des Nachladens eine fein gezeichnete Grafik eingeblendet. Nur selten gab es Spiele, die solch eine stimmungsvolle Musik aufwiesen, wie Shadow of the Beast. Leider kann kein aktueller Spielstand abgespeichert werden, was sich mitunter sehr schmerzlich bemerkbar macht. Ansonsten ist Shadow of the Beast absolut Spielhallenqualität.

(hs)

AMIGA DOS Blitzlicht

Name: Shadow of the Beast
Hersteller: Psygnosis
Quelle: Fachhandel
Preis: 89,- DM

Positiv:

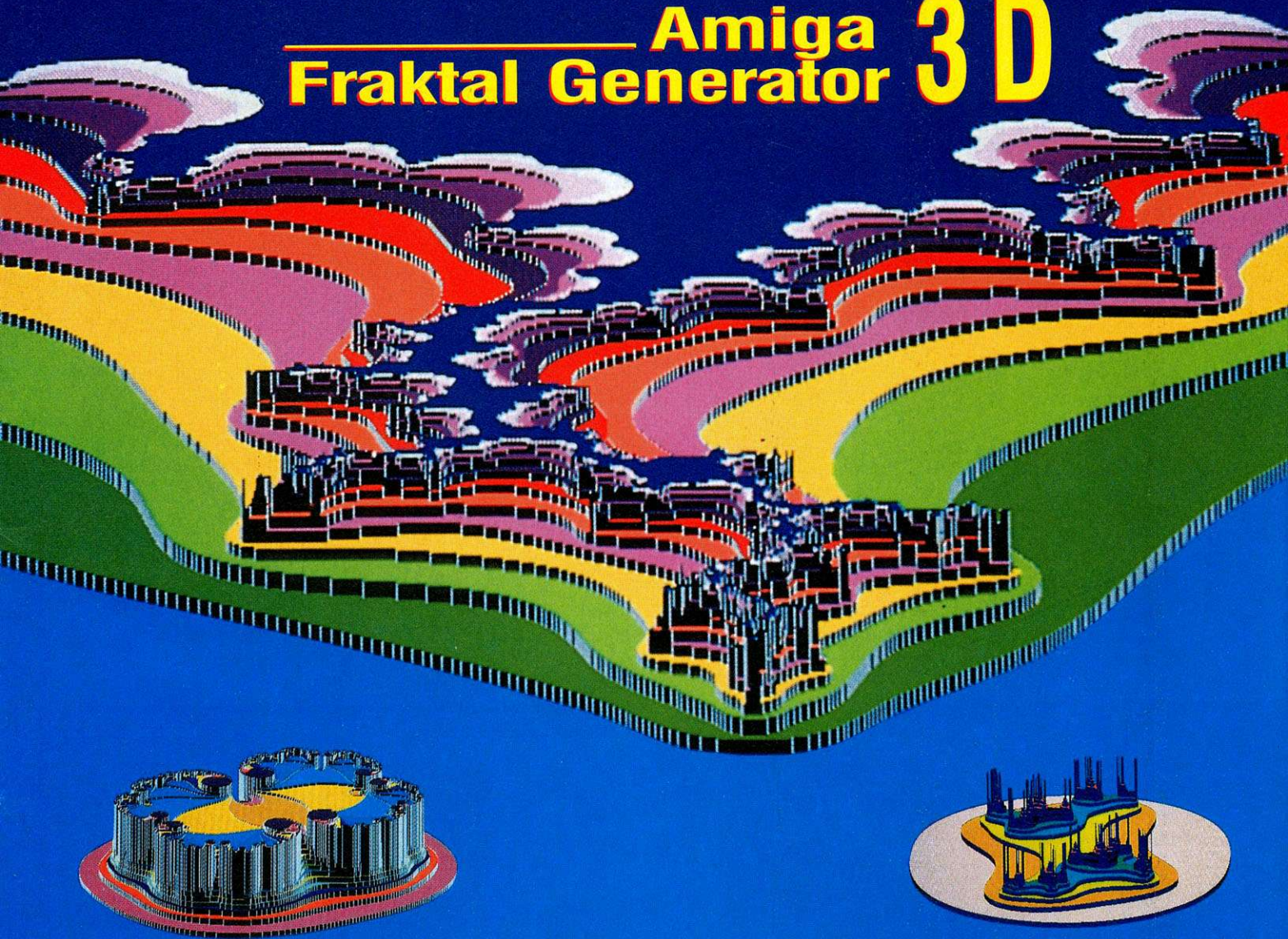
- ausgezeichnetes Parallaxscrolling
- Supergrafik
- Spitzen-sound
- stimmige Spielatmosphäre

Negativ:

- Spielstand nicht speicherbar



Amiga 3D Fraktal Generator



Supergrafik im Sekundentakt

Vergessen Sie alles, was Sie bisher über Fraktalgrafik-Programme gehört haben
– die unendliche Weite phantastischer Bilder erschließt sich nur über ein
superschnelles Programm: **Fraktal Generator 3D**

High-Speed

Nur noch 7 Sekunden für das Urbild!

Super-Parallel-Projektion

Frei wählbarer horizontaler Blickwinkel mit 360 Grad:
Betrachten Sie das "Fraktalobjekt" von allen Seiten
Stufenloser vertikaler Blickwinkel:
Wahlweise Sicht von oben und unten, schräg oder in der Totalen

Speicherung im IFF-Standard

Einladen der Fraktal-Bilder in Mal- und Zeichenprogramme:
Verwendung als Hintergrund, Motiv oder Vorlage

Voller Bedienungskomfort

Auswahl komplett mit Pull-Down-Menüs
Wahlweise Maus- oder Tastensteuerung

Phantastische Farbmöglichkeiten

32 Farben im Low-Resolution-PAL-Modus
Eigenes Farbrequester mit stufenloser Schiebereglung

Mehrere separate Bildspeicher

Bis zu vier Bilder gleichzeitig abrufbar
Separate Farbzuoordnung und Animationsmöglichkeit

Farb-Animationen

Phantastische Effekte durch Amiga-Color-Cycling

Amiga 3D Fraktal Generator

3 1/2"-Disk. Best.-Nr. 2901

69,- DM (unverbindliche Preisempfehlung)

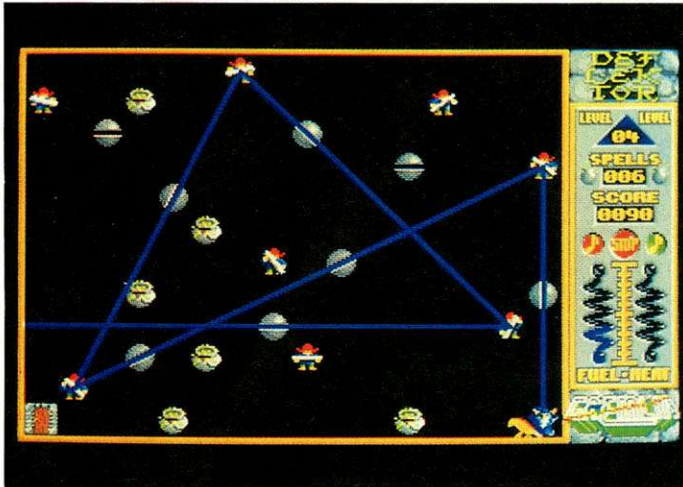
Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:

Inland:		Ausland:	
Einzelpreis	69,- DM	Einzelpreis	69,- DM
zzgl. Versandkosten	4,- DM	zzgl. Versandkosten	6,- DM
Endpreis	73,- DM	Endpreis	75,- DM

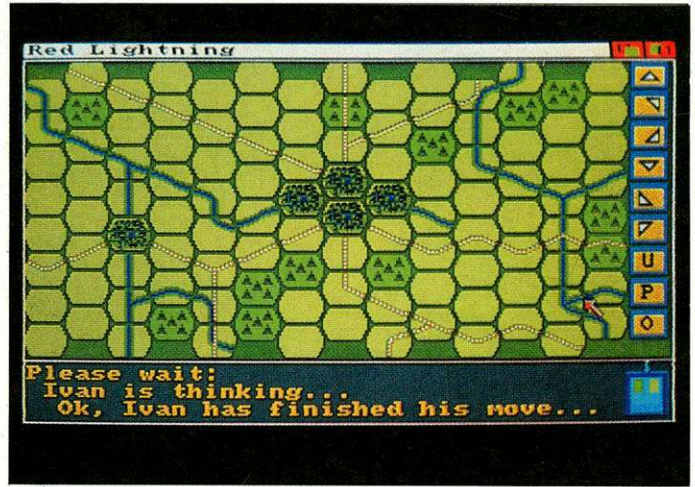
– Bitte benutzen Sie die Bestellkarte im Heft. –

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV
Daten- und
Medienverlag



Zerstören Sie alle Hindernisse mit dem Laserstrahl. Drehspiegel und andere Extras machen's möglich



Welchen Bomber hätten's denn gerne? Red Lightning bricht den dritten Weltkrieg vom Zaun

Mindbender

Man nehme einen Laser, eine Menge Hindernisse, die mit dem Laser zerstört werden müssen, und ein Kontingent drehbarer Spiegel. Schaffen Sie alle Zutaten auf eine Diskette, und Sie erhalten...
– Mindbender.

In der Vergangenheit hat sich eine Weisheit immer wieder bewährt: Die einfachsten Spielideen sind die besten. Die Spielidee hinter Mindbender ist so einfach, daß Sie jedermann nach dem ersten Spiel verstanden hat. Jedes Level besteht aus einem Bildschirm, auf dem ein kleiner Zauberer steht, von dem ein Lichtstrahl (Laser) ausgeht. Dieser Zauberer kann gedreht werden, womit sich auch die Richtung des Laserstrahls verändert. Neben dem Zauberer stehen noch einige Männchen und Hindernisse auf dem Spielfeld. Jedes der Männchen trägt einen Spiegel, der den Strahl reflektiert. Die Männchen können einzeln gedreht werden, auch dadurch verändert sich der Verlauf des Laserstrahls. Jedes Hindernis reagiert auf einen einfallenden Laserstrahl anders. Einige schlucken den Strahl einfach, andere zerplatzen, sobald sie getroffen wurden. Darüber hinaus gibt es spezielle Hindernisse, wie z.B. die Glühbirnen, die eingeschaltet werden, sobald sie der Strahl trifft, und sich ausschalten, sobald der Laser sie zum zweitenmal trifft. Um ein Level zu meistern, muß der Spieler den Strahl nur so lenken, daß alle zerstörbaren Hindernisse

verschwinden und alle Glühbirnen erleuchtet sind. Ist das geschafft, braucht der Strahl nur noch zu der Tür gelenkt werden, die sich in jedem Level befindet, und es ist geschafft. Mindbender macht höllisch viel Spaß und ist, obwohl es schon ein sehr ähnliches Spiel gab (Deflektor), erstaunlich frisch geblieben.

(hs)

AMIGA DOS Blitzlicht

Name: Mindbender
Hersteller: Magic Bytes
Quelle: Fachhandel
Preis: Stand bei Redaktionsschluß noch nicht fest.

Positiv:

- leicht erlernbar
- hohe Spielmotivation
- Leveleditor vorhanden

Negativ:

- zu wenige Levels



Red Lightning

Irgendwann in den 90er Jahren dieses Jahrhunderts wird ein dritter Weltkrieg stattfinden. Dieser globale Konflikt wird im wesentlichen in Westeuropa ausgefochten, denn dort prallen NATO und Warschauer Pakt erstmals aufeinander.

Militärische Plan-
spiele, die von Was-wäre-wenn-Voraussetzungen ausgehen, gibt es zuhauf. So ziemlich jeder historische Konflikt, von Hannibals Alpenüberquerung bis zu Napoleons Eroberungszügen wurde schon einmal als Computer-Strategiespiel umgesetzt. Red Lightning geht einen Schritt weiter und bricht den dritten Weltkrieg vom Zaun. Die Simulation kann alleine oder zu zweit gespielt werden, wobei jeder Spieler die Kontrolle über einen militärischen Machtblock übernimmt. Zug um Zug verteilen die Kommandeure ihre Kampfgebote, wobei jeder Zug einem Zeitraum von 12 Stunden entspricht. Gespielt wird auf einem abstrakten Sechseckraster, das die Landkarte von Mitteleuropa darstellt. Das Ziel des Warschauer-Pakt-Kommandeurs ist die Eroberung des Ruhrgebietes, das des NATO-Oberbefehlshabers, diese Invasion zu verhindern.

Die Einflußmöglichkeiten des Spielers sind so vielfältig wie die vorhandenen Waffensysteme, und von denen gibt es reichlich. Laut Herstellerangaben wurden bei Red Lightning sogar die letzten Neuheiten internationaler Vernichtungstechnologie berücksichtigt.

Red Lightning ist eine technisch routiniert gemachte

Konfliktsimulation. Moralisch unbeschwerte Gemüter können vielleicht auch so etwas wie Spielspaß entdecken, während sie diverse Bomberstaffeln in Richtung Osten auf den Weg schicken. Mir persönlich bereitete allein der Gedanke, einen dritten Weltkrieg spielerisch auszufechten, nicht wenig Unbehagen.

(hs)

AMIGA DOS Blitzlicht

Name: Red Lightning
Hersteller: S.S.I.
Quelle: Fachhandel
Preis: 99,- DM

Positiv:

- gute Dokumentation
- verschiedene Szenarios

Negativ:

- Kriegspiel derbster Machart



AMIGA DOS - Spieletips

Um bei diesem Spiel Erfolg zu haben, ist es ratsam, einige generelle Regeln zu beachten. Wenn Sie Ihr Zimmer im Pub verlassen, müssen Sie immer darauf achten, daß die Zimmertür, die Schranktür und die Haustür geschlossen sind.

Personal Nightmare - Die Komplettlösung

Gegenstände, die mit 'Police' gekennzeichnet sind, müssen spätestens am Nachmittag des vierten Tages dem Polizisten übergeben werden. Noch besser, man gibt eventuell mitgeführte 'Polizei-Gegenstände' gleich dem Polizisten, sobald man ihn sieht.

Nun zur eigentlichen Lösung.

Der erste Tag

Zu Anfang warten Sie, bis Jimmy den Pub verläßt. Verfolgen Sie ihn. Wie Sie sehen werden, hat Jimmy einen Unfall und liegt tot auf der Straße. Jimmy hat einen Schlüssel und einen Film bei sich. Beides sollten Sie an sich nehmen. Außerdem liegt noch ein abgerissenes Nummernschild (Polizei) auf der Straße, das ebenfalls mitgenommen werden sollte. Das nächste Ziel unserer Exkursionen ist das Registry Office. Der Safe, den Sie dort finden, kann mit dem Kommando 'DIAL 1815' geöffnet werden. Dort finden Sie ein Zertifikat, das einem Polizisten übergeben werden sollte.

Der nächste Weg führt uns zum Ivy Cottage. Suchen Sie dort die Küche. Westlich neben der Küche ist der Eingang zu einer Dunkelkammer. Dort finden Sie nach der Eingabe von 'LOOK UNDER TABLE' ein Fotomagazin. Wenn Sie dies in Ihrem Inventory-Bildschirm untersuchen, finden Sie ein Voucher. Das Voucher behalten wir, das Magazin wird wieder abgelegt. Südlich hinter dem Ivy Cottage liegt ein Geräteschuppen, der eine genauere Untersuchung wert ist. Durch das Kommando 'LOOK BEHIND PAINT PODS' fördern wir einen Wetzstein (Oilstone) zu Tage, den Sie mitnehmen. Ferner liegen hier noch ein Vorschlaghammer (Sledgehammer) und ein Secateurs (Skal-

Mit dem Amiga spielen? Nichts leichter als das. Eine Maschine, die Sound und Grafik so gut im Griff hat, wie der Amiga, mit der kann auch gespielt werden. Ein reichhaltiges Spieleangebot existiert auch, also steht einer spannenden Sitzung nichts mehr im Wege. Ein spannendes Abenteuer in einer Fantasiewelt oder lieber ein wildes Actionspiel? Und wenn Sie mit einem Spiel Probleme haben, nicht mehr weiter kommen, wir helfen Ihnen. Auf unseren Helplineseiten finden Sie Lösungswege, Tips und Tricks zu aktuellen und auch älteren Spielen.

pell). Wir nehmen alles mit und begeben uns auf den Weg zur Kirche. Gehen Sie in die Kirche. Achtung, wenn Sie vor der Kirche stehen, befindet sich keine Tür unmittelbar vor Ihnen, und unwillkürlich werden Sie danach suchen. Sie brauchen nur die Ihnen zugewandte Wand der Kirche anzuklicken und gelangen so in den Vorraum der Kirche. Suchen Sie das Büro des Pfarrers auf, und schlagen Sie mit dem Vorschlaghammer auf den Schreibtisch, der daraufhin zerbricht und sein Innenleben offenbart. Den dort liegenden Schlüssel nehmen wir natürlich an uns. Weiter geht es zum Haus des Vikars. In dessen Flur geben Sie dreimal 'HIT WALL WITH HAMMER' ein. Solchermaßen haben wir nun eine Frau freigelegt, die schon lange tot ist und offensichtlich eingemauert wurde. Das Gebetsbuch, das die Dame hält, können Sie nicht ohne weiteres an sich nehmen. Die steifen Finger der Toten halten es fest. Schneiden Sie die Finger mit der Eingabe 'CUT FINGERS WITH SECATEURS' ab. Nun können Sie das Buch nehmen und es lesen. Danach wird es einstweilen nicht mehr gebraucht und im Flur abgelegt. Gehen Sie jetzt in das Jugendzimmer im ersten Stock. Wenn Sie hier 'LOOK IN BOOKS' eingeben und dann das 'Room'-Icon anklicken, finden Sie eine alte Ausgabe der Schatzinsel (Treasure Island), in dem noch ein 'Savings Book' (Sparbuch) steckt. Das Sparbuch nehmen wir an uns, der Rest kann liegen bleiben.

Nun geht es zum Friedhof zurück, wo wir uns zur Gruft begeben. Dort liegt hinter den

Büschen verborgen ein wenig Knoblauch, an den Sie mit den Eingaben 'LOOK IN WEED' und 'GET GARLIC' herankommen. In der Gruft steht ein Sarg, welcher Michael Williams, der eigentlich ein Vampir ist, als Heilstatt dient. Nicht weiter beachten, diesen Untoten machen wir später fertig. Einstweilen treten wir durch die Tür hinter dem Sarg. Im ersten Stock des nun folgenden Traktes wird es gefährlich. Hier ist das Schlafzimmer von Alice, die zum einen die örtliche Lehrerin und zum anderen, schenkt man den Gerüchten Glauben, eine Hexe ist. Unvorsichtige können sich schnell davon überzeugen, das Alice tatsächlich eine Hexe ist. Da wir darauf aber gar keinen Wert legen, ziehen wir vor der Tür die Schuhe aus, so daß uns die Schlafende nicht hört. In ihrem Schlafzimmer sollten Sie Alice mit dem Kommando 'LOOK IN ALICE' genauer untersuchen.

Den Schlüssel nehmen wir mit und schauen noch unter das Bett. Die dort liegende Stehleiter paßt genau unter die Dachluke im Flur. Nachdem Sie die Leiter erklommen haben, müssen Sie feststellen, daß die Dorfbewohner den Bock zum Gärtner gemacht haben. Alice, die Hexe und Lehrerin, hält auf dem Dachboden eine junge Frau gefangen. Während wir die Frau befreien, wacht Alice auf und beginnt laut fluchend die Leiter hinaufzusteigen. Racheakte der Hexe lassen sich mit dem Krug und dem Wasser vereiteln. 'FILL JUG WITH WATER' und 'POUR WATER' sind die Eingaben, die die böse Alice in die ewigen

Jagdgründe schicken. Jetzt können wir mit 'UNLOCK JUDY' die junge Frau aus ihrer mißlichen Lage befreien.

Unser nächster Weg führt uns zum Manor House. Dieser einst stattliche Landsitz ist nach einer Feuersbrunst nicht wieder aufgebaut worden. Vor der Tür des Manor House warten wir, bis es 23:00 geworden ist. Während Sie warten, sollten Sie darauf achten, daß Michael Williams, der Vampir, Ihnen nicht zu nahe kommt. Hinter dem Gittertor finden Sie einen alten Zaun, an dem eine Stange lose ist. Diese Stange müssen wir unbedingt mitnehmen. Denn schon wenig später kommt ein Höllenhund auf uns zu, den Sie nun mit 'HIT DOG WITH STAKE' in die Flucht schlagen können. Nun steht dem Betreten des Hauses nichts mehr im Wege. Den klaffenden Abgrund im Flur überwinden wir, indem wir uns an dem dort befindlichen Seil darüberschwingen, 'SWING OVER HOLE'. Im ersten Stock finden Sie eine ziemlich ramponierte Leiche.

Wir untersuchen Sie, und fördern eine Flasche zu Tage, die wir natürlich an uns nehmen.

Nun wieder zurück zum Pub. Zunächst gehen wir auf unser Zimmer und entledigen uns einiger momentan nicht benötigter Gegenstände, als da wären: Brass Key, Shiny Key, Yale Key, Vicarage Key, Secateurs, Certificate, Maul, Briefcase und Tie. Jetzt gehen wir noch einmal in den Schankraum des Pubs und versorgen uns dort mit der Trompete (Bugle).

Weiter geht es in die Küche, in der wir ohne zu zögern die Trompete in den Abfluß stopfen (PUT PLUG IN SINK), andernfalls erscheint ein Poltergeist, der uns unumwunden ein Messer ins Hirn jagt, worauf wir ja nicht scharf sind. Ist der Abfluß richtig verstopft, können wir ungehindert die Waschmaschine und die darin befindliche Jeans untersuchen. Den Schlüssel, den Sie dabei finden, sollten Sie gut aufbewahren. Nun können Sie auf Ihr Zimmer gehen und den Rest der Nacht schlafen (SLEEP).

Der zweite Tag

Um sechs Uhr früh ist die Nacht vorbei. Unsere erste Exkursion führt uns in die Zimmer unserer Mitbewohner. Im Nebenzimmer, dem Schlafgemach der Jones, finden wir einen Spiegel, den wir zur Beseitigung des garstigen Vampirs noch benötigen. Noch ein Zimmer weiter fördert eine genaue Untersuchung der Bettdecke ein Tagebuch ans Licht. Mit 'UNLOCK DIARY' kann es aufgemacht und gelesen werden. Starker Tobak, der wohl in den Händen der Polizei am besten aufgehoben ist. Suchen Sie nun Mr. Masons Garage auf. Sobald Mr. Mason angekommen ist, dies ist leicht an der offenen Garagentür erkennbar, geben Sie 'LOOK IN BIN' ein. Hier finden Sie ein weiteres Nummernschild, das die örtliche Polizei gewiß brennend interessiert. Gehen Sie zum Postamt weiter. Der nette Beamte hinter dem Schalter geben wir unser 'SAVINGS BOOK' und können nun mit der Eingabe 'BUY STAMP' eine Briefmarke kaufen. 'LOOK IN

DISPLAY STAND' zeigt uns, daß dort ein Umschlag (ENVELOPE) liegt. Den nehmen wir, stecken unser Voucher und den Film hinein, kleben die Marke darauf und werfen das Ganze in den Briefkasten (POSTBOX) vor dem Amt.

Als nächstes widmen wir uns einem alten Bekannten, dem Vampir. Taghell ist es, also müßte der Untote in seinem Sarg in der Gruft liegen, zu der wir nun gehen. Unterwegs sollten Sie noch einmal kontrollieren, ob Sie auch wirklich den Spiegel bei sich haben. In der Gruft gehen wir bis an den Sarg heran, der sich nach einem kleinen Moment öffnet. Heraus steigt, wer hätte es gehaut, Michael Williams, der örtliche Ersatzdrakula. Jetzt heißt es, so schnell wie möglich viermal in Richtung Westen auszuweichen. Dann geben Sie noch 'REFLECT LIGHT WITH MIRROR' ein, und der Vampir war einmal. Nun stellen wir uns auf die Straße und warten, bis die Turmuhr 0:00 Uhr schlägt. Jetzt gehen wir zurück zum Pub und warten

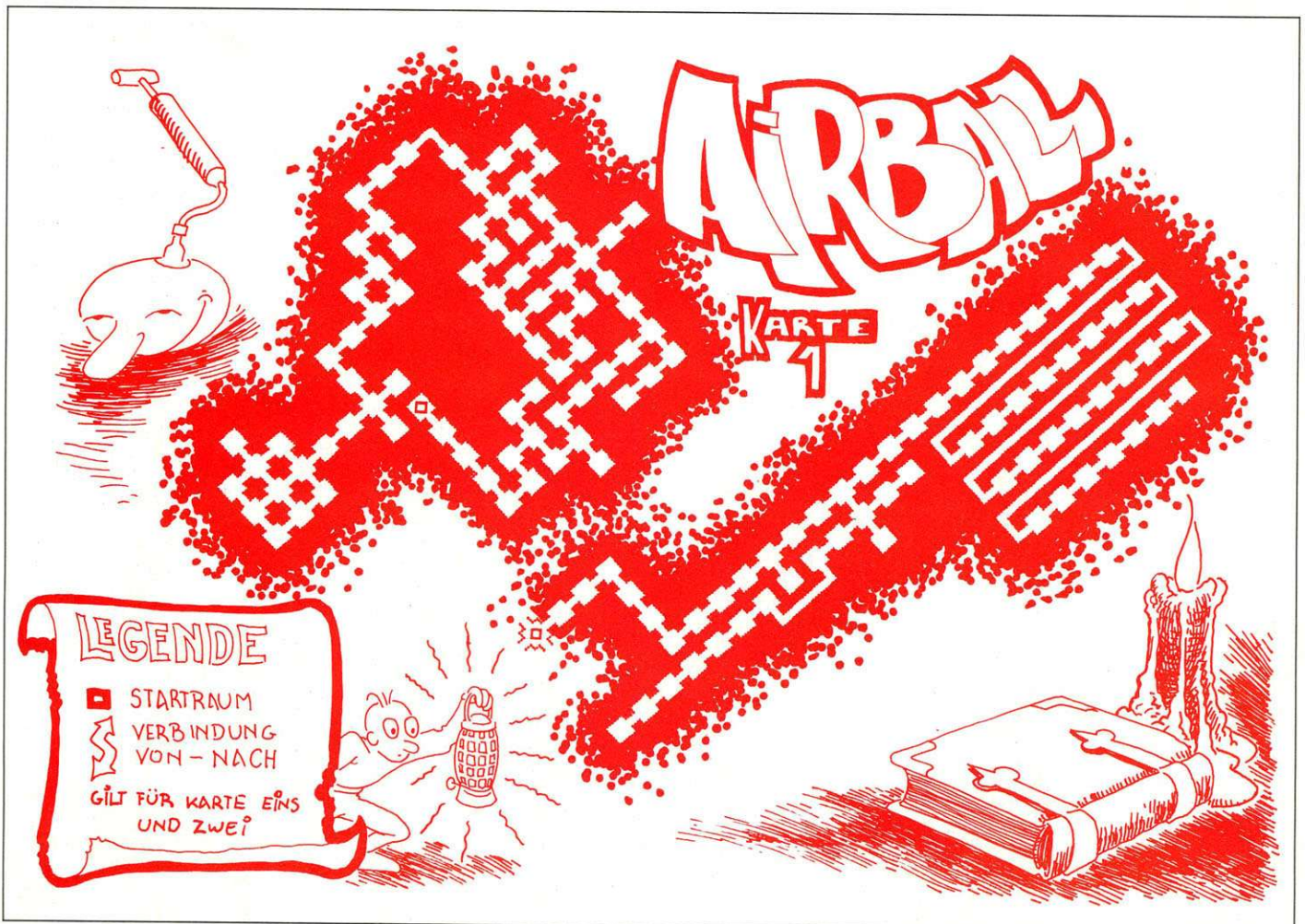
hier. Es sollte nicht lange dauern, bis Mr. Mason nebst seinem Auto erscheint und versucht, Sie zu überfahren. Sobald der Wagen auf uns zukommt, weichen wir in gekonnten Nord-Ost-Nord-Schritten aus, und zwar solange, bis Mason an den Baum knallt. Wir gehen in das Auto hinein und untersuchen das Handschuhfach (GLOVEBOX). Dort finden wir einen Schlüssel, der natürlich mitgenommen wird. Nun können wir wieder zurück in den Pub gehen und uns zur Ruhe begeben. Allerdings sollten Sie, auch wenn dies nicht Ihren Gewohnheiten entspricht, die Trompete mit ins Bett nehmen.

Der dritte Tag

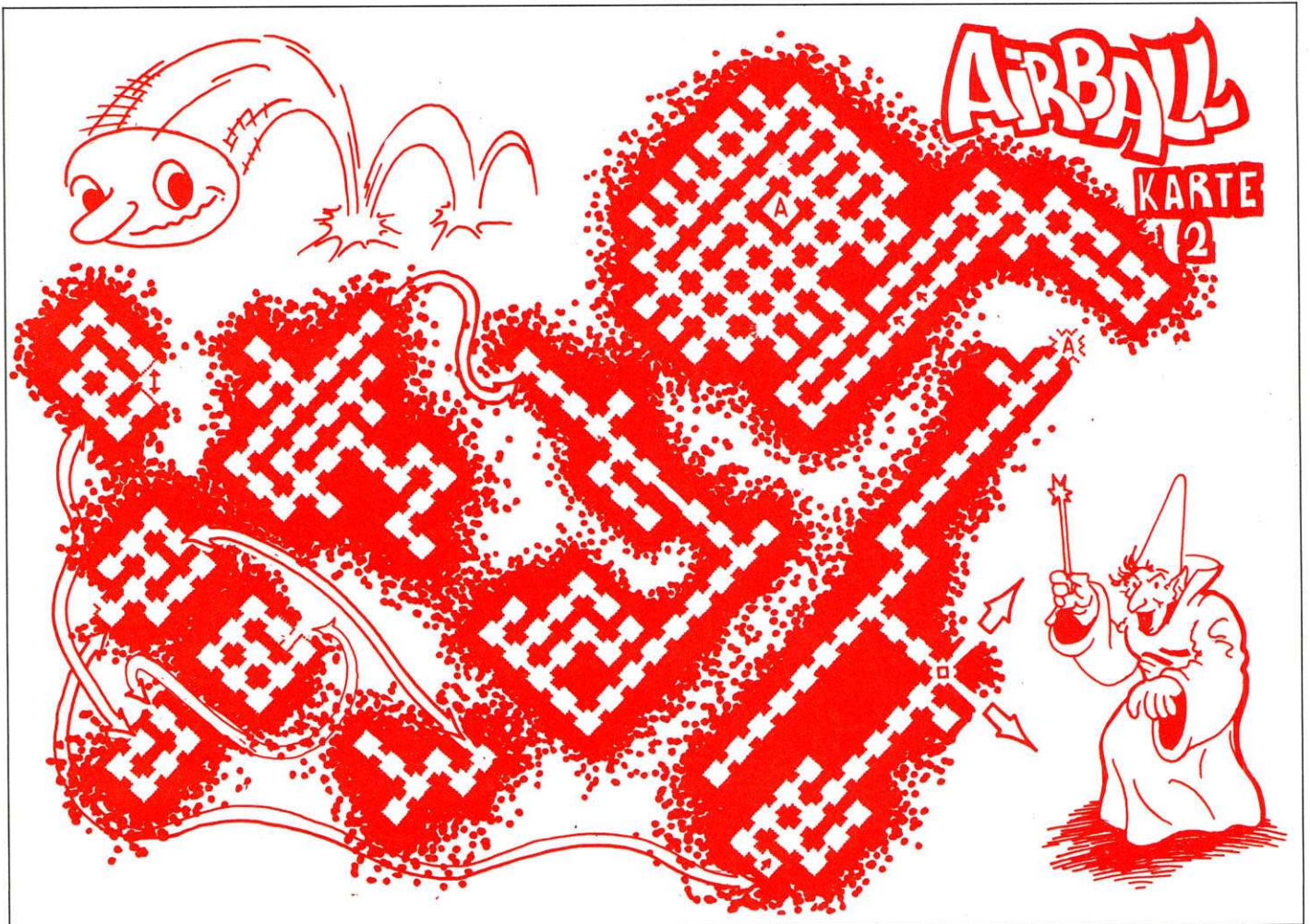
Der dritte Tag beginnt gleich mit einer unangenehmen Überraschung. Die Horde Spielzeugsoldaten aus dem Jugendzimmer versucht, von unheimlichem Leben erfüllt, Sie in Ihrem Bett zu erschießen. Benutzen Sie die Trompete 'SOUND BUGLE', und das Regiment kratzt die Kur-

ve. Vor der Türe finden wir bei genauerer Untersuchung des Raumes eine Spielzeugpistole und Tonys Katze. Die Pistole nehmen wir mit, die 'WOOL' lassen wir fallen. Mit 'SHOOT POPGUN' wird die Spielzeugpistole benutzt, und nun können wir den Korken, der bisher im Lauf steckte, mitnehmen. Den Korken nehmen wir mit, die Pistole wird abgelegt. Mit 'CLEAN BOTTLE' reinigen wir die Flasche und bereiten sie somit auf die geplante Verwendung vor. Begeben wir uns in die Kirche, stellen uns dort vor dem steinernen Engel auf und beten (PRAY). Daraufhin beginnt die Skulptur zu weinen. Die Tränen fangen wir mit der Eingabe 'PUT TEARS IN BOTTLE' auf. 'PUT CORK IN BOTTLE' versiegelt die Flasche mit dem Korken.

Jetzt, da Mr. Mason nicht mehr in seiner Garage ist, können wir diese ungehindert betreten. Gehen wir dort hin und schauen uns ein wenig genauer um. Zunächst unterziehen wir das Büro einer genauen Untersuchung. Die



Karte 1



Karte 2

alten Papiere 'OLD PAPERS' scheinen sehr interessant zu sein. Der 'BILL', den wir dort finden, wird der Polizei übergeben. Wir gehen in die Werkstatt, wo die Eingaben 'LOOK IN BENCH' und 'GET SPANNER' weiterhelfen. Stellen Sie sich nun hinter das Auto und betätigen Sie den Schalter der Hydraulikbühne. Unter dem Auto muß die 'SUMP NUT' gereinigt werden und wird dann mit 'TURN SUMP NUT' geöffnet. Jetzt wird der Ölstein (OILSTONE) mit den Kommandos 'PUT OILSTONE IN OIL' und 'GET OILSTONE' eingeölt.

Unser nächstes Ziel ist das Rose Cottage, in dessen Schlafzimmer wir eine Robe finden. Eine genauere Durchsuchung des Wohnzimmers fördert eine Notiz (NOTE) zu Tage. Auch dieser Zettel ist in den Händen der Polizei am besten aufgehoben. Für heute haben wir alles erledigt und gehen deshalb wieder auf unser Zimmer. Hier warten wir noch, bis es Schlafenszeit (18:00 Uhr) ist und schlafen (SLEEP) uns dann aus.

Der vierte und letzte Tag

Nach dem Aufwachen gehen Sie in die Schankstube des Pubs und warten, bis die Bar geöffnet wird. Mr. Jones händigt uns einen Umschlag aus, der die Fotos enthält, die wir zur Entwicklung geschickt hatten. Wir nehmen die Fotos heraus und übergeben Sie der Polizei. Wir gehen wieder auf unser Zimmer und nehmen folgende Gegenstände mit: Suit, Shoes, Raincoat, Oilstone, Sledgehammer, Rob, Bottle und Maul. 'REMOVE HANDLE FROM MAUL' eintippen und den 'MAUL' wieder in den Schrank legen. Jetzt gehen Sie zum Manor House und warten davor, bis es 23:00 Uhr geworden ist. Westlich des Hauses liegt das Mausoleum. Mit 'HIT MAUSOLEUM WITH HAMMER' verschaffen wir uns Zutritt. Hier besorgen wir uns mit 'GET AXE' eine Waffe, die mit 'PUT HANDLE ON AXE' einsatzbereit gemacht wird. 'SHARP AXE WITH OILSTONE' sorgt dafür, daß die Klinge der Axt scharf wird. Jetzt verkleiden wir uns mit

'WEAR ROBE'. 'MOVE COFFIN' legt den Zugang zu tiefergelegenen Geheimgemächern in dem Mausoleum frei. Wir steigen hinunter, begehen uns vor die Tür und warten. Kurz nach zwölf tauchen aus der Tür zwei Acolytes auf. Gehen Sie durch die Tür und geben Sie gleich dahinter 'HIT ACOLYTE WITH AXE' ein. Sobald die Meldung 'You realize that the Chief acolyte is your father' den eben gegebenen Befehl durch Return abschicken. Jetzt haben wir es nur noch mit Luzifer persönlich zu tun. Hat man jedoch die richtige Waffe bei sich, erweist sich auch dieser als leichter Gegner. Wir geben 'HIT DEVIL WITH BOTTLE' ein, und schon ist der Tynham Cross vom Bösen befreit.

Wo halten sich die Geheimlevel von Hard and Heavy versteckt?

Eine Frage, die sicher mehr als einen begeisterten Amigafreund interessiert. Wir haben

es für Sie herausgefunden. Und zwar sollten Sie die Level **Vier, Sechs, Vierzehn, Achtzehn und Zweiundzwanzig** genauer durchsuchen, denn hier irgendwo sind die Terminals versteckt, durch die man in die Geheimlevel gelangt.

New Zealand Story – Geheimtüren-latein und Monster-meuchelhilfen!

Tikki Kiwi ist ein gerngesehener Gast auf meinen Computermontitor und mehr als einer unserer Leser hat dem armen kleinen Vogel im Kampf gegen das garstige, immerhungrige Walroß beigestanden. Inzwischen haben sich auch einige Tips zu diesem Spiel eingefunden, die wir Ihnen nicht vorenthalten wollen.

Ganz zu Anfang sollte man sich mit den Verhaltensweisen der einzelnen Gegner vertraut machen. Auch die Wirkung der einzelnen Gegenstände, die Ihre Gegner hinterlassen, sollte erkundet

werden. Insbesondere Laser und Bomben zeigen große Wirkung, allerdings ist die Munition der Laserpistole begrenzt.

Während Sie durch die einzelnen Level kurven, sollten Sie darauf achten, wohin Ihre Pfeile fliegen. Mitunter können Sie dann einen erstaunlichen Effekt beobachten: Scheinbar ohne Grund verschwinden Ihre Geschosse. Hier sind Sie nicht etwa über einen Programmierfehler gestolpert, sondern haben eine Geheimtür entdeckt. Wenn Sie diese Geheimtüren benutzen, gelangen Sie ohne Mühen in höhere Level. Die erste Türe findet sich schon in Level 1-1 und führt ohne Umweg in Level 1-4. Dort taucht Tikki Kiwi in einem geheimen Raum auf, der nur durch die Geheimtür erreicht werden kann. Nach diesen Türen sollten Sie intensiv Ausschau halten, da Sie durch deren Benutzung bis zu vier Level überspringen können.

Am Ende der Auckland-Zone begegnen Sie dem ersten Oberbösewicht. Wer hier ver-

sucht, den Riesenwal einfach abzuschießen, wird schnell feststellen, daß er so nicht ans Ziel gelangt. Es gilt, den Schüssen des Wals solange auszuweichen, bis dieser Tikki verschluckt. Jetzt muß Tikki der herabtropfenden Magensäure des Monsterwals ausweichen und dabei schießen, schießen, schießen. Wenig später zerfällt der Wal in seine Einzelteile und Zone eins ist geschafft.

Airball - grenzenlos

Gute Nachricht für alle, die sich im Labyrinth von Airball schon verloren glaubten. Anhand unserer Karte können Sie sich nun orientieren und wichtige Details selbst einzeichnen. Durch den Umstand, daß das Airball-Labyrinth sich über mehrere Stockwerke erstreckt und durch Treppen miteinander verbunden sind, präsentiert sich der Grundriß einigermaßen verwirrend. An den Stellen, an denen sich Räume überlappen, wurden die ent-

sprechenden Teile der Karten mit Pfeilen untereinander verbunden.

Lotterhilfen für streßgeplagte Larryfans

Die wilden und amourösen Abenteuer des Larry Laffer treiben nun auch auf dem Amiga mehr als einen Spielefreund zur Verzweiflung. Interesse an ein paar knackigen Tips für knackige Probleme?

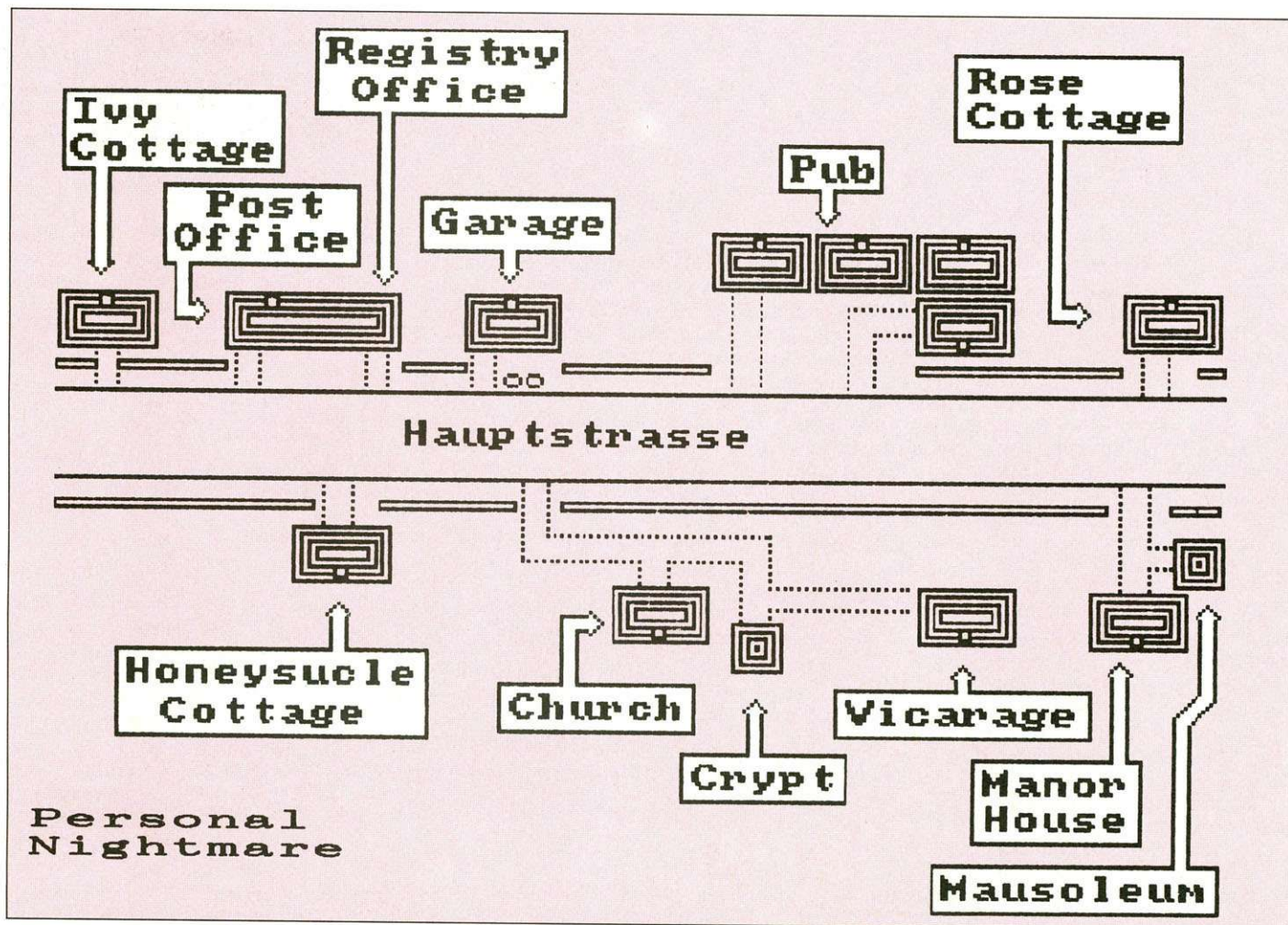
Also, der Betrunkene in Leftys Kneipe möchte einen Drink haben, den Sie ihm an der Theke besorgen können.

In der Toilette der Bar zeigt sich, daß Graffiti mitunter interessante Informationen enthalten können. Mit der Fernbedienung können Sie dem Zuhälter das Programm auf den Bildschirm zaubern, das er gerne sehen möchte. Der Dame im ersten Stock sollte man nicht ungeschützt zu nahe kommen. Der Convenience Store hält für diese Gelegenheit einige farbenfrohe Problemlösungen bereit.

Nachdem Sie bei der Dame im ersten Stock ein paar nette Stunden verlebten, sollten Sie über die Feuerleiter verschwinden. Übrigens, wenn kein Sprungtuch zur Hand ist: Man landet auch in einem gefüllten Müllcontainer weich. Des öfteren macht Larry die Erfahrung, daß manche Menschen Mundgeruch gar nicht mögen und beim geringsten Anzeichen übler Dünste die Quelle des Geruchs ignorieren. Hier schafft das Atemspray, das Larry von Anfang an bei sich hat, Abhilfe.

Wenn einer eine Reise tut, dann kann er was erzählen: Journey

Sie stehen immer noch verzweifelt vor dem Berg des Zauberers, sehen sich Ihre Landkarte an und gelangen doch nicht zum Ziel? Zaubern Sie doch einfach ein wenig mit der Karte herum, dann gibt dieses Stück Pergament seine Geheimnisse preis. (hs)



Karte 3



Im Inneren des Planeten der Helikoptermenschen ist einiges los, verlieren Sie nicht den Überblick

Mr. Heli

Auf dem fernen Planeten der Helikoptermenschen treibt ein verrückter Wissenschaftler seine düsteren Pläne voran. Er und seine Schüler haben tiefe Stollen in das Innere des Planeten getrieben und zerstören von dort aus nach und nach die Welt der Helikoptermenschen.

Die anderen Bewohner dieses Planeten haben die Pläne des bösen Wissenschaftlers schon lange durchschaut, nur hatte bis jetzt noch keiner den Mumm, ihm entgegenzutreten. An dieser Stelle kommt Mr. Heli ins Spiel. Dieser soll nämlich bis zu dem fiesen Gelehrten vordringen und dessen Pläne vereiteln, und Sie dürfen ihm dabei helfen.

Es gilt den kleinen Mr. Heli durch Höhlen und Stollen zu steuern, die durch die Bösewichter angelegt wurden. Diese treiben sich natürlich nach wie vor darin herum und erwarten Mr. Heli schon. Überall in den Höhlen verstreut liegen und hängen Felsformationen, die Mr. Heli mittels seiner Bordkanone vernichten kann. In einigen der Steinblöcke verbergen sich wertvolle Kristalle, für die Mr. Heli Extra-Ausrüstung kaufen kann. Je tiefer unser kleiner Held in den Planeten eindringt, desto rabiate werden die Bösewichter und um so dringender benötigt Mr. Heli die Extras.

Mr. Heli ist ein feines Arcadenspiel mit viel Action. Die einzelnen Höhlen werden mal horizontal, mal vertikal gescrollt. Zwar reicht die Amiga-Version nicht an das Automatenoriginal heran, bereitet aber viel Spielspaß. Die Grafik ist bunt und nett anzusehen.

muß aber alles in allem als recht konventionell bezeichnet werden. Letztendlich bleibt ein passables Actionspiel, das ob seines Spielkonzeptes für viel Spielspaß gut ist, wenn auch die technische Umsetzung bestenfalls als durchschnittlich bezeichnet werden kann.

(hs)

AMIGA DOS Blitzlicht

Name: Mr.Heli
Hersteller: Firebird
Quelle: Fachhandel
Preis: 89,- DM

Positiv:

- Extrawaffen
- passable Grafik
- akkurater Sound

Negativ:

- zu teuer
- programmtechnisch mittelmäßig



BRANDHEISSE KNÜLLERPREISE

Commodore		Atari Computer Mega ST 1 mit Maus + Monochrommonitor SM 124		1549,-	
Commodore Farbmonitor 1084	599,-	Mega ST 1 + SM 124 + Megaflo 30	2399,-		
Series-Farbmonitor Highscreen KP 1448	899,-	Atari Mega ST 1 + Monochrommonitor SM 124	2199,-		
Commodore AMIGA 2000	1829	Atari Mega ST 2 + SM 124 + Megaflo 30	3049,-		
AMIGA 2000 + Farbmonitor 1084	2369,-	Atari Mega ST 4 + Monochrommonitor SM 124 + Festpl. Megaflo 30	4049,-		
3,5"-Laufwerk intern für Amiga 2000	209,-				
PCAT-Karte mit 5,25"-Laufwerk	799,-				
AT-Karte mit 5,25"-Laufwerk	159,-	Epsondrucker (dt. Handbücher)			
20-MB-Festplatte für Amiga 2000 mit SCSI-Controller	949,-	Anschlußfertig an Amiga, Schneider PC, Atari ST und Mega, sonstige IBM-Compatibles			
SCSI Controller Commodore A 2090 A	649,-	AX 400		419,-	
20-MB-Filecard (Seagate, 40 ms Zugriffsf.) für A 2000 mit PC-Karte od. A 1000/Scidac	799,-	LC 400 (24 Nadeldrucker)		689,-	
40-MB-Filecard (Seagate)	699,-	LC 550 (24 Nadeldrucker)		889,-	
40-MB-Filecard (Western Digital, 29 ms)	1069,-	LC 850 (24 Nadeldrucker)		1419,-	
50-MB-Filecard (Seagate, 40 ms)	999,-	Tintenstrahl-Drucker			
2 MB-RAM-Erweiterungskarte für A 2000 aufrüstbar bis 8 MB (Commodore A 2058)	1249,-	(8 Düsen, N.L.Q. max. 240 Zeichen/Sekunde)		619,-	
Externes 3,5"-Laufwerk abschaltbar	2249,-	Stardrucker (dt. Handbücher)			
Externes 3,5"-Laufwerk (Commodore 1010)	119,-	LC-10 mit Centronicsinterface		429,-	
Externes 5,25"-Laufwerk für alle AMIGA	949,-	LC-10 Color Farbdrucker mit Centronics		649,-	
Externe A 500-Festplatte 20 MB Commodore	299,-	LC-24-10 mit Centronicsinterface			
		NEC-Drucker (dt. Handbücher)			
Atari		NEC P 7 Plus 1743,-	3	P 3000	1579,-
Poppay-Disk 1050 (2. Wahl mit Garantie)	299,-	NEC P 6 Plus 1299,-	3	Farption	249,-
Festplatte Atari Megaflo 30	949,-	Einzelteilnetz für NEC P 6 Plus			44,-
Festplatte Atari Megaflo 60	1199,-				
1040 STFM + Monochrommonitor SM 124	1699,-	NEU: Speichereinheit für A 1000 (Amiga 1050)			99,-
1040 STFM + SM 124 + Megaflo 30	2099,-	Druckerkabel 5 m lang für Amiga, ST			49,-
		Atac Multiscan Monitor 0330 mit			949,-
		24-Nadeldrucker LC (35 mm Druckerkabel			499,-
Versandkostenpauschale (Warenwert bis DM 1000,-) drüber: Vorauskasse (DM 5,- ZD).					
Nachnahme (DM 12,50 ZD), Ausland (DM 20,-) Lieferung nur gegen Vorauskasse, Ausland nur Vorauskasse.					
Preisliste (Computertyp angeben) gegen Zusendung eines Freiumschlags. Preise gültig ab 8.10.89.					

CSV RIEGERT Schloßhofstr. 5, 7324 Rechberghausen,
Tel. 0 71 61/5 28 89, Fax 0 71 61/1 35 87

Ist Ihr Programm der HIT ?

Der DMV Verlag sucht ständig
nach neuer, interessanter
Software zur Aufnahme in
unser Softwaresortiment.

Dabei ist es einerlei, ob Sie nun ein Anwendungs- oder ein Spielprogramm geschrieben haben.

Der DMV Verlag bietet Ihnen sein
Software-Know-How an!

Nürnberg Public Domain Studio

Humboldtstr.141 , 8500 Nürnberg 40
Tel.:0911/45 77 54 Fax :0911/446 72 79

Preiswerte Restposten für Ihren Amiga

Ball Raider	39.90	Kwasimodo	25.90
Deep Space	69.90	Roger Rabbit.....	55.90
Drum Studio	25.90	Speed Ball.....	69.90
Die Fugger	49.90	Starways	39.90
Hollywood Poker	25.90	Thexder	49.90
The Jungle Book dt.....	59.90	Thunderblade.....	55.90

Ca.7000 Public Domain + Shareware

Disketten für Atari ST , Amiga und PC

Gratis Katalog-Disk und Info

für Ihren Computer anfordern !!!

Für Amiga: ACS, AMICUS, AMSEL, AMUSE, AUGE, AUSTRIA, BARRACUDA, BAVARIAN, CACTUS, CHIRON, ES-SOFT, FAUG, FISH, KICKSTART, KISS, MSS, NICKLAS, PANORAMA, PFALZ, POSEIDON, R. WOLF, RHS, RLO, RPD, RUHR-PD, TAIFUN, TAIL, TAURUS, TBAG, TIGER CRACK, TORNADO, UKAUG u.s.w.

Disketten

3,5" 2DD 10 Stück DM 17,90

5,25" 2DD 10 Stück DM 6,90

Versandkosten:

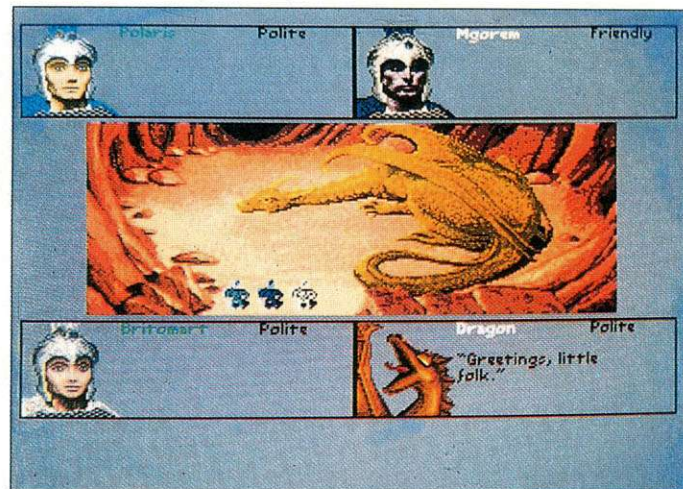
Nachnahme: DM 6.- Vorkasse: DM 3.-

Die abenteuerliche Reise beginnt im Schloß der schönen Königin Gloriana. Sie teilt Ihnen mit, daß Sie die gefährvolle Reise nicht alleine antreten müssen. Zusammen mit Ihnen machen sich noch zwei tapfere Kämpfer auf den Weg. Da niemand weiß, wo der Schattenlord seine Residenz errichtet hat, gilt es für die Helden erst einmal, das sie umgebende Land zu erkunden. Schon bald gelangen die Krieger an ein blau glühendes Tor. Dies ist der Eingang, der zu der Regenbogenbrücke führt. Diese Brücke verbindet alle Regenbogenwelten miteinander. In einer großen Höhle unmittelbar hinter dem Eingangstor wartet ein Wächter. Ein riesiger Drache stellt den Abenteurern ein Rätsel. Erst nach der richtigen Antwort darf die Truppe passieren. Nach der Reise über die irisierende Regenbogenbrücke liegt eine neue Welt vor Ihnen. Werden die Helden hier die Antworten auf ihre Fragen erhalten?

Ein Rollenspiel ganz neuer Machart – Hält Swords of Twilight was es verspricht?

Nach einem kurzen Erkundungsgang erreichen die Abenteurer eine Burg. Dort begegnen sie allerlei Anwohnern und Passanten, die einem Gespräch nicht abgeneigt sind. In diesen Unterhaltungen gelingt es der Truppe, durch geschicktes Fragen einiges an Informationen und Gerüchten über die Aktivitäten des Schattenlords und seiner Verbündeten in Erfahrung zu bringen. Je länger die Helden sich unterhalten, desto mehr kristallisiert sich ein Weg für das weitere Vorgehen aus den gesammelten Informationen heraus.

Swords of Twilight ist ein Rollenspiel, dem man auf den ersten Blick anmerkt, daß sich die Macher vorgenommen hatten, einiges anders zu machen, als dies bisher der Fall war. Interaktion mit Spielcharakteren (also den Personen, die man während des Spiels treffen und befragen kann) wird bei Swords of Twilight großgeschrieben. Die Helden können sich mit fast jeder auftauchenden Person unterhal-



Drachen bewachen den Weg zur Regenbogenbrücke, die die Welten miteinander verbindet

Swords of Twilight

Der Schattenlord hat nach Äonen der Ruhe sein dunkles Haupt wieder über die Regenbogenwelten erhoben. Gloriana, die Königin der Welt Albion, hat Sie erwählt, der Held zu sein, der dem Schattenlord und dessen Verbündeten die Stirn bieten soll.

ten. Die umherziehende Heldentruppe besteht aus drei Mitgliedern, die wahlweise von menschlichen Spielern oder vom Computer gesteuert werden. So ist es möglich, alleine mit zwei Computerhelden loszuziehen oder aber mit bis zu drei menschlichen Mitspielern die Regenbogenwelten zu erkunden.

Unfähige Computerkrieger und der Fluch ausgeprägter Interaktion

So gut diese Idee im ersten Moment erscheint, im Spiel mit zwei Computercharakteren wird der dritte, menschliche Spieler mehr als einmal fluchen. Die Computercharaktere entwickeln Eigeninitiative, die sich nicht unbedingt mit den Zielen des menschlichen Spielers deckt. Die Burschen rennen in der Regel immer dort herum, wo es gar nichts Interessantes zu entdecken gibt. An Wegkreuzungen entscheiden sich die Kerle natürlich grundsätzlich für einen anderen Weg als Sie selbst.

Glücklicherweise hat man die Möglichkeit allzu ausführliche Gespräche der vom Computer gesteuerten Partymit-

glieder zu unterbrechen. Dazu braucht man sich von der Gruppe mit der gerade gesprochen wird nur zu verabschieden. Aber trotz dieser Option geraten die Gespräche mit anderen Personen aus der Spielwelt meist ziemlich langweilig und spielwichtige Informationen tauchen nur sporadisch auf.

Da die Party immer drei Mitglieder hat, es ist gar nicht erst möglich eine Partie mit weniger Teilnehmern zu beginnen, muß der Spieler sich mit dem Eigensinn der Computercharaktere abfinden. Aber nicht nur Negatives gibt es über die beiden anderen Helden zu berichten. Immer wieder gelangt die Truppe an Türen, die nur durch Einsatz vereinter Kräfte geöffnet werden können. Obwohl die interaktiven Aspekte des Spiels in gewissen Situationen eher nervtötend wirken, tragen sie an anderer Stelle wesentlich zum Spielwitz bei. Wenn Sie z.B. mordend und brandschatzend durch eine Stadt ziehen, brauchen Sie sich nicht zu wundern, daß die Bewohner Sie beim nächsten Besuch nicht freundlich empfangen. Die grafische Gestaltung von Swords of Twilight macht auf den ersten Blick einen guten Eindruck. Das häufige Wie-

derholen nur wenig veränderter Bilder und Darstellungen zeitigt den Effekt, daß die Grafik nach längerem Spiel eher langweilig wirkt. In puncto Musik wird der Spieler auch nicht gerade verwöhnt. Das, was da aus den Lautsprechern tönt, geht dem Zuhörer schon nach wenigen Minuten auf den Geist.

Viel Drumrum und wenig drin

Anne Westfall und John Freeman hatten sich mit diesem Spiel einiges vorgenommen. Dem Anspruch eine Reihe von interaktiven Welten zu schaffen, die mit Spielpersonen bevölkert sind, die einen eigenen Willen (oder scheinbar einen eigenen Willen) haben, sind die beiden ja noch halbwegs gerecht geworden. Was die Spielbarkeit betrifft, haben sie das Klassenziel weit verfehlt. Die anfängliche Lernphase zieht sich wie Kaugummi. Alles in allem ist Swords of Twilight eigentlich nur eingefleischten Rollenspielern zu empfehlen. Wer sich einmal mit dem Spielablauf vertraut gemacht hat und gelernt hat, über bestehende Mängel hinwegzusehen, kann sicherlich auch mit Sword of Twilight manche abenteuerliche Spielsitzung haben.

(hs)

AMIGA DOS Blitzlicht

Name: Swords of Twilight
Hersteller: Electronic Arts
Quelle: Fachhandel
Preis: 89,- DM

Positiv:

- sehr komplex
- lange Spieldauer
- viele Rätsel und Aufgaben

Negativ:

- magere Grafik
- magerer Sound
- Steuerung sehr gewöhnungsbedürftig





Steuern Sie den Gnom Ulopa durch die einzelnen Ebenen der Twinworld. Es gilt, ein zerschmettertes Amulett wieder zusammenzusetzen

Twinworld

Ulopa, so lautet der Name des Helden dieses neuen Plattformspiels. Dieser Gnom hat die ehrenvolle und gefährliche Aufgabe, ein magisches Juwel, das durch einen bösen Druiden in zwanzig Teile zerschmettert wurde, zusammenzusetzen.

Nur müssen diese zwanzig Teile erst einmal gefunden werden. In bester "Jump & Run"-Tradition steuert der Spieler den kleinen Ulopa über Plattformen, Steinbrüstungen und Gerüste. Dabei begegnet er immer wieder greulichen Ungeheuern. Jedoch steht der Gnom nicht wehrlos da, er kann kleine, magische Bällchen schleudern, die für Ungeheuer äußerst ungesund sind. Viele der Orte, an denen die Splitter des Amuletts liegen, kann Ulopa nur durch Tunnel erreichen. Die Türen zu diesen unterirdischen Spielabschnitten befinden sich überall in Ulopas Welt. Einmal kann er ohne Probleme betreten, für andere benötigt er einen Schlüssel.

Twinworld ist eines der wenigen wirklich interessanten "Jump & Run"-Spiele, die es für den Amiga gibt. Die Aufgabe, Ulopa zu steuern, beginnt recht harmlos, wird aber von Zone zu Zone schwieriger. Versteckte Extras, Geheimtüren und andere Überraschungen sorgen dafür, daß auch nach längerem Spiel noch Neues entdeckt werden kann.

Die Grafik des Spiels ist zweckmäßig und nett anzuschauen. Jedesmal wenn Ulopa eine Zone geschafft hat, wechselt die Grafik und die

Hintergrundmelodie. Alles in allem ist Twinworld ein feines "Jump & Run"-Spiel. Außerdem muß angemerkt werden, daß unsere Testversion noch nicht dem 100prozentig fertigen Spiel entspricht, demnach kann Twinworld nur noch besser werden.

(hs)

AMIGA DOS Blitzlicht

Name: Twinworld
Hersteller: Blue Byte
Quelle: Fachhandel
Preis: ca. 70,- DM

Positiv:

- spannender Spielverlauf
- versteckte Extras
- passende Melodien

Negativ:

- mäßige Animationen

URTEIL

GRAFIK
6

SOUND
6

MOTIVATION
7

Gold Vision präsentiert:

Amiga Desktop Publishing PageStream

Nur 398.- Version 1.8 mit deutschem Text-Import

PageStream Font-Disks Nr. 1-13 je DM 89.-

GoldCOMMANDER
Die CLI-Erweiterung mit Mausunterstützung!
Die Intuition-Anbindung für das CLI!
- Texte (z.B. Programmieren aus einem Directory) können durch Anklicken in die Kommando-Zeile übernommen werden. Programme können Sie nun durch einmaliges Anklicken starten. Aufwendiges Abtippen entfällt!
- Multiselekt-Funktion zur Anwendung von komplexen Kommandos auf Gruppen von Dateien (z.B. "rename *.S *.C" oder "backup").
- Erstellen Sie sich bis zu 16 Menüs mit den wichtigsten Befehlen oder Kommandofolgen (z.B. Compiler-Aufruf).
- Installieren Sie Gadgets für weitere Befehle.
- Änderung der Fenstergröße durch Tastendruck.
- Positionierung des Cursors mit der Maus.
- Belegung der Funktionsstasten.
- Deutsches Handbuch. Nur DM 49.80

Bestellungen bitte schriftlich oder telefonisch an Gold Vision, Kurlurstandamm 64-65, 1000 Berlin 15, Tel. 030/88 33 505. Lieferung gegen Vorkasse (Bar, Scheck zzgl. DM 3.- Versandkosten) oder per Nachnahme (zzgl. DM 6.- Versandkosten). Infos und Katalog gegen frankierten Rückumschlag!

Version 1.1
DM 298.-

PixelScript

Der PostScript-Interpreter für den Amiga

PixelScript ermöglicht es Ihnen PostScript-Dateien auf Ihrem grafikfähigen Preferences-Drucker in der höchsten Auflösung auszugeben. Helvetica- und Times-Kompatible Fonts werden mitgeliefert. Intuition-Benutzeroberfläche, ausführliches deutsches Handbuch!

HIGH RESOLUTION WORKBENCH
Jetzt in der Version 1.2 !!! Nur DM 39.80
28 % größerer Workbench Screen (bis zu 736 x 568 Punkte) als Softwarelösung.
28 % höhere Auflösung für die Workbench und viele Anwenderprogramme wie Professional Page, PageStream, Deluxe Photolab, Becketext oder WordPerfect.
Die Daten:
- Bis zu 736 x 568 Punkte Workbench, bis zu 92 Zeichen/Zelle, einfache Installation und Bedienung durch grafische Benutzeroberfläche, resettet, bis zu 16 Farben, beliebiges Aus- und Einschalten, manuelle Positionierung (z.B. für Flicker-Fixer) möglich, volle Mausunterstützung. Deutsches Handbuch und Software in stabiler Kunststoffkassette.

Teletone

Telefax

von Panasonic, Mitsubishi, Schneider u.a.
48 Mon.
mtl. ab **49,-**

Anrufbeantworter

50 verschiedene Modelle von Panasonic, Tiptel, Uher, Zettler u.a.
ab **198,-**

Cityruf, Eurosignal, Telefonanlagen, sämtl. Zubehör. Katalog anfordern!

Nordring 98 • 4630 Bochum
Tel. 0234/1 39 50 • Fax 1 49 31

Lüding und Co.
Hohenzollernstr. 64 • 4350 Recklinghns.
Tel. 02361/ 5 95 94 • Fax 5 95 93

WIR HABEN COMPUTER, ZUBEHÖR, LITERATUR... ZUM SPIELEN, LERNEN, ARBEITEN...

BESUCHEN SIE UNS ODER FÖRDERN SIE KOSTENLOSE PREISLISTE AN!

B&C EDV-Systeme GmbH
Tel.: 0222/5054978 Fax: 0222/5054978
Mo-Fr 9⁰⁰-12⁰⁰ u. 13⁰⁰-18⁰⁰ Sa 9⁰⁰-12⁰⁰

SECOND HAND COMPUTER

Ankauf • Verkauf • Vermittlung • Inzahlung

Gebraucht-Computer:

- * Zubehör
- * Neugeräte
- * alle Marken
- * Konkurrenz-Ankauf
- * Ankauf defekter Geräte

Angebote: solange Vorrat

386'er z. B.: DELL, 30 MHz, 8 MB-RAM, 300 MB/15 Ms HD, VGA, VGA-Farb-Monitor, Coproz., ESDI-Controller, etc. 1990,-
IBM z.B.: PS/2 8530, 2 Laufwerke, 640 KB, neuwertig 1298,-
AT-LAPTOPS z.B.:
REIN 420 SLC, 12 Mhz, 640 KB, 40 MB/23Ms HD 5885,-
TOSHIBA LT 3100 4485,-
SCHNEIDER TARGET, 20 MB Harddisk, etc. 3785,-
MODEM 2400 Baud, Hayes-kompat., neu, (*) 399,-
EGA + VGA Karten, neuwertig, ab 248,-

WO SONST ?

Wir kaufen und verkaufen:

- * Homecomputer
- * XTs und ATs
- * Büroanlagen
- * Anrufbeantworter

NEU: Jetzt auch in Kassett!

COMMODORE 1541-I, neu 279,-
MANNESSMANN MT 81 Drucker, 130 Zls, NLQ, neu 389,-
FESTPLATTEN, z.B.: 170 MB Micropolis SCSI, neu 1498,-
ANRUFBEANTWORTER m. Fernabfrage + Ferntextländen, etc. neu (*) 179,-
ANRUFBEANTWORTER m. Fernabfrage, postzugelassen 389,-

Bei mit (*) gekennzeichneten Geräten ist der Betrieb in der BRD lt. Par. 15 FAG bei Strafe verboten.

Weitere gebrauchte und neue Drucker, Monitore, XTs, ATs und 386er sowie Laptops, Bücher, Software, RAM-Erweiterungen und vieles mehr auf Tel. Anfrage !!!

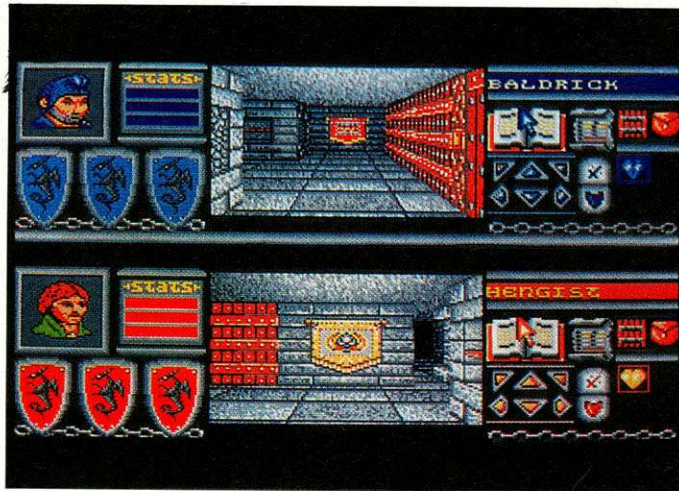
ALPHA 2000 GmbH 24 Std. Info: 069-443000, 6 Frankfurt/M. 1, Ingolstädter Str. 27
ALPHA 2001 GmbH, 24 Std. Info: 05 61-52 50 66, 3501 Niestetal (bei Kassel), Witzenhäuser Str. 10

Hört sich wie ein Auszug aus einem spannenden Fantasy-Roman an, ist es aber nicht. Tatsächlich entstammt diese Szene einem neuen Rollenspiel, in dem die Ereignisse, von fehlenden Dialogen einmal abgesehen, wie in einem Film ablaufen. Das Abenteuer, das in diesem Spiel auf Sie wartet, findet in einem unheimlichen, lange vergessenen, unterirdischen Labyrinth statt. In diesen Gewölben gibt es viele Schätze zu finden, jedoch werden diese durch viele umherstreifende Monster bewacht.

Aber keine Bange, die Monster werden nicht gleich auf Sie losgelassen, zunächst müssen Sie sich eine Gruppe von Helden zusammenstellen, die Ihnen bei Ihrem gefährlichen Abenteuer zur Seite stehen, denn alleine haben Sie nur wenig Überlebenschancen.

Glücklicherweise haben sich in den Eingangsgewölben des Dungeons einige Helden versammelt, die wie Sie auf der Suche nach einer Abenteurergruppe sind, der sie sich anschließen können. Zwar sind einige der Kämpfer nicht gleich bereit, bei Ihnen mitzumachen. Doch wenn Sie sich mit den Helden unterhalten, weicht die anfängliche Reserviertheit, und die meisten schließen sich Ihrer Party an. Sind drei Kämpfer gefunden, ist die Truppe komplett, und das Abenteuer kann beginnen. Hinunter geht es, in die schummerigen Kavernen des alten Verlieses, und schon nach wenigen Metern stößt die Truppe auf einen an der Wand hängenden Gobelin, auf den eine Warnung eingestickt wurde: Hinter der nächsten Tür befinden die monsterverseuchten Teile des Verlieses. Ab hier gilt es, Vorsicht walten zu lassen, wie sich so gleich nach dem Öffnen der Tür zeigt. Denn hier erwartet uns der erste Gegner, der sich nicht mit einem netten Gespräch zufriedengibt. Er will Ihrer Party ans Leder, und dies gilt es zu verhindern. Dieser erste Kampf verläuft noch glimpflich, aber schon der nächste Gegner könnte viel stärker und erfahrener sein.

Eine weitere Tür versperrt den Weg und ist mit einem normalen Schlüssel nicht zu öffnen. Eine Durchsuchung der Räume fördert einen bronzenen Schlüssel zu Tage. Dieser



Die Dungeons von Bloodwych können auch zu zweit erkundet werden. Splitscreen macht's möglich

Bloodwych

Vorsichtig spähte Blodwyn um die Ecke des feuchten, dunklen Korridors, während Zothien, unser Zauberer, den Gang hinter uns absicherte. Die letzten Meter hatten wir uns schleichend vorangetastet. Mit gutem Grund, denn im Gang hinter der Ecke liefen einige Skelette mit trägen, tappenden Schritten Wache. Noch hatten sie uns nicht entdeckt, und so hatten wir noch einen Moment Zeit, uns eine Kampftaktik für die bevorstehende Auseinandersetzung zurechtzulegen.

Schlüssel paßt. Die Tür öffnet sich. Vor uns erstreckt sich ein weiter Raum, dessen Ende in der hier herrschenden Dunkelheit nicht zu sehen ist, und zu allem Überfluß wankt uns schon das nächste Monster entgegen. Ein erneuter Kampf scheint unumgänglich...

Ein Blick auf den Monitor versetzt Sie in die Mitte des abenteuerlichen Geschehens

Was bis hierher in erzählerischer Form geschildert wurde, wird genau so abenteuerlich vor dem Computer erlebt. Das ganze Spiel wurde so gestaltet, daß Sie die Aktionen, die auf dem Bildschirm ablaufen, so sehen, als würden Sie inmitten des Geschehens stehen. Auf dem Bildschirm präsentiert sich Bloodwych in wohldurchdachter Konzeption. Die einzelnen Partymitglieder werden durch schildförmige Icons dargestellt, die mit Porträts der Personen geschmückt sind. Das Verlies wird als Walk-Thru-Szenario dargestellt. Sie sehen den

Dungeon also genau so, als würden Sie ihn selbst durchwandern.

Im ersten Moment erscheint die Bildschirmdarstellung, recht gedrungen, wenn Sie alleine spielen. Wieso, zeigt sich beim Spiel mit zwei menschlichen Teilnehmern. In diesem Falle wird der Bildschirm zweigeteilt, jeder Spieler stellt sich seine eigene Truppe zusammen, und beide Partys können unabhängig voneinander die Labyrinth erkunden. Da jede Party für sich ein eigenes Window hat, in dem das Dungeonszenario dargestellt wird, können die Partys einander sehen, wenn sie sich gegenüber stehen oder die eine Truppe in das Blickfeld der anderen gelangt. Eine gemeinschaftliche Erkundung der Gewölbe ist absolut möglich.

Die Helden einer Party sind zu Beginn des Spieles eher Halbstarke, aus denen erst noch Helden werden müssen. Dies geschieht über Erfahrung. Jeder gewonnene Kampf bringt zusätzliche Erfahrungen mit sich, die die Kampfkünste der Charaktere verbessern. Dies schlägt sich in Erfahrungspunkten nieder, und jedes-

mal, wenn genügend Punkte gesammelt wurden, steigt der Charakter um ein Level auf. Entsprechend verhält es sich mit den Monstern die den Spielern begegnen.

Ich habe Bloodwych nachts getestet. Mein dunkles Arbeitszimmer wurde nur durch eine trübe Schreibtischlampe und dem Monitor erhellt. Was wie ein abendliches Entspannungsspiel begann, geriet nach kurzer Zeit zum handfesten Abenteuer, und mein Monitor wurde zu einem Fenster in eine unheimliche, fremde Wunderwelt. Bloodwych verzichtet auf Musik während des Spiels und gibt nur dann Soundeffekte aus, wenn auch wirklich etwas passiert. So geschah es, das meine Helden in eine Falle tappten, die mit Getöse ausgelöst wurde. Der Krach, den die Falle verursachte, ließ mich vor meinem Computer regelrecht zusammenfahren, so erschreckt war ich. Schon bald wurden aus Charakteren, die eigentlich nichts anderes als abstrakte Datenstrukturen sind, in meiner Phantasie richtige Personen einer höchst interessanten Spielwelt.

Bloodwych ist eine der interessantesten Neuerscheinungen der letzten Monate. Ein Spiel, das schon alleine unglaublich viel Spaß macht und zu zweit zur echten Herausforderung wird.

(hs)

AMIGA DOS Blitzlicht

Name: Bloodwych
Hersteller: Image Works
Quelle: Fachhandel
Preis: 84,95 DM

Positiv:

- geniale Zwei-Spieler-Option
- viel Spiel fürs Geld
- gelungene Spielatmosphäre
- gute Bedienerführung

Negativ:

- Monstergrafik stellenweise unzulänglich



Speichererweiterungen

Amiga 500 - 512 KB intern, abschaltbar	ab DM	179,-
Amiga 500 - 1 MB intern, abschaltbar, mit Uhr	ab DM	449,-
Amiga 500 - 1.5 MB intern, abschaltbar, mit Uhr	ab DM	569,-
Amiga 500 - 2 MB intern, abschaltbar, mit Uhr	ab DM	689,-
Amiga 1000 - 512 KB intern, abschaltbar	ab DM	398,-
Amiga 1000 - 2 MB extern, abschaltbar	ab DM	889,-
Amiga 2000 - 2 MB intern, abschaltbar	ab DM	998,-
Amiga 2000 - 4 MB intern, abschaltbar	ab DM	1.698,-
Amiga 2000 - 8 MB intern, abschaltbar	ab DM	2.998,-

Alle Speichererweiterungen sind autokonfigurierend, abschaltbar und mit sehr schnellen RAM's (100ns und schneller) ausgerüstet.
Durch Megabit-Technologie minimaler Strombedarf.
*****12 Monate Garantie*****

Floppylaufwerke u. Disketten

Laufwerk 3.5" intern f. Amiga 2000	DM	198,-
Laufwerk 3.5" extern, durchgeschl. Bus, abschaltbar	DM	249,-
Laufwerk 5.25" extern, durchgeschl. Bus, abschaltbar	DM	298,-
3.5" Disketten 2DD 135 tpi 10St. / 16,90 50St. / 82,-	100 St.	DM 159,-
5.25" Disketten 2D 10 St. / 6,90 50St. / 33,-	100 St.	DM 59,-

B & S Computer-Vertriebs GmbH
Beethovenstr. 33 ; 4172 Straelen 1
Tel: 02834/1249 ; Fax: 02834/6979

5000

3000

AMIGA MS-DOS

PUBLIC DOMAIN

**KOPIERT AUF 3,5" UND 5,25"
2DD-DISKETTEN ZU TOP-PREISEN**

**100 Leerdisketten (3,5"-2DD)
in 10er Packs DM 170,-**

INFO-MATERIAL VON:
A.P.S.-ELECTRONIC
Sonnenborstel 31, 3071 Steimbke
TEL. 0 50 26/17 00 FAX 0 50 26/16 15

PD-Schnell...Versand!!

AMIGA-PD-Software

Komplett!!!

20 PD-TOP-Disketten, z.B.:

Anti-Virus-Disk., Haushaltsbuch, Kontoführung, Textverarbeitung, mCAD, Spiele, Vokabeltrainer, Schallplatten-Videodatei, Datenbank, Adreßverwaltung, Utilities, Schach-, Musikprogramm usw.... nur **79,- DM**

512 K SPEICHERERWEITERUNG

abschaltbar nur **189,- DM**
abschaltbar, mit Uhr nur **210,- DM**

Das goldene PD-Buch + 10 Disketten nur 105,- DM

MONEY-Player

Geldspielgerät wie in der Spielhalle, Risikoautomatik etc., das ORIGINAL nur **39,- DM**

2 Katalogdisketten mit vielen Infos und Diskettenbeschreibungen (bar, Briefmarken) **5,- DM**
Preise zzgl. Versand

Hard-Software SCHOLLE 02 34/77 03 88
Pilgrimstr. 6, 4630 Bochum 1 bis 21.00 Uhr

RUHRSOFT

0234/411958

Erfahrung und Qualität hat ihren Preis
Wir kopieren auf farbigen 2DD-Disketten

EIGENE SERIEN:	Deutsche Serien:
RPD - 200	Auge - 34
Ruhr - 28	Berlin PD - 32
	Taifun - 120
Importierte Serien:	Poseidon - 400
Fish - 260	ACS - 191
TBAG - 33	KICKSTART - 220
Panorama - 98	RW - 17
CC - 145	RHS - 107
FAUG - 85	RMS - 37
Erotic Bord.* - 42	KISS - 135
UKAUG - 46	CACTUS - 33
S.A.F.E. - 36!!	FRANZ - 35
Amicus - 26	ES - 75

* nur mit Altersnachweis (Super Serie)

3 DEUTSCHE Katalogdisketten DM 8,-

Ruhrsoft Scheer - Hängebank 8a - 4630 Bochum 5



Tino Hofstede
Computerservice
An der Windmühle 8
5010 Bergheim 5



WARRIORS, SF-Adventure	19,90
HOTELMASTER, Spiel	19,90
ENGLISCHTRAINER incl. Vokabeln	39,90
FRANZÖSISCHTRAINER incl. Vokabeln	39,90
VIDEOVERWALTUNG	49,90
DISK-ETIKETTEN-Druck/Programm	19,90
LIGAVERWALTUNG (6-20 Mannschaften)	49,90
PD-DISKETTEN	ab 2,60
KATALOG (Schutzgebühr in Briefmarken)	1,00

PD-SOFTWARE

für Amiga

»24-Stunden-Versandservice«

Wir haben weit über 2500 PD-Disks im Archiv
Fred Fish 1-250; Taifun 1-110; Panorama 1-32d; ACS 1-168; Chiron 1-115; TBAG 1-32; Amuse 1-3; RPD 1-167; Tornados 1-30; R. Wolf 1-15; Auge 1-33; Amicus 1-26; Faug 1-75; Ruhr PD 1-20; Kickstart 1-200; RHS 1-90; RMS 1-25; SAFE 1-36; Cactus 1-30; E.S. 1-75 u.a.
Wir bekommen ständig die neueste PD-Software

SUPER PREISE

Alles auf 3,5"-2DD-Qualitätsdisketten
»Alle Disks sind etikettiert«

Einzeldiskette	4,-	DM/Stück
ab 20 Stück	3,50	DM/Stück
ab 40 Stück	3,20	DM/Stück
2DD-Sentinel-Qualitätsdisk.	19,50	DM/10 Stück
Viruskiller (mit Anleitung)	8,95	DM/Stück

2 Infodisketten 5,- DM

Wir kopieren auch auf 5,25" sowie auf eigene Disks (dann Preise s. o. ab 1,- bzw. 2,- DM/Stück)
Versand: NN 7,- DM; VK 4,- DM; Ausland nur VK 7,- DM

Schramm PD-Versandservice

Kai Michael Schramm
Philipp-Holl-Straße 18b, 6200 Wiesbaden
Tel. 06121/403921 oder 401709

DONAU-SOFT

24 h-Schnellversand

**Neutrale
Markendisketten**

3,5" 2DD (100 % errorfree)

	Sentinel	SONY/Collossus
bis 99 Stück	2,00 DM	2,30 DM
ab 100 Stück	1,80 DM	2,05 DM
ab 500 Stück	1,60 DM	1,95 DM

Laufwerke mit allen Extras

3.5" intern	180,- DM
3.5" extern, abschaltbar, Busdurchführung	235,- DM
5.25" extern, wie 3.5" + 40/80-Trackumschaltung	269,- DM
Sim City	97,- DM
B.A.D.	77,- DM
Pagestream	385,- DM
Zoetrope 1.1	189,- DM
GFA-Basic	189,- DM
GFA-Compiler	97,- DM
512 KB-Erw. (A500)	240,- DM
MiniMax ab	389,- DM

Vorkasse: +5,- DM, Nachnahme: +8,- DM, Ausland: +10,- DM

MAIK HAUER

Postfach 1401, 8858 Neuburg Fax: 08431/49800
Tel.: 08431/49798 (bis 22 Uhr) BTX: *Donau-Soft#

SUPER-FAT AGNUS

8372A

für A-500/2000

**1 MB adressierbares
CHIP-RAM-MEM**

Lieferung incl. Einbauanleitung

DM 193,00

CIK-Computertechnik • Ingo Klepsch
Postfach 1331 5828 Ennepetal 1
Tel. 02333/80202 Fax 02333/70345

GIERO System-Software präsentiert:

PROFESSIONAL



TOOLS

mit 90 UNIX-kompatiblen CLI-Befehlen.

- Linker
- Variable im CLI
- Update-Service
- Alles in deutsch
- Diskmonitor im CLI
- Dateibefehle aus UNIX
- Harddisk - Backup - Befehle
- Multiuser - Message - System
- Ablaufsteuerung in Batchdateien

NUR 79.00 DM

Für Schüler/Studenten 64.00 DM (Ausweiskopie).
Porto/Verp. 5.00 DM. Nachnahme 4.00 DM.

GIERO System-Software
Lessingstr. 14
3167 Burgdorf

Tel.: 05136/8 42 87



Ballerspiele gibt es reichlich, aber nur ein Xenon II, das nicht umsonst den Untertitel 'Megablast' trägt

Xenon II, The Megablast

Die Xeniten sind wieder da! Nachdem es uns vor etwa zwei Jahren gerade noch gelungen ist, die drohende Invasion dieses aggressiven Völkchens zu verhindern, sinnen die Bösen natürlich auf Rache. Sie haben sich eine neue Armada gebaut und versuchen erneut, der Erde habhaft zu werden.

Wenn es darum geht, gute Ballerspiele für den Amiga zu nennen, darf ein Name nicht fehlen: Xenon. War es doch eines der ersten Spiele seiner Art, die die Fähigkeiten des Amiga gut zu nutzen verstanden und nebenbei auch noch wirklich spannend war. Die Bitmap Brothers, die schon für den ersten Teil verantwortlich waren, haben wieder zugeschlagen und die zweite Xeniteninvasion inszeniert.

Der Spieler hat die Aufgabe, mehrere vertikal scrollende Strecken zu durchfliegen und dabei so gut wie möglich den angreifenden Xeniten Paroli zu bieten. Schon im ersten Level, in dem es noch verhältnismäßig ruhig zugeht, präsentiert das Spiel sich von seiner besten Seite. Ruckelfreies Scrolling, jede Menge rasend schneller Aliens, tolle Hintergründe und Extrawaffen, daß dem Ballerspielfreund das Herz im Leibe lacht. Gerade um die Extrawaffen haben die Bitmap Brothers sich viele Gedanken gemacht, was man dem fertigen Spiel deutlich anmerkt. Extrawaffen kann der Spieler entweder durch das Vernichten kompletter Alienformationen bekommen oder aber in einem Shop kau-

fen. In dem Geschäft kann man sogar Waffen, die nicht mehr gebraucht werden, wieder verkaufen und die Erlöse in nützlichere Extras stecken. Xenon II ist ein echter Hit und trägt den Untertitel 'Megablast' mit vollem Recht. (hs)

AMIGA DOS Blitzlicht

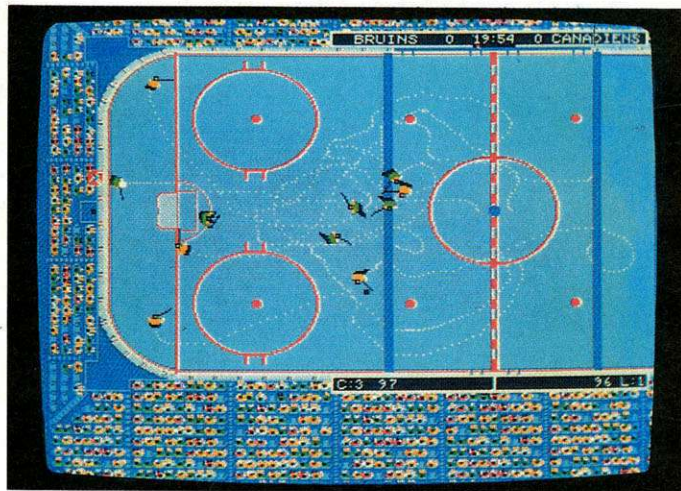
Name: XENON II
Hersteller: Image Works
Quelle: Fachhandel
Preis: 84,95 DM

Positiv:

- gute Grafik
- spitzen Extrawaffen
- super Action
- guter Sound - Continue Option

Negativ:

- stellenweise recht schwer



Heißes Match auf kaltem Eis. Wayne Gretzky Hockey begeisterte auch unsportliche Typen

Wayne Gretzky Hockey

Winter ist's geworden, und wieder entbrennt der heiße Kampf auf dem kalten Eis. Eishockey gilt nicht umsonst als eine der härtesten Sportarten, die es gibt.

Um so schöner ist es, daß man, seit es Computer gibt, Eishockey auch ohne ausgeschlagene Zähne, blaue Augen und diverse Prellungen spielen kann. Wayne Gretzky Hockey begnügt sich aber nicht damit, ein einfaches Hockeyspiel zu sein, es beinhaltet eine Vielzahl von Optionen, die nach Belieben vom Spieler eingestellt werden können. So ist es möglich, alleine oder zu zweit zu spielen, der Schwierigkeitsgrad kann eingestellt werden. Der Spieler kann die Rolle eines Trainers übernehmen und ausschließlich von der Trainerbank Spielanweisungen geben und ansonsten nur dem Match zusehen. Die Länge der Spieldrittel kann manipuliert werden. Spielergebnisse und Tabellen können auf einem Drucker ausgegeben werden, und sogar an eine Option zum Abspeichern von angefangenen Spielen, die später beendet werden sollen, wurde gedacht.

Das Spielfeld wird in der Draufsicht dargestellt und der Spieler kann wahlweise über Joystick oder Maus seine Mannschaft kontrollieren. Je ein Sportler kann unmittelbar gesteuert werden, während die anderen, vom Computer gelenkt werden und entsprechend weniger aktiv sind.

Alles in allem ist Wayne Gretzky Hockey ein gutes Sportspiel, das nicht nur das Flair des Eishockeysports einzufangen vermag, sondern auch spielerisch überzeugen konnte. (hs)

AMIGA DOS-Blitzlicht

Name: Wayne Gretzky Hockey
Hersteller: Bechesda Softworks
Quelle: Fachhandel Softworks
Preis: ca. 80 DM

Positiv:

- gute Grafik
- viele einstellbare Optionen
- überzeugende Animation
- unterstützt Drucker

Negativ:

- stellenweise unübersichtlich





Der Ober-Abenteurer in Aktion. Diesmal ist Indiana Jones auf der Suche nach dem legendären Heiligen Gral

Indiana Jones and the last Crusade

Noch bevor der langerwartete dritte Teil der Indiana-Jones-Filmserie in deutsche Kinos gelangt ist, gibt es ein Actionspiel mit Dr. Jones in der Hauptrolle!

Hollywoods Wunderknabe hat wieder zugeschlagen. Steven Spielberg hat den langerwarteten dritten Teil der Indiana-Jones-Serie gedreht. Und wie es schon fast zur lieben Gewohnheit geworden ist, folgt dem außerordentlich erfolgreichen Film das dazugehörige Computerspiel. Die einzelnen Spielszenen sind der Filmhandlung entlehnt, und das erste Level führt den Spieler dann auch gleich ohne lange Umwege in die Minen von Coronado. Dort muß Indy, so lautet Indiana Jones' Spitzname, ein legendäres Kreuzifix finden. Das ganze Spiel ist als Plattformspiel im klassischen Stil angelegt. Die Schauplätze werden im Querschnitt dargestellt. Wird Indy mittels des Joysticks bewegt, scrollt der Computer die Hintergründe weiter.

In der Mine hat Indy zwei wesentliche Probleme. Einmal muß er dafür sorgen, daß er immer genügend Licht in Form einer Fackel bei sich hat, zum anderen laufen in der Mine einige Banditen herum, die entweder auf Dr. Jones schießen oder Messer nach ihm schleudern. Diesen Attacken kann Indy durch Ducken oder Springen ausweichen, mit dem Licht ist es nicht so leicht.

Giraffen, Banditen und ein Zirkuszug

Die mitgeführte Fackel brennt kontinuierlich ab, und so wird auch das umgebende Szenario nach und nach dunkler. Ist die Fackel ganz abgebrannt, erlischt sie, und Indiana Jones hat eines seiner Bildschirmleben verwirkt.

Ist es nach einigen Anläufen gelungen, durch die Mine von Coronado zu kommen und dort auch das sagenhafte Kreuz zu erlangen, geht es auf den Waggondächern eines fahrenden Zuges weiter. Dieser Zug transportiert gerade einen Zirkus zu dessen nächster Station. Hier muß Indiana Jones sich vom Ende des Zuges bis zum ersten Waggon vorkämpfen, was natürlich nicht so einfach ist, wie es sich anhört. Die Zwischenräume zwischen den einzelnen Waggons muß Indy überspringen. Weitere Hindernisse, wie Giraffen und Nashörner, die ab und an aus den Waggons heraus schauen, erschweren die Zugüberquerung zusätzlich.

Außerdem treiben sich natürlich auch auf den Zugdächern die inzwischen sattam be-

kannten Banditen herum, mit denen in gewohnter Weise (gezielter Hieb auf die Kinnspitze) verfahren werden sollte. Hat Indy den vordersten Waggon des Zuges erreicht, ist Level eins geschafft.

Die Gewölbe der Burg Brunwald

Die nächste Station auf der Suche nach dem legendären Kleinod ist eine Burg in Österreich, in der Indiana Jones seinen Vater befreien muß. Das Burgszenario präsentiert sich grundsätzlich wie die Hatz durch die Mine, nur das hier ein ganz neues und in anderer Grafik gestaltetes Gangsystem entworfen wurde. Auch neue Gegner erwarten den Spieler in Level zwei. Diesmal muß Indy Ratten und Feuerbällen ausweichen.

Das Action-Spiel Indiana Jones hat seine Reize. Die ansprechend gestalteten Hintergründe, die großen, sauber animierten Spielfiguren und ein, zwar älteres, aber immer noch interessantes Spielkonzept. Weniger gut hat uns gefallen, daß die einzelnen Schauplätze wenig komplex sind und der Spieler die genaue Lage spielrelevanter Stellen recht schnell auswendig kennt.

(hs)

AMIGA DOS Blitzlicht

Name: Indiana Jones and the last Crusade
Hersteller: Lucasfilm
Quelle: Fachhandel
Preis: 69,95 DM

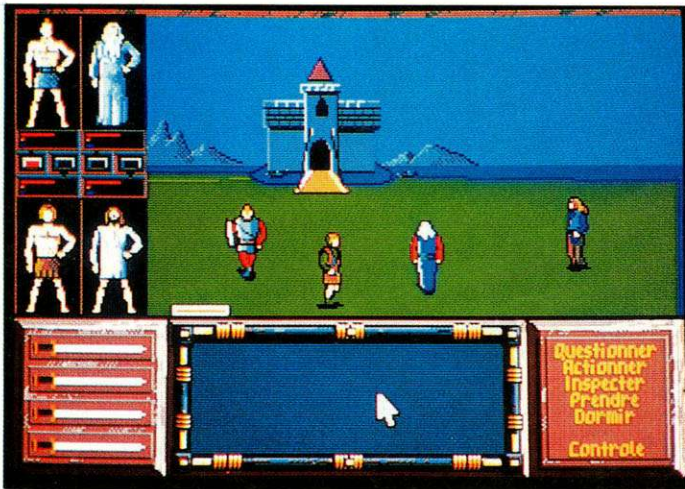
Positiv:

- gute Grafik
- gute Animation
- reizvolle Spielatmosphäre

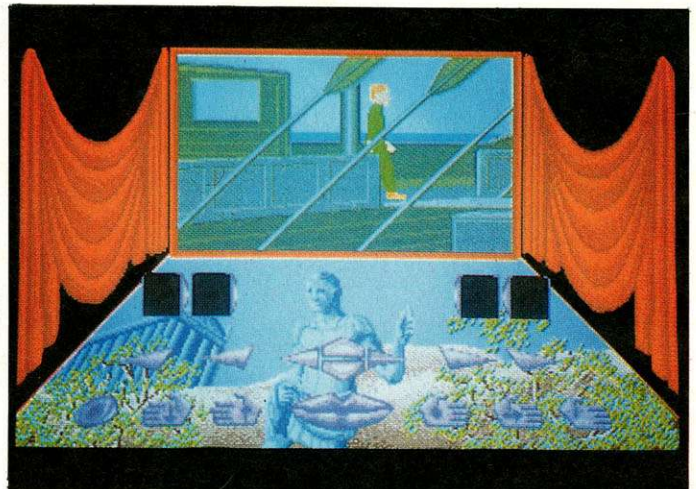
Negativ:

- zu kurze Level





Ein Rollenspiel mit 3-D-Szenario – Drakkhen macht's möglich



Quer durch alle Epochen der Weltgeschichte führt T.I.M.E., ein neues Adventure

Drakkhen

Aus Frankreich kommt Drakkhen. Der Spieler wird in eine phantastische Welt versetzt, die von acht Drachen beherrscht wird. Jeder dieser Drachen bewacht ein magisches Juwel, das der Spieler benötigt, um Drakkhen zu lösen. Bei diesem Spiel soll es möglich sein, zwei Amigas aneinander zu koppeln und simultan auf zwei Computern an ein und derselben Partie zu spielen. Die Darstellung der Landschaft erfolgt in Form eines 3-D-Window. Natürlich findet sich auch in Drakkhen ein Magiesystem, durch das sich mehr als 200 verschiedene Zaubersprüche benutzen lassen.

T.I.M.E.

Auf eine abenteuerliche Reise durch Raum und Zeit entführt Sie T.I.M.E., das neue animierte Adventure von Empire. Das Spiel beginnt auf einer

Demnächst auf Ihrem Computer

Weihnachtszeit, Softwarezeit. Während in den Sommermonaten Neuerscheinungen unter den Computerspielen eher rar sind, rüstet sich nun jeder für das kommende Weihnachtsgeschäft. Wenn Sie wissen möchten, was da so alles auf uns zukommt, lesen Sie weiter...

großen Raumstation. Dort erfährt der Spieler von seiner Aufgabe: Er muß einen perfekten Androiden zusammenbauen. Leider wurden die Einzelteile der Maschine durch einen schlimmen Unfall quer durch die Geschichte verteilt und der Spieler muß mit einer Zeitmaschine durch die Epochen der Weltgeschichte reisen und die verstreuten Teile einsammeln. Die Suche führt den Spieler in den Zirkus Maximus im antiken Rom, nach Camelot, der

Residenz des legendären König Arthur und an viele andere bekannte historische Orte.

European Superleague

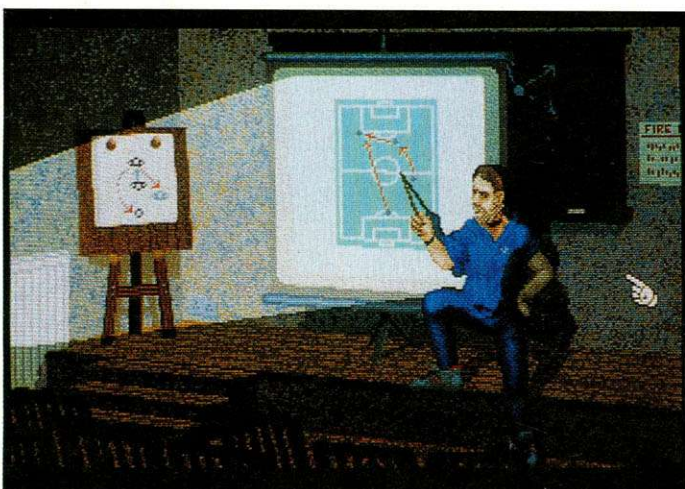
European Superleague versetzt den Spieler in die Rolle des Managers eines Fußballclubs. Während die Spieler versuchen müssen, für den Club den Europacup zu gewinnen, ist es Ihre Aufgabe, den Club zu managen. Zu Ihrem Pensum gehört es, Spieler einzu-

kaufen, Werbeverträge abzuschließen und im Umkleide-raum bei den Spielern für eine gute Atmosphäre zu sorgen.

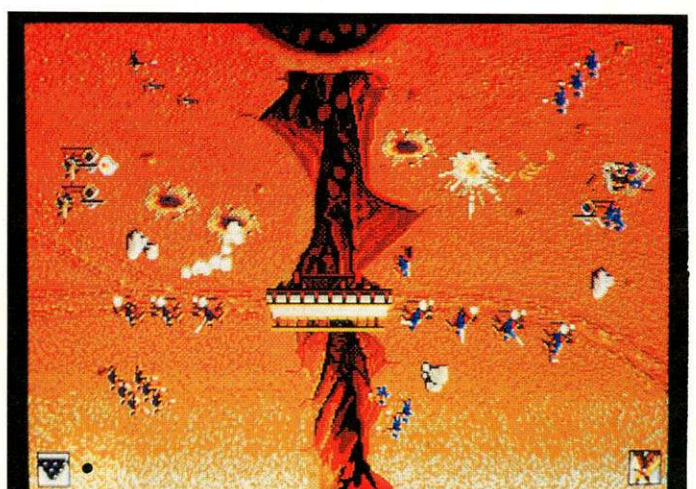
North and South

Nicht nur die Fernsehserie Fackeln im Sturm beschäftigt sich mit dem Sezessionskrieg. Mit North and South findet diese Thematik nun auch Einzug in die Reihen der Computerspiele. Wer hier allerdings eine knallharte Taktik und Schlachtensimulation erwartet, wird enttäuscht werden. Zwar beinhaltet das Spiel auch eine ganze Menge Taktik, präsentiert sich aber in comic-haftem Stil. Kein Wunder, denn für dieses Spiel stand eine in Frankreich sehr erfolgreiche Comic-Serie Pate. Der Spieler muß sich für eine Partei entscheiden, und, wie sollte es anders sein, für diese den Bürgerkrieg gewinnen.

(hs)



Wird der Verein es schaffen, den Europacup zu erringen



Bürgerkrieg als Computerspiel. North and South nimmt die Sache jedoch nicht allzu ernst

Ziellos? Vielleicht! Aber wie der Zufall es will, kreuzt das kleine Boot gerade die Kiellinie eines intergalaktischen Müllschiffes, dessen Aufgabe im Freihalten der Raumstraßen und im Einsammeln von Müll besteht.

Doch wenden wir unsere Aufmerksamkeit kurz einer Holografiesendung zu, die gerade auf einem Planeten des Andromeda-Systems in allen Haushalten empfangen werden kann. "Wir unterbrechen unser Programm für eine wichtige Meldung. Unbestätigten Gerüchten zufolge sind zwei Softwareautoren, die Two Guys from Andromeda, entführt worden! Offensichtlich wurden die beiden von Piraten entführt, die bei Scumsoft unter Vertrag stehen."

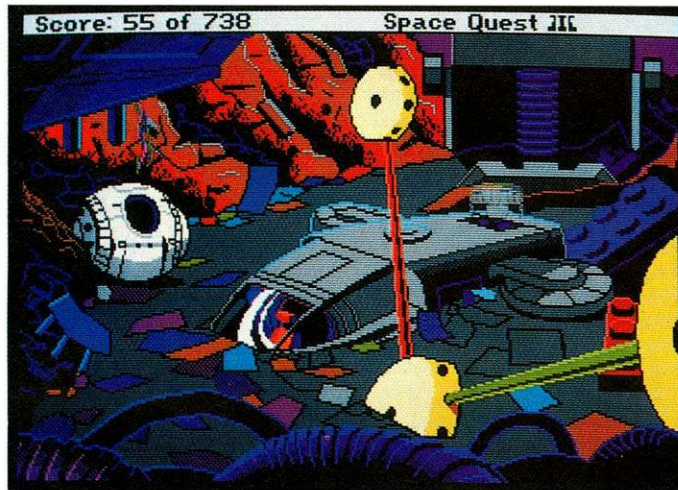
Mit Bindfaden und Leukoplast...

Ein dumpfer Aufschlag reißt Roger Wilco unsanft aus seinen Träumen. Die Reise ins Ungewisse hat offensichtlich ein Ende gefunden. Nach dem Verlassen des kleinen Schiffes enthüllt sich eine malerische Szenerie. Berge von Müll umgeben unseren Helden.

Eine genaue Durchsuchung der Räumlichkeiten bringt manch brauchbaren Gegenstand ans Licht. Der Draht, der in der Passage zu finden ist, ist sicher ebenso brauchbar, wie der Reaktor, der einige trübe Funzeln mit der notwendigen Energie versorgt. Nach der Entdeckung des Schienenfahrzeuges geht die Arbeit schon wieder leichter von der Hand. Mit seiner Hilfe ist es nun möglich, den Warp-Motor in das am besten erhaltene Wrack einzusetzen. Nachdem mit Hilfe der Leiter auch die Innenbereiche des Wracks zugänglich sind, lassen sich mit dem Reaktor und dem Draht die notwendigen Reparaturen leicht ausführen.

Phleebut - ein Sightseeing-Paradies

Zunächst steht nach dem erfolgreichen Verlassen des Frachters ein Besuch auf dem Planeten Phleebut auf der Tagesordnung. Der einzige dort ansässige Händler hält einige interessante Artikel bereit, die sicherlich einmal nützlich sein können. Die aus Space



Wird es Ihnen gelingen, aus den vorhandenen Schrottbergen ein funktionables Raumschiff zusammenzustellen? Eigentlich haben Sie keine andere Wahl, wenn Sie den Schrottfrachter je wieder verlassen wollen...

Space Quest III He's Back!

In Space Quest II hatte Roger Wilco, der Held der Space Quest Serie, gerade den bösen Vohaul samt seiner Asteroidenfestung im Vakuum des Alls verteilt. Knapp gelang die Flucht in einem kleinen Raumschiff. Roger suchte die Schlafkammer auf, um Sauerstoff zu sparen. Seit dieser Zeit treibt nun das kleine Schiff ziellos durchs Universum...

Quest II mitgebrachte Gem ist zumindest auf Phleebut ein akzeptables Tauschmittel.

Der Terminator, der Roger beim Verlassen des Ladens auflauert, will die Schulden eintreiben, die sich seit den Tagen von Space Quest II auf zirka 40000 Buckazoids summiert haben. Die eingeräumten zehn Sekunden Vorsprung können gut genutzt werden, um die Statue zu erklimmen. Dort oben bietet sich dem aufmerksamen Betrachter eine

gute Möglichkeit, sich des Terminators zu entledigen und einen Gürtel an sich zu bringen. Kämpfen macht bekanntlich hungrig, und so sollte der nächste Besuch doch der Monolith-Burger-Station gewidmet sein. Das dort feilgebotene "Fun-Meal" hat es im wahrsten Sinne des Wortes in sich. Nach dem Essen kommt die leichte Muse zu ihrem Recht. Einige Partien Astro-Chicken offenbaren nicht nur ein neues Video-

spiel, sondern einige interessante Informationen für Roger. Denn noch immer schmachten die zwei gefangenen Programmierer ihrer Rettung entgegen.

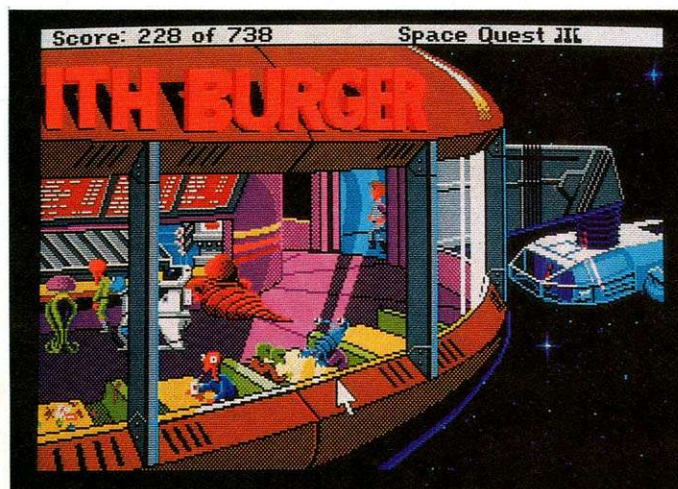
Die Hinweise aus dem Videospiel bringen Roger der Lösung dieses Problems jedoch schon erheblich näher...

Die Rettung der Programmierer

Richtig ausgewertet, birgt das Videospiel alle Informationen, die Roger benötigt, um die Programmierer aufzuspüren. Doch wird es auch gelingen, sie zu befreien? Dies dürfte wesentlich von Ihnen abhängen, denn irgend jemand muß ja Roger bei seiner schwierigen Aufgabe unterstützen. Und wo die List versagt, kann (manchmal) rohe Gewalt helfen.

Space Quest III ist mit Fug und Recht eine der besten Umsetzungen für den Amiga. Die Grafiken sind faszinierend, und die Animation zeichnet sich durch flotte Geschwindigkeit und sehr realistische Bewegungsabläufe aus. Und wer eine Stereoanlage an seinen Amiga angeschlossen hat, kommt in den vollen Genuß der Musik, die Rogers Abenteuer untermauert. Einfach ein Muß für Space-Quest- und Sierra-Fans!

(mm)



Ein Imbiss bei Monolith Burgers kann schwer im Magen liegen

AMIGA DOS Blitzlicht

Name: Space Quest III
Hersteller: Sierra Online
Vertrieb: Fachhandel
Preis: ca. 120 DM

Positiv:

- liebevoll gestaltete Grafiken
- hörenswerter Sound
- hohe Motivation
- verbesserte Animationen

Negativ:

- stellenweise zu einfach

AMIGA
DOS
URTEIL

GRAFIK SOUND MOTIVATION

7 7 9



Impressum

Herausgeber

Christian Widuch

Chefredaktion

Markus Matejka

Redaktion

Jürgen Borngießer (jb), Bernhard Rinke (br),
Heinrich Stiller (hs), Joachim Freiburg (jf),
Martin Schlöter (ms)

Freie Autoren dieser Ausgabe

Dipl.-Phys. Dr. Peter Kittel, Edgar Meyzis,
Bernd Rudolph, Robert Marz, Jürgen Seibel,
Michael Winterberg, Klaus Rupp, Garry Glendown,
Robert Haas, Peter Woop, Oliver Heggelbacher,
Hermann-Joseph Cuylen

Korrespondent USA

Marshall M. Rosenthal

Redaktionsassistentz

Anke Kerstan (ke), Susanne Eska (es)

Koordination

Stefan Ritter

Produktionsleitung

Gerd Köberich

Bereichsleitung

Christian Heckmann (Fotografie/Layout),
Claudia Küllmer (Fotosatz/Lektorat),
Margarete Schenk, Helmut Skoupy (Montage/
Reprografie)

Layout

Yvonne Hendricks

Fotografie

Klaus Jatho, Heike Reifer

Fotosatz

Gabriela Joseph

Lektorat

Susanne Mias

Montage/Reprografie

Manuela Eska, Dieter Schnobl

Werbegestaltung

Mohamed Hawa, Petra Küch

Anzeigenverkaufsleitung

Wolfgang Schnell

Anzeigenverkauf

DMV-Verlagsbüro München
Zaunkönigweg 2c, 8000 München 82
Telefon (0 89) 4 39 10 87, Telefax 0 89/4 39 10 80
Leitung: Britta Fiebig
Anzeigenverkauf: Monika Schöbel, Jens Dhein,
Peter Schätzle

Anzeigenpreise

Es gilt die Anzeigenpreisliste Nr. 1 vom 01.01.1990

Anschrift Verlag/Redaktion:

DMV Daten & Medien-Verlag
Widuch GmbH & Co. KG
Fuldaer Straße 6
3440 Eschwege, Telefon (0 56 51) 8 009-0,
Telefax (0 56 51) 8 009-33

ab 01.01.1990

Telefon (0 56 51) 8 09-0

Telefax (0 56 51) 8 09-333

Vertrieb

Verlagsunion Erich Pabel-Arthur Moewig KG (VPM),
Friedrich-Bergius-Straße 20, 6200 Wiesbaden

Druck

Druckerei Jungfer, 3420 Herzberg

Bezugspreise

»AMIGA DOS« erscheint monatlich.
Einzelpreis DM 6,50/sfr. 6,50/6S 52,-

Abonnementpreise

Die Preise verstehen sich grundsätzlich einschließlich
Porto und Verpackung.

Inland:

12 Ausgaben: DM 70,-

6 Ausgaben: DM 35,-

Europäisches Ausland:

12 Ausgaben: DM 100,-

6 Ausgaben: DM 50,-

Außereuropäisches Ausland:

12 Ausgaben: DM 120,-

6 Ausgaben: DM 60,-

Bankverbindungen:

Postscheck Frankfurt/M: Kto.-Nr.: 23043-608

Raiffeisenbank Eschwege:

BLZ: 522 603 85, Kto.-Nr.: 245 7008

Die Abonnementbestellung kann innerhalb einer
Woche nach Auftrag beim DMV-Verlag, Postfach 250,
3440 Eschwege, schriftlich widerrufen werden. Zur
Wahrung der Frist reicht der Poststempel. Das Abonne-
ment verlängert sich automatisch um 6 bzw. 12 Monate,
wenn es nicht mindestens 6 Wochen vor Ablauf beim
Verlag schriftlich gekündigt wird.

Für unverlangt eingesandte Manuskripte und Datenträger
sowie Fotos übernimmt der Verlag keine Haftung.
Die Zustimmung zum Abdruck wird vorausgesetzt.
Eine Haftung für die Richtigkeit der Veröffentlichungen
kann trotz sorgfältiger Prüfung durch die Redaktion vom
Herausgeber nicht übernommen werden. Die geltenden
gesetzlichen und postalischen Bestimmungen sind zu
beachten.

Die gewerbliche Nutzung, insbesondere der Programme,
Schaltpläne und gedruckten Schaltungen, ist nur mit
schriftlicher Genehmigung des Herausgebers zulässig.
Das Urheberrecht für veröffentlichte Manuskripte liegt
ausschließlich beim Verlag. Nachdruck sowie Vervielfäl-
tigung oder sonstige Verwertung von Texten nur mit
schriftlicher Genehmigung des Verlages.

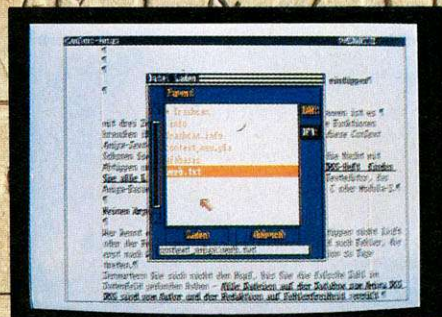
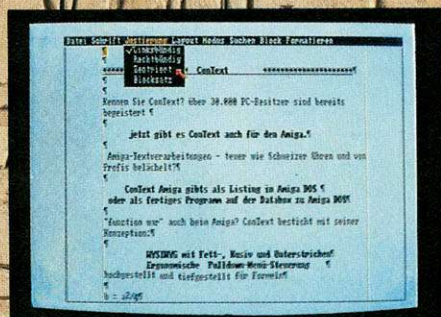
Namentlich gekennzeichnete Fremdbeiträge geben nicht
in jedem Fall die Meinung der Redaktion wieder.

Die Inserenten

ABAG	49	Gold-Vision	137
A + L	113	Hima Computertechnik	31
Alpha 2000	137	HK-Computer	59
A.P.S Electronic	139	Hofstede	139
ARIOLASOFT	115	IDS Hard + Software	57
B + C EDV-Systeme	137	Interest Verlag	75
B + S Computer	139	KONYO	49
CIK Computertechnik	139	Lechner Verlag	13
Commodore	19	Merkens-EDV	51
Compedo	67	Message-Computer	148
CompuCamp	Postkarte	Nürnberger PD Studio	135
Compy-Shop	29	Ossowski	147
CPS-Computertechnik	45	Reis-Ware	51
CSV Riegert	135	Ruhrsoft	139
CWTG	127	Sanz Software	127
Data Becker	70,71	Schewe	27
Diezemann	81	Scholle	139
DMV	22,23,123,129	Schramm	139
Donau-Soft	127,139	SCS Schomburg	99
3-State Computertechnik	67	Sky Ware	29
Ebensberger	99	Soft Shop	81
EZ Appel + Grywatz	113	Stalter	27
Funhouse	49	Strauß Elektronik	7
GFA-Systemtechnik	2	Teletone	137
Giero Systemsoftware	139	TS-Datensysteme	27

Textverarbeitung im Wandel der Zeit

CONTEXT AMIGA



Stele im Besitz des Kestner-Museums, Hannover

In der PC-Welt ist ConText bereits ein Begriff für Textverarbeitung:

Mehr als 30.000 Anwender sind von ConText PRO und ConText EASY begeistert und jeden Tag werden es mehr.

Jetzt können Sie die einzigartige Konzeption von ConText auch auf dem Amiga erleben.

Amiga DOS bringt ConText Amiga als Listing zum Abtippen oder als fertiges Programm auf der Diskette über den Databox-Service.

Was ist am ConText-Konzept so einmalig?

Zuerst einmal das WYSIWYG-Prinzip: Sie sehen am Bildschirm Fettschrift auch wirklich fett und kursiv als kursiv - ohne die sonst üblichen, zumeist unverständlichen Steuerzeichen.

Dann die doppelte Ansteuerung der Funktionen über eine mausbedienbare Menüleiste und Tastenkombinationen: So kommen Sie immer auf dem schnellsten und bequemsten Weg zum Ziel.

Anwenderfreundliche und durchdachte Funktionen, die leicht zu bedienen und zu verstehen sind, machen Schluß mit dem "function war", der den Anwender mehr überfordert als ihm nutzt.

ConText Amiga zum ersten

In dieser Ausgabe von Amiga DOS stellen wir Ihnen das Grundprogramm von ConText Amiga als Listing mit folgenden Funktionen vor:

Datei:	neu, laden, speichern, quit, ConText-Format, Layout-Format, ASCII-Format
Schrift:	normal, fett, unterstrichen, kursiv, hoch, tief
Justierung:	linksbündig, rechtsbündig, zentriert, Blocksatz
Layout:	linker Rand, rechter Rand, Zeilenzahl
Modus:	insert, overwrite
Suchen:	suchen, ersetzen, weitersuchen
Block:	Start, Ende, kopieren, verschieben, löschen, freigeben
Formatieren:	Absatz

ConText Amiga zum zweiten

Im Heft 2/90 von Amiga DOS werden Sie, ebenfalls als Listing, ein separates Druckprogramm zu ConText Amiga finden, mit dem Sie Ihre Texte entsprechend zu Papier bringen können.

Separat deshalb, damit ConText Amiga für Sie im Hintergrund drucken kann, während Sie im Vordergrund weiter Ihrer Arbeit nachgehen können.

ConText Amiga zum dritten

Natürlich bleibt es nicht dabei. - Wir arbeiten bereits an einer ConText-Amiga-Profi-Version, die nicht nur die Möglichkeiten der PC-Version wie Breitschrift, Fußnotenverwaltung, Serienbriefe etc. umfassen wird, sondern diese noch übertreffen soll.

Neben eigenen Schriftfonts wird es auch möglich sein, Bilder in den Text zu integrieren und damit DTP-Dokumente zu erstellen. Bleiben Sie am Ball: ConText Amiga - Ihr Programm für Textverarbeitung.

So bekomme ich ConText Amiga

In Amiga DOS 1/90 und 2/90 finden Sie die Listings in GFA-Basic Amiga.

Das Listing und das lauffähige, kompilierte Programm finden Sie auf der Databox zu Amiga DOS, ein Service, der alle Listings des Heftes auf Diskette anbietet.

Bitte benutzen Sie die Databox-Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV
Daten- und
Medienverlag

Im nächsten Heft

■■■ **Schwarz auf weiß!** Welche Qualität weisen Drucker im unteren Preisbereich auf? Lohnt es sich für jeden Anwender, einen semiprofessionellen Drucker anzuschaffen? AMIGA DOS ist diesen Fragen nachgegangen. Im nächsten Heft stellen wir Ihnen einige Drucker im Preisbereich unter 1000 DM vor.



Ein Drucker im unteren Preisbereich – Wir haben uns den LQ 400 von Epson näher angesehen

■■■ **Textverarbeitungen** Das neue UBM-Text 2.3 verfügt jetzt auch über die Möglichkeit, Grafiken mit in Texte einzubinden. AMIGA DOS hat sich das Update der bekannten Textverarbeitung einmal für Sie angesehen.

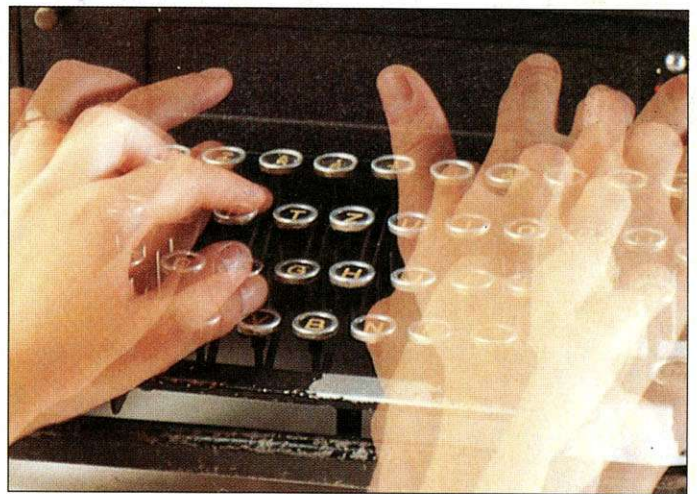
■■■ **Spiele** Mit Pharaos liegt ein Action-Adventure aus der Zeit des alten Ägyptens vor. Beginnen Sie die Eroberung des Pharaonenreiches auf der untersten sozialen Stufe, und steigen Sie auf bis zum Pharaos von Ober- und Unterägypten!

■■■ **Amiga Big Band** Der Soundeditor SIDMON setzt neue Akzente in der Musikerstellung! Mit 100 Instrumenten wird Ihr Amiga zum Sinfonieorchester!

■■■ **Listing** Sprites, Sprites, Sprites. Wer Spiele programmieren möchte, der benötigt in den meisten Fällen Sprites. Leider ist das richtige Handling von Sprites nicht unproblematisch. AMIGA DOS hat wesentliche Fakten zu diesem Thema zusammengetragen.

■■■ **Grundlagen** Kaum ein Computer eignet sich besser für Grafik und Animation als der Amiga. Wissenswerter zu diesem Thema im nächsten Heft.

■■■ **Und natürlich** wieder jede Menge Tips&Tricks zu AmigaBASIC, ASSEMBLER, Modula-2 und C!



Textverarbeitung zu Großvaters Zeiten – UBM-Text 2.3 kann mehr

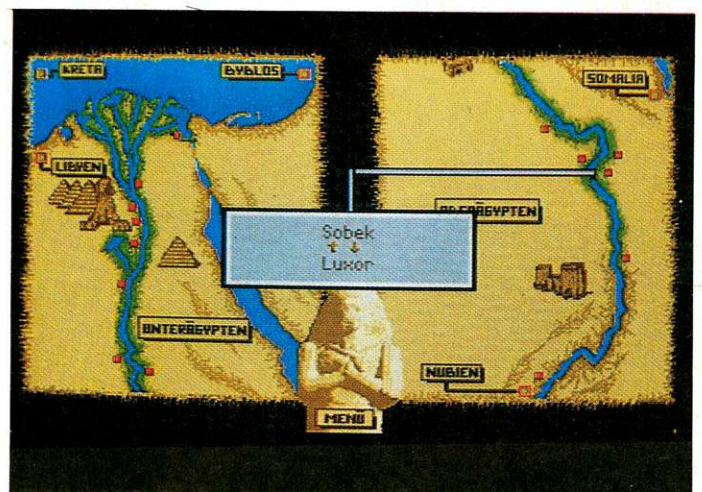
Die nächste



finden Sie ab

10. Januar

bei Ihrem Zeitschriftenhändler



Pharaos entführt Sie in die Welt des alten Ägyptens

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,-
- ㉕ **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 **DiskKey** 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,-

* 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. 260 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-Versand Stefan Ossowski
Testen Sie uns!

Professionelle Anwendungssoftware für den anspruchsvollen User:

- ⑩⑩ **DME T & W = Text & Wörterbuch**
 DME T & W ist ein leistungsstarker Texteditor sowohl für den Viel- als auch für den Gelegenheitsschreiber. DME T & W ist äußerst flexibel und sogar programmierbar. Sie erhalten DME T & W bei uns mit **deutscher Anleitung** und einem eingebauten **Deutsch-Englisch-Wörterbuch** mit ca. 3000 Vokabeln. DM 30,-
 DME mit englischer Anleitung und ohne Wörterbuch: DM 6,-
- ⑩① **RIM-5 = Relationale Datenbank**
 RIM-5 ist eine äußerst leistungsfähige relationale Datenbank, die sowohl für den geschäftlichen als für den privaten Bereich prädestiniert ist. Natürlich liefern wir RIM-5 mit **deutscher Anleitung** und einem **ausführlichen Einführungskurs**. DM 30,-
- ⑩② **AnalytiCalc = Tabellenkalkulation**
 Eine leistungsstarke Tabellenkalkulation mit **deutscher Anleitung**. Schon nach kurzer Zeit wird AnalytiCalc ein unverzichtbares Hilfsmittel für Kalkulationsaufgaben im Büro oder zu Hause sein. DM 30,-
- ⑩③ **DEA Arithmetica = Die Göttin Arithmetica**
 DEA Arithmetica ist ein Programm, auf das Schüler, Studenten und Lehrer schon lange gewartet haben. Ableitungen, Kurvendiskussionen und das Skizzieren von Funktionsgraphen stellen von nun an kein Problem mehr dar. **Made in Germany!** DM 30,-
- ⑩④ **Haushaltsbuch Version 2.0**
 Die Komplettlösung zur Verwaltung Ihrer privaten Finanzen. Wollen Sie einen Überblick über Ihre Finanzen, Ihr Geld effektiver nutzen und Schwachstellen in Ihrer Budgetierung erkennen? Haushaltsbuch V. 2.0 ist leicht bedienbar und auch für Computerneulinge hervorragend geeignet! Demo-Dateien unterstützen sie bei Ihren ersten Schritten und das Programm ist frei von buchhalterischen Fachausdrücken. Ein Funktionsüberblick: Führen von verschiedenen Dateien, mehrere Kontenlisten, bis zu 10 Bilanzen, 40 freidefinierbare Konten, Suchroutinen, viele Voreinstellungen, doppelte Buchführung, Datei-Ex- und -Import, Filterfunktionen, Jahresübertrag, Mausunterstützung ... **Selbstverständlich mit umfangreicher deutscher Dokumentation!** Haushaltsbuch V 2.0 benötigt 1 MB Speicher. DM 98,-
- ⑩⑤ **Xytronic II**
 Bei diesem Spiel können Sie intergalaktischen Handel betreiben. Um Ihren Reichtum zu mehren, müssen Sie geschickt kaufen und verkaufen und natürlich den Weltraumpiraten aus dem Weg gehen. Sehr gute Grafik und sehr guter Sound garantieren zusätzlich eine hohe Motivation. DM 29,-
- ⑩⑥ **Data-Manager**
 Eine sehr leicht zu bedienende Dateiverwaltung, mit der Sie Adressen, Mitgliederlisten etc. perfekt verwalten können. Ein **deutsches Programm** mit **deutscher Anleitung**. DM 15,-
- ⑩⑦ **EGOS = European Game of Strategy**
 Ein Spiel für 2 Spieler. Sie kämpfen um die Vorherrschaft in Europa. Versuchen Sie, die Taktik Ihres Gegners zu durchschauen und Ihre eigene Strategie entsprechend auszurichten. Ein Muß für Strategiefans. **Deutsch!** DM 29,-
- ⑩⑧ **Bundesliga 2000**
 Ein leistungsstarkes Programm zur Bundesliga-Ergebnisverwaltung. Auf einer Diskette können Sie bis zu 40 Ligen verwalten. Mit Historie seit 1985. **Komplett in Deutsch!** DM 15,-
- ⑩⑨ **Money Player Deluxe – Geldspielgerät**
 Sie fühlen sich wie in der Spielhalle! Start- und Risikoautomatik, Sonderspiele, viele Extras und ein toller Spielspaß. Spielkomfort durch Maussteuerung, Pal-Auflösung ... DM 39,-
- ⑩⑩ **Wizard of Sound 2.0**
 WoS ist ein phantastisches Musik- und Komponierprogramm. Erstellen Sie eigene Lieder entweder durch Noten und per Klaviatur. **WoS** verfügt über 60 verschiedene Instrumente, einen Instant-Replay-Modus und eine große Funktionsvielfalt. **WoS wurde in Deutschland entwickelt und wird mit umfangreicher deutscher Dokumentation** komplett auf 2 Disketten geliefert! Inkl. Demosongs und Player DM 35,-
- ⑩⑪ **Broker 2.0 + Das Börsenprogramm**
 Broker 2.0 ist eine sehr **realistische Börsensimulation**. Alleine oder mit bis zu 3 Mitspielern versuchen sie Ihr Kapital zu mehren. Nur einer kann gewinnen! Dieses Spiel ist unterhaltsam, lehrreich, zeichnet sich durch lang anhaltende Motivation aus – 1MB Speicher DM 29,-
- ⑩⑫ **Vokabel-Trainer-English 1.0**
 Der **Vokabel-Trainer** ist ein **einfach zu bedienendes, leistungsfähiges Lernprogramm**. Zusätzlich zu den bereits vorhandenen Vokabeln kann man den Wortschatz leicht erweitern. ein Programm, mit dem das Lernen Spaß macht! **Made in Germany!** DM 15,-
- ⑩⑬ **Pente**
 Ein „5-Gewinnt-Spiel“, erweitert um einige Regeln. Sie spielen gegen den Computer. Für jung und alt mit hoher Motivation DM 19,-

Die Programme 100 bis 103, 105 bis 108 und 110 bis 113 erhalten Sie in dieser Ausführung ausschließlich beim PD-Vertrieb Stefan Ossowski!

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 1000 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

Heiße Tipps für Weihnachtsmänner ...

SCASI-Drive
ab DM 998,-
SCSI-Autoboot-Harddisk
für A 500/2000

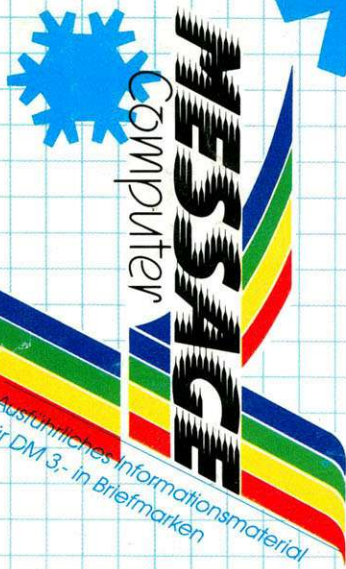
M-512 c
SUPERPREIS ...
Anruf lohnt!
512 KByte Speichererweiterung
für A 500, incl. Echtzeituhr,
abschaltbar

MIDFACE PLUS
DM 129,-
MidiInterface für
A 500/1000/2000

Crocodile-Box
DM 398,-
Expansion-Box mit 3 Slots
Harddiskmontageplatz
für A 500

Bestellung und Versand:
telefonisch: **0208-2 40 47**
per FAX **0208-2 40 49**
per BTX **020824049**

oder schriftlich:
Message-Computer
Thomas Martin
Stöckmannstr. 78
4200 Oberhausen 1



Ausdrückliches Informationsmaterial
für DM 3,- in Briefmarken