

AMIGA DOS

9/90

ISSN 0937-2717
DMV-Verlag

TOS statt DOS

– *Der ST-Emulator*

VorecOne

– *Das Ende der Maus?*

PD-Power

- *Anwendungen*
- *Spiele*
- *Utilities*

Super-Spiele

- *Tests*
- *Helpline*



- Entwicklung
- Produktion
- Hardware
- Software
- Service

HK-Computer

Ihr Amiga-Spezialist



Professional Drive Diskettenlaufwerke

- * Top in Qualität, Funktion und Design *
- 3 1/2" Laufwerk AMIGA 2000 intern DM 159,—
- komplett mit Einbautkit und Anleitung
- 3 1/2" Laufwerk für alle AMIGAs extern DM 199,—
- abschaltbar, Busdurchführung, AMIGAfarben
- 5 1/4" Laufwerk für alle AMIGAs extern DM 259,—
- abschaltbar, Busdurchführung, 40/80 Tracks, AMIGAfarben

Professional RAM-Board II

- A500 auf 1 MB** DM 169,—
- superschnelle Megabit-RAMs (4*514256)
- mit Uhr & Datum
- Hard- und Softwaremäßig abschaltbar
- Superniedriger Stromverbrauch
- dto. Platine mit Uhr & Schalter ohne RAMs DM 79,—

Professional RAM-Board III

- A500 auf 2,3 MB** DM 555,—
- intern, incl. Gary-Adapter
- superschnelle Megabit-RAMs (16*511000)
- mit Uhr & Datum
- Hard- und Softwaremäßig abschaltbar
- dto. Platine mit Uhr & Schalter ohne RAMs DM 198,—

Professional RAM-Board III B

- A500 auf 2,5 MB** DM 555,—
- mit dem neuen Big Fat Agnus volle 2,5 MB!!
- intern, incl. Gary-Adapter
- superschnelle Megabit-RAMs (16*511000)
- mit Uhr & Datum
- Hard- und Softwaremäßig abschaltbar
- Hinweis: eine Lötstelle nötig
- dto. Platine mit Uhr & Schalter ohne RAMs DM 198,—

Professional RAM Board

- A2000** DM 798,—
- 8 MB mit 2 MB bestückt, autokonfigurierend, einfach durch zusätzliche RAMs und Jumper umstecken aufrüstbar, keine neuen PALs erforderlich
- dto. Platine bestückt mit 4 MByte DM 1198,—
- dto. Platine bestückt mit 8 MByte DM 1998,—
- dto. Platine komplett ohne RAMs DM 498,—
- RAM-Satz für 2 Megabyte DM 400,—

Software

- MEDUSA der Atari ST-Emulator DM 598,—
- RAM-Test Amiga DM 24,50
- zeigt defekte Speicherstellen grafisch an, 100 % Assembler
- PACKIT DM 39,—
- Superschneller Cruncher nicht nur für Text und Grafik, sondern auch für Programme. Verschiedene Kompaktiermodi, Auto- oder Loaderstart, räumt nicht nur Ihre Disketten, sondern auch Festplatten auf.
- XCOPY II DM 49,—
- XCOPY II mit Hardwarezusatz DM 69,—
- Turboprint II DM 89,—
- Turboprint professional DM 188,—
- Quarterback (Festplatten-Backup) DM 119,—
- DPaint III DM 248,—
- Beckertext DM 189,—
- GFA-Basic 3.5 DM 229,—

AMIGA-Bremse

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

Amiga-Bremse für A500 extern mit LED

DM 69,—

Kick-ROM

DM 49,—

- Kickstartumschaltplatine für zwei Original-ROMs

Kick-ROM mit ROM
OriginalROM 1.3

DM 98,—
DM 65,—

Kickstartumschaltplatine 3-fach

DM 59,—

- für zwei OriginalROMs und eine Epromversion
- Umschaltplatine mit OriginalROM DM 108,—
- Brennservice incl. einem Epromsatz DM 79,—
- (gegen Einsendung einer OriginalDiskette)

Maus & Joystick-Adapter

DM 44,50

- für gleichzeitigen Anschluß von Maus und Joystick
- mit LED-Anzeige
- alle Maussteuerleitungen elektronisch geschaltet

PowerFire Das Superding!

DM 19,90

- 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

Trackdisplay

DM 79,—

- extern DF0: bis DF3:
- für jedes Laufwerk einstellbar

Drive-Expander

DM 39,—

- bis zu drei Laufwerke direkt am Rechner anschließbar
- einstellbare Laufwerksnummer
- keine Kabellängenprobleme
- abschaltbar
- z. B. für externe Laufwerke ohne Busdurchführung
- bei Verwendung eines Boot-Selectors kann von jedem externen Laufwerk gebootet werden

BOOT-Selector elektronisch

DM 49,—

- wahlweise Booten von allen Laufwerken

BOOT-Selector für Amigas

DM 14,50

- wahlweise Booten von DF0: oder DF1: oder DF2: oder DF3: (bei Bestellung bitte angeben)

BTX/VTX

Decoder mit FTZ (Drews)

DM 199,—

Midi-Interface

DM 89,—

- In/Thru/2*Out Im Metallgehäuse für A500/A2000

Staubschutzhäuben

- AMIGA 500 DM 16,50
- AMIGA 2000 Keyboard DM 16,50
- Monitor 14" DM 50,50
- Drucker 10" DM 24,50
- Drucker 15" DM 29,50

Die Staubschutzhäuben sind aus Kunstleder mit weichem antistatischen Innenfutter

Disketten

- 3 1/2" NoName 2DD 10 St. DM 14,95
- 3 1/2" NoName 2DD 100 St. DM 129,—
- 3 1/2" Verbatim Verex 2DD 10 St. DM 25,—
- 3 1/2" Verbatim Verex 2DD 100 St. DM 225,—
- 5 1/4" NoName 2S2D 10 St. DM 5,90
- 5 1/4" NoName 2S2D 100 St. DM 57,—
- 5 1/4" Verbatim Verex 10 St. DM 14,90
- 5 1/4" Verbatim Verex 100 St. DM 129,—

Hinweis: Alle unsere externen Geräte haben — soweit erforderlich — keine FTZ-Zulassung, wenn nicht gesondert angegeben. Ein Betrieb im Bereich der Deutschen Bundespost ist verboten.



AMIGA-Computer

- Amiga 3000-16 Mhz Preis auf Anfrage
- 68030 CPU 16 Mhz, 32 bit, 2 MB RAM, 40 MB SCSI-Harddisk
- Amiga 3000-25 Mhz Preis auf Anfrage
- 68030 CPU 25 Mhz, 32 bit, 2 MB RAM, 40 o. 100 MB SCSI-Harddisk
- Amiga 2500/30 Preis auf Anfrage
- 68030 CPU 16 Mhz, 3 MB RAM, 40 MB Harddisk
- Amiga 2000 DM 1898,—
- Amiga 500 DM 928,—
- Harddisk A590 20 MB für A500 DM 998,—
- Colormonitor Commodore 1084P DM 598,—

Preisänderung bei Festplatten

HK-COMPUTER Festplatten

Kapazität	Speed	Filecard	A2000	A500
20MB/3 1/2"	35 ms	898,—	848,—	1048,—
30MB/5 1/4"	65 ms	—	898,—	1098,—
30MB/3 1/2"	35 ms	998,—	948,—	1148,—
40MB/5 1/4"	28 ms	—	1098,—	1298,—
50MB/3 1/2"	35 ms	1198,—	1148,—	1348,—
60MB/5 1/4"	28 ms	—	1298,—	1498,—

Alle unsere Festplatten werden mit Autoboot-Software ausgeliefert.

- AUTOBOOTMODUL für A2000 DM 119,—
- AUTOBOOTMODUL für A500 DM 149,—
- AUTOBOOT-KARTE für A2090-Controller DM 119,—
- TURBO-CHIPSATZ für A2090A-Controller DM 149,—
- Alle zum Nachrüsten, incl. Software und Anleitung

Festplatten-Controller

- OMTI 5520B für MFM-Platten (20/40MB) auf Anfrage
- OMTI 5528B für RLL-Platten (30/50/60MB) auf Anfrage

Festplatten-Interface DM 99,—

Die Adapterplatine paßt den PC-BUS eines Festplatten-Controllers an den AMIGA-BUS an. (Bitte Rechnerart angeben)

Autoboot-Set MFM

- OMTI 5520B, Autoboot-Modul, Festplatten-Interface, Kabelsatz für Amiga 2000 (interne Slotkarte) auf Anfrage
- für Amiga 500 (extern mit Gehäuse) auf Anfrage

Autoboot-Set RLL

- OMTI 5528B, Autoboot-Modul, Festplatten-Interface, Kabelsatz für Amiga 2000 (interne Slotkarte) auf Anfrage
- für Amiga 500 (extern mit Gehäuse) auf Anfrage

Festplattengehäuse

A500/1000 DM 379,—

Amigafarbenes Metallgehäuse, komplett mit Schaltenteil, LEDs, Anschlußadapter mit durchgeführtem Bus, Leds

Autoboot-Filecard DM 299,—

Harddisk-Trägerplatine mit integriertem Autoboot-Modul, Controlleradapter, Autoboot-Software

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!

HK-Computer

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

Telefon: 0221/31 1606 · Telefax: 0221/32 1166
Mo.-Fr. 10.00 - 13.30 u. 14.30 - 18.30, Sa. 10.00 - 14.00 Uhr
BTX: * 22446606 # oder * HK #
Stadtsparkasse Köln, Kto. 634 2133, BLZ 370 501 98

UPS-Versand: Nachnahme + 10,— DM, Vorkasse + 5,— DM
Großgeräte nach UPS-Tabelle ohne Aufschlag, Ausland nur gegen Vorkasse + 15,— DM. Bei Vorkasse nur Eurochecks bis DM 400,— oder Überweisung. Fordern Sie unser kostenloses Info an.
Händleranfragen erwünscht.



Public Domain hoffähig?

Es soll ja tatsächlich noch Amiga-User geben, die noch nie etwas von Public Domain gehört haben. Eigentlich jammerschade, denn sie haben mit Sicherheit etwas verpaßt.

Stellen Sie sich doch einmal folgende durchaus denkbare Konstellation vor: Sie sitzen vor Ihrem Amiga und suchen verzweifelt nach der Lösung Ihres Problems in Gestalt eines Programms. Doch woher nehmen, wenn nicht stehlen. Aber dies wäre ein ganz falscher Gedanke. Wozu gibt es denn die Public Domain – ein Eldorado guter Programme.

Fred Fish, Mister Public Domain, war der Vorreiter schlechthin, indem er zahlreiche Programme zusammentrug und unter dem Label Fish-Disk herausbrachte, das heute weltweit verbreitet ist. Eine Idee, die bereits auf anderen Computersystemen regen Zuspruch bei der Anwenderschaft fand, war nun auch für den Amiga umgesetzt worden; qualitativ gute Programme zu Preisen, die den Geldbeutel nicht unnötig belasten.

Die Software-Entwicklung ist in den vergangenen Jahren mit Siebenmeilenstiefeln vorangeschritten. In gleichem Maße, wie sich kommerzielle Produkte weiterentwickelt haben, steigerte sich auch die Qualität der PD-Programme. Aus der grauen Maus ist ein ernstzunehmender Gegenpart zum Kommerz geworden. Bleibt nur noch anzumerken: Bei der PD stimmt das Preis-Leistungsverhältnis oder mit anderen Worten, hier erhält man wirklich "Value for Money".

Viel Spaß bei der Lektüre wünscht Ihnen, liebe Leser

Vera Brinkmann

Ihre Vera Brinkmann
Redakteurin AMIGA DOS

AMIGA NEWS

Neues vom Amiga-Markt 6

SOFTWARE

Fleißige Helfer 14
Musikalische Erleuchtung 15
Sound-Erstellung leicht gemacht
PagesetterII/TransScript 19
Unter dem Viruscope 22
Erkennung von neuen Viren

HARDWARE

Der Amiga hört aufs Wort 12
Spracherkennung auf dem Amiga
Schnappschuß vom Video 16
Echtzeit-Digitizer im Test
Schneller, höher, weiter 24
AT-Bus-Platte von FSE
Verwandlungskünstler 26
Medusa: Der Atari ST wird simuliert

TIPS & TRICKS

Optimaler Code mit Lattice C 42
Wem die Stunde schlägt 46
Speicherbelegung und Uhrzeit auf einen Blick
Gewußt wie 52
Gefahr erkannt, Gefahr gebannt 54
Fast-File-System besser genutzt
Sector Label 55
Preiswerte Aufrüstung der XT-Karte 57

TITEL

Die Schatzkiste 28
PD-Tophits 32

KURS

Freie Fahrt für Modula2 60
AmigaDos leicht und verständlich 66
Assembler ist keine Hexerei 70

LISTING

EasyCalc 48
Tabellenkalkulation für jedermann



Gespräche führen mit dem Computer? Die Elektronik macht es möglich. Lesen Sie dazu

Seite 12



Unser Listing des Monats: Mit der Jingle-Machine wird der Amiga zum Tonstudio

Seite 78



Medusa: Der Atari ST jetzt als kombinierte Hard- und Software-Lösung im Amiga

Seite 26

LISTING DES MONATS

"Und jetzt, ein Hit-t-t"
Digitalisiertes hörbar gemacht 78

WERKSTATT

Vielseitige Requester für alle Fälle 88

Reflections-Werkstatt
Materialien und Licht 96

Hilfe bei Viren
Gib dem Virus keine Chance 100

PUBLIC DOMAIN

Die PD-Werkzeugkiste 106
Kommerzspiele als Public-
Domain-Demos 108
Public-Domain-Spieleshows 109

SPIELE

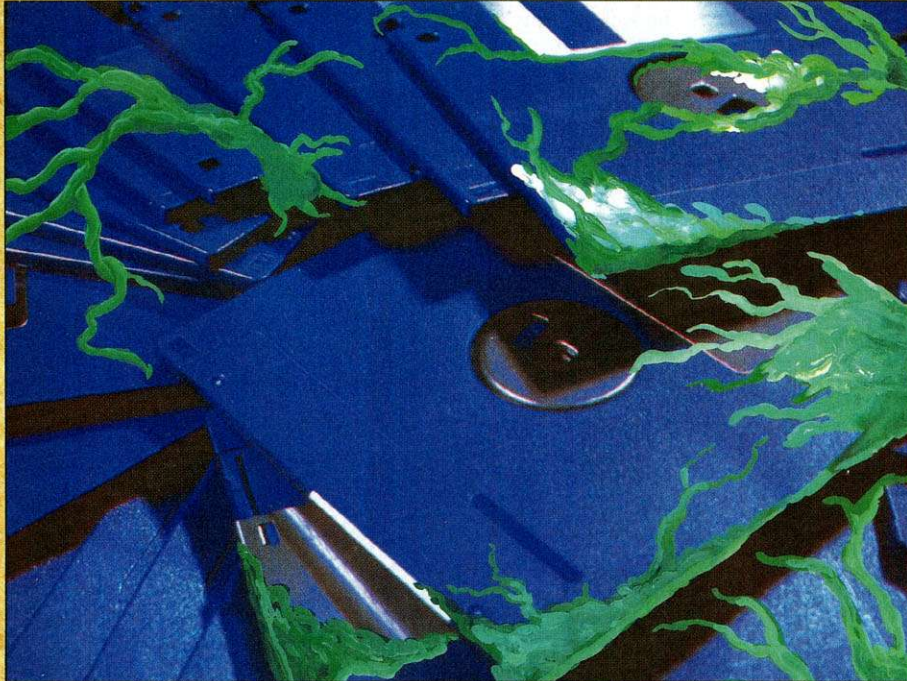
E-Motion 110
Crack-Down 110
Man Hunter 2 111
Astate 111
Heavy Metal 112
Chronoquest II 112
Die Drachen von Laas 113
Lin Wu's Challenge 118
Resolution 101 118
Legend of Fearghail 119
Dark Century 119
Knights of the Crystallion 120
Projectyle 121
Ninja Spirit 121
Escape from the Planet
of the Robot Monsters 122
Treasure Trap 122
Sword of Aragon 123
Champions of Krynn 124
International Championship
Wrestling 125
Grand Prix Circuit 125
All Time Favorites 126
Titano 127

AMIGA-DOS-Spieletips 114

Demnächst auf Ihrem Computer 128

RUBRIKEN

Editorial 3
Leserbefragung 9
Leserbrieft 75
Wettbewerb 103
Bücher 104
Impressum 129
Inserentenverzeichnis 129
Vorschau 130



Schluß mit dem Ungeziefer in Speicher und Bootblock - »ZeroVirus« räumt auf
Seite 100



So mancher Juwel wurde in der Public Domain gefunden. Finden auch Sie Ihren
Schatz. Wir geben wertvolle Hilfen

Seite 28

Richtigstellung

In dem Bericht »Es lebe die Gemütlichkeit« zur Amiga Basel'90 in der Ausgabe 8'90 der AMIGA DOS auf S. 10 wurde die Firma Merkens EDV fälschlicherweise mit der Firma Videocomp in Verbindung gebracht.

Dies entspricht nicht der Wahrheit, und wir bitten dies im Namen der Redaktion zu entschuldigen. Das von der

Firma Merkens EDV vorgestellte Digitalisierungssystem VD 2001 wurde in Zusammenarbeit mit Herrn Peter Reuschling erstellt, der die zum System gehörige Software entwickelt hat.

Info:
Merkens EDV
Fuchstanzstr. 6a
6231 Schwalbach
Tel: 06196/3026



Merkens EDV präsentiert das VD 2001

Flimmerfreier Amiga 2000

Für hohe Ansprüche an die Ergonomie und Bildwiedergabe der Amiga-2000- und -2500-Reihe weist Commodore jetzt auf die Eizo-Flexscan-Monitore 9060-Z und 9070-Z sowie die dazu passende Bildverdopplerkarte X-tension Pro Video von IOAG hin. Die Grafikkarte kostet knapp 2000 DM, die Monitore zirka 2000 DM für 14 Zoll beziehungsweise 2750,- DM für 16 Zoll.

Beide Monitore verfügen über eine antistatische Bildröhrenbeschichtung und reduzierte magnetische Abstrahlung und entsprechen den schwedischen Empfehlungen für Bildschirmarbeitsplätze. Die automatische Synchronisation für Horizontal- und Bildwechselfrequenz deckt die Bereiche 15,5 bis 38,5 kHz/50 bis 90 Hz (9060-Z) und 20 bis 50 kHz/50 bis 80 Hz (9070-Z) ab. Der 14-Zoll-Monitor erreicht eine maximale Auflösung von 820 x 620 Bildpunk-

ten, der 16-Zoll-Bildschirm erreicht maximal 1024 x 768 Punkte.

Die Bildverdopplerkarte X-tension Pro Video für Amigas der 2000-B-Serie bis hin zum 2500/30 belegt keinen Bussteckplatz, sondern ist als Zusatzkarte für den Video-Slot ausgelegt. Sie kann mit einer Bildwechselfrequenz von 60 bis 70 Hz beliebige Multisync-Monitore ansteuern und erlaubt damit eine absolut flimmerfreie Darstellung aller Amiga-Bildformate inklusive HAM-Modus ohne Overscan.

Info:
Grafikkarte:
IOAG
Herr Beck
Bubenreuter Str. 23
8523 Beiersdorf-Igelsdorf
Tel: 09133/5505
Eizo-Monitore:
Rein Elektronik GmbH
Herr Caspers
Lötscher Weg 66
4054 Nettetal 1
Tel: 02153/733-0

Bernoulli-Wechselplatten für Amigas

Bernoulli-Wechselplatten basieren auf einem aerodynamischen Prinzip, das von dem schweizer Mathematiker Daniel Bernoulli im 18. Jahrhundert beschrieben wurde. Bernoulli stellte fest, daß sich der Druck einer Flüssigkeit beziehungsweise eines Gases umgekehrt proportional zur Strömungsgeschwindigkeit verhält. Nach diesem Prinzip werden die Cartridges stabilisiert. Die Bernoulli Drive Cartridges stellen somit ein höchst zuverlässiges, wechselfähiges Speichermedium.

Die Firma Jörg Grenz Computersysteme bietet solche, bereits vom Apple Macintosh und IBM PC/PS-2 bekannte Wechselplatten an. Lieferbar sind diese als 20-MByte- und 44-MByte-Versionen. Für den Amiga 500 ist die Platte mit

bis zu 4 MByte zusätzlichem Fast-RAM erhältlich. Eine externe oder interne Version wird für den Amiga 2000 angeboten, hier sind sogar bis zu 8 MByte Fast-RAM möglich. Auch der Amiga 2500 und 3000 können mit solchen Harddisks ausgerüstet werden. Als Preisbeispiel wurde eine Konfiguration für den Amiga 2000 genannt, mit einem 44-MByte-Drive, SCSI-Controller und 2 MByte Fast-RAM. Diese Zusammenstellung kostet 4500,- DM. Die Zugriffszeit der Platten liegt zwischen 17 ms (20 MByte) und 22 ms (44 MByte).

Info:
Jörg Grenz Computersysteme
Holtener Str. 67
2300 Kiel 1
Tel: 0431/569337
Privat: 0431/567721

Cyberpunk-Abenteuer jetzt auch auf dem Amiga!

Electronic Arts kündigt für Juli das Erscheinen eines seit langem von Fans erwarteten Rollenspiels an: »Neuromancer«. Das auf dem inzwischen zum Kultbuch avancierten Science-fiction-Roman von W. Gibson basierende Spiel handelt von einer Zukunft, in der Mikroelektronik, Biotechnik und Chemie nahezu alle wirtschaftlichen und privaten Bereiche kontrollieren. Computereffreaks mit chirurgisch implantierter Elektronik, die sogenannten Cyberpunks, treiben sich in einem riesigen Computernetzwerk, dem Cyberspace herum, um gegen Bezahlung den Datenschutz fremder Firmen zu knacken. Im Japan des einundzwanzigsten Jahrhunderts streift der Spieler durch die Straßen, auf

der Suche nach besserer Ausrüstung und Informationen. Nahezu jede Information kann über das weltweite Kommunikationsnetz abgerufen werden, was aber unter Umständen tödlich sein kann. Es gehört schon einiges dazu, das EIS fremder Software zu überlisten. Wer die bessere Ausrüstung, die besseren Informationen und die bessere Software hat, gewinnt.

Info:
Electronic Arts
Miss Sara Shrapnell
Langley Business Centre
11-49 Station Road
Langley
Nr. Slough
Berkshire SL3 8YN
Great Britain
Tel: 0044-753-49442
Fax: 0044-753-46672

16-Bit-Sampler für A2000

Von der Firma OMEGA Datentechnik wird ab Herbst/Winter 1990 ein 16-Bit-Sampler in Form einer Steckkarte für den Amiga 2000 und den Amiga 3000 erhältlich sein. Mit der passenden Steuer-Software sollen Echtheiteffekte, aktive Höhen- und Tiefenregelung und Echoeffekte möglich sein. Die Samplingrate ist von 20 kHz bis 55 kHz einstellbar und verfügt über einen Rauschab-

stand von mehr als 72 dB/a. Somit lassen sich auf einem Amiga 2000 bis zu 90 Sekunden, auf dem Amiga 3000 bis zu 270 Sekunden Klang aufnehmen. Der Preis soll unter 2000,- DM liegen.

Info:
OMEGA Datentechnik
Quellenweg 20
2900 Oldenburg
Tel: 0441/71109

Core Design auf eigenen Füßen

Das britische Software-Label Core Design Limited, bereits bekannt durch Spiele wie »Rick Dangerous« und »Impossible«, vertreibt seine Produkte ab sofort selbst. Den Einstieg stellen zwei Spiele dar, von denen uns leider nur Bildschirmfotos vorlagen, die wir Ihnen aber trotzdem nicht vorenthalten wollen.

In »Corporation« muß sich der Spieler in der Gestalt eines Top-Agenten in eine Firma einschleichen, um dort den Beweis für illegale Gen-Experimente an ultimativen Killerwesen zu sichern. Dazu ist er mit Waffen, Gasmaske, Infrarot-Sichtgerät und ähnlichem ausgerüstet. Viele Gegner in Form von Menschen, Wach-Robotern und gentechnischen Kreaturen müssen ausgeschaltet werden, bis das Ziel, ein Embryo des neuesten »Produkts«, ans Tageslicht gebracht werden kann.

Ein klassisches Hüpf- und Prügelspiel stellt »Torvak – the Warrior« dar. Der nimmermüde Held, diesmal als schwertschwingender Krieger verkleidet, rennt und schlägt sich durch etliche Levels, bis es ihm endlich gelingt, dem bösen Necromancer, der das Land mit dunklen Heeren zu überschwemmen droht, den Garaus zu machen. Von der Spielidee und der Hintergrund-Story nun nicht gerade überwältigend, kann Torvak jedoch mit hübsch bunter Grafik aufwarten. Man darf auf jeden Fall gespannt sein.

Info:
CORE Design Limited
Suite C
Tradewinds House
69/71A Ashbourne Road
Derby. DE3 3FS.
Great Britain
Tel: 0044-332-297797
Fax: 0044-332-381511



... zeigt sich »Torvak« eher barbarisch!

Comic-Spiel mit AMOS

Das erste mit AMOS erstellte kommerzielle Spiel erscheint jetzt bei Mandarin Software. AMOS ist eine BASIC-ähnliche Programmiersprache mit über 500 Befehlen, die es ermöglicht, ohne großen Aufwand schnelle und professionelle Software zu schreiben. Wir werden Ihnen AMOS in einer der nächsten Ausgaben vorstellen.

»Carton Capers starring Judo Jake and Karate Kat« ist ein Comic-Karatespiel, das den von Mandarin Software gestifteten STOS-Preis von 5000,- US\$ gewann. Das Spiel besitzt eine Zwei-Spieler-Option, humor-

volle Grafiken und fließende Animationen der Figuren. Neben den üblichen Prügeltätlichkeiten können auch Bomben, fallende Safes und Falltüren benutzt werden, um sich den Gegenspieler vom Leib zu halten. Carton Capers zeigt sehr gut die Fähigkeiten, die AMOS zu bieten hat.

Info:
Mandarin Software
Europa House
Adlington Park
Adlington
Macclesfield SK 10 4NP
Great Britain
Tel: 0044-625-859333
Fax: 0044-625-879961



Während sich »Corporation« mit High-Tech befaßt...

Amiga-Programme auf MS-DOS-Rechnern entwickeln

Lattice bietet einen C-Cross-Compiler an, der es ermöglicht, Amiga-Programme auf MS-DOS-Rechnern zu entwickeln. Dies vereint die Vorzüge des MS-DOS-Systems, wie zum Beispiel Netzwerke, preiswerte Backup-Medien und schnellere Kompilierzeiten, mit der Leistungsfähigkeit der Amiga-Bibliotheken. Dabei wurden alle von Lattice bekannten Funktionen, wie C-Schlüsselwörter und Komman-

dozeilenoptionen, vollständig übertragen. Die neueste Version 5.0 dieses Lattice-C-Compilers kostet je nach Ausführung zwischen 350,- und 750,- US\$.

Info:
Lattice, Incorporated
2500 South Highland Avenue
Lombard, IL 60148
USA
Tel: 001-708-916-1600
Fax: 001-708-916-1190

Strukturierte Software-Entwicklung leicht gemacht

Die Abschottung der DDR in den letzten Jahrzehnten hat dort zu einem Informationsdefizit geführt, das nun in einer gewaltigen Anstrengung aufgeholt werden muß, wenn dieser Staat existenzfähig bleiben will. Die Bundesrepublik Deutschland hat nur eine Chance, ihre Waren dort abzusetzen, wenn wir helfen, unsere westlichen Technologien zu verstehen und sich anzueignen.

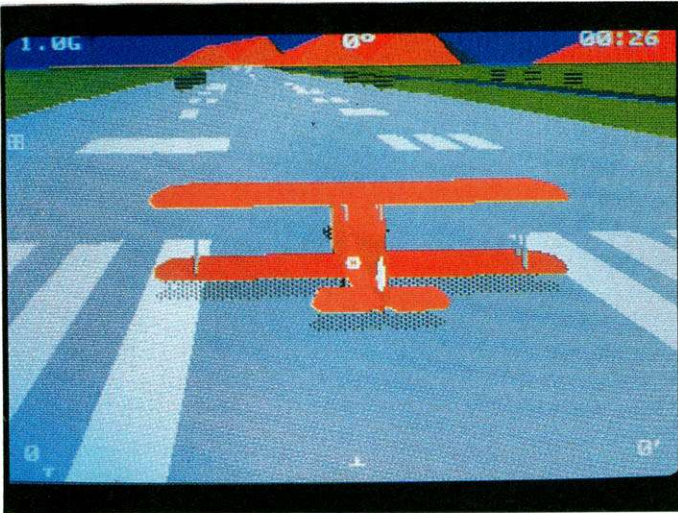
Dies bedeutet, daß auch für uns »alte Themen« wieder neu aufbereitet werden müssen. Die »Flußdiagrammtechniken für strukturiertes Programmieren« nach I. Nassi und B. Shneiderman sind ein solches »altes Thema«.

Die Firma Structo.Soft hat es sich zur Aufgabe gesetzt, diese Hilfeleistung verständlich zu machen, und bietet für diverse Rechnersysteme Struktogrammgeneratoren an, die aus den Quellcodes verschiede-

ner Programmiersprachen (ALGOL, C, dBase, Modula, Pascal, PLM und PL/1) die notwendigen Struktogramme erzeugt. Die Generatoren lassen sich an unterschiedliche Dialekte anpassen, sind schnell und gestatten die Auswahl von Struktogrammausschnitten für die Einzeldarstellung. Bestimmte Befehle oder Kommentare können wahlweise bei der Ausgabe, die auf Drucker oder in eine Datei erfolgt, unterdrückt werden.

Für die Benutzer der Struktogrammgeneratoren stellt Structo.Soft eine Hotline sowie einen Update-Service zur Verfügung. Der Preis beläuft sich auf 299,- DM zuzüglich Versandkosten.

Info:
Structo.Soft Peter Ax
Simthäuser Straße 16
D-3552 Wetter/Hessen
Tel: 06423-7197



»C. Yeagers AFT«: Noch ist der Vogel auf der Startbahn

Fliegen wie ein Profi

»Chuck Yeagers Advanced Flight Trainer« war bisher ein Klassiker auf dem PC und einigen 8-Bit-Rechnern. Nun kommen auch Amiga-Besitzer in den Genuß der perfekten Flugsimulation und das gleich in einer verbesserten Version 2.0. Electronic Arts kündigte für Juli 1990 das Erscheinen des AFT 2.0 an, der zur Vorgängerversion einige neue Features bekommen hat. So erwarten die passionierten Piloten mehrere Flugzeugtypen, ein völlig neues Terrain, eine Audio-Kassette von Chuck Yeager persönlich und eine bildschirmgesteuerte, sechstägige Flugschule. Alle 18 wählbaren Flugzeugtypen

werden mit einer individuellen 3D-Grafik dargestellt und erstrecken sich vom Doppeldecker bis zum Kunstflug-Jet. Auch Formationen können geflogen werden, dabei übernimmt der Amiga die Steuerung der anderen beteiligten Flugzeuge.

Info:
Electronic Arts
Miss Sara Shrapnell
Langley Business Centre
11-49 Station Road
Langley
Nr. Slough
Berkshire SL3 8YN
Great Britain
Tel: 0044-753-49442
Fax: 0044-753-46672



»C. Yeagers AFT«: Exzellente 3D-Grafik

MW-500

Um Mißverständnissen bei der Leserschaft vorzubeugen. Der Preis von 298,- DM für das MW-500 beinhaltet natür-

lich nicht das Zweitlaufwerk und die Zusatzplatine. Diese müssen selbstredend extra bezahlt werden.

Messen und Ausstellungen

Zum vierten Mal findet in Karlsruhe der »Tag des Computers« statt. Wie in den Vorjahren bieten die Initiatoren, die Veranstaltergemeinschaft Bruchsaler Computerclubs, eine Auswahl an Hardware und Software für alle Systeme und alle Bedürfnisse an. Für Bastler und Flohmarktjäger wird ein Teil der Halle reserviert. Tischreservierungen zum Preis von 15,- DM werden unter der unten angegebenen Adresse angenommen. Die Veranstaltung findet am 13.10.1990 von 9 Uhr bis 18 Uhr statt. Der Ausstellungsort, die Bruchbühlhalle in 7528 Karlsruhe-Neuthard, ist über die Bundesautobahn A5 erreichbar und ab der Abfahrt Bruchsal ausgeschildert.

Info:
Papas Computerclub
Postfach 4309
7520 Bruchsal
Tel: 07255-4326
Fax: 07255-9846

Die Münchner Computertage 90 für Handel, Heim und Handwerk finden vom 1.9. bis 3.9. im Deutschen Museum in München unter der Schirmherrschaft des DMV-Verlages statt. Diese Ausstellung ist keine Herstellermesse, sondern bietet Software-Entwicklern und Computerfachhändlern die Gelegenheit, ihr Gesamtprogramm vorzustellen. Erfahrungsgemäß werden

freiberuflich Tätige, Entscheidungsträger aus Klein- und Mittelbetrieben sowie der Privatanwender den Weg ins Deutsche Museum finden.

Info:
Münchner Werbe- und Messeagentur
Jägerweg 31
8031 Gilching/München

Das Amiga Meeting 90 findet vom 25.10.90 bis 28.10.90 in Saarbrücken und vom 18.11.90 bis 21.11.90 in Wiesbaden statt. Auf beiden Ausstellungen erwarten den Besucher Präsentationen, Workshops, Seminare und Kommunikation. Neben Shows und Anwendungen werden auch Verkaufsgeschäfte getätigt, wobei wieder günstige Messekonditionen angeboten werden. Jeweils der erste Messtag ist für Fachpublikum vorgesehen. Amiga-Entwickler werden in Seminaren und Workshops über die neusten Techniken informieren.

Als Veranstaltungsorte wurden in Wiesbaden die Rhein-Main-Halle (Eingang Diana-brunnen) und in Saarbrücken das Messegelände gewählt.

Info:
Teamwork Messen und Ausstellungen
Mainzerstraße 36
6501 Schwabenheim
Tel: 06130-6423
Fax: 06130-6634

2 MByte unter 500,- DM

Erstmals liegt der Preis für eine 2-MByte-Erweiterung unter 500,- DM. Die Firma Roßmüller Computertechnik bietet eine neue Speichererweiterung für den Amiga 500 an, die inklusive Uhr 498,- DM kosten soll. Die Erweiterung läuft sowohl mit dem neuen Big AGNUS als auch mit dem alten Fat AGNUS. Mit Big AGNUS beträgt der verfügbare Speicher 1 MByte Chip-RAM und 1,5 MByte Fast-RAM. Der alte Fat AGNUS erhält durch diese Karte 512 kByte Chip-RAM und 1,8 MByte Fast-RAM. Bei beiden Versionen muß die CPU 68000 nicht gesockelt werden, was bei älteren Speichererweiterungen der Fall war. Diese Tatsache erlaubt auch den Einsatz von Beschleunigerkarten. Preissenkungen gab es auch bei weiteren Produkten von

Roßmüller. So wird die 2-MByte-Speichererweiterung für den Amiga 2000, die auf bis zu 8 MByte aufrüstbar ist, in Zukunft 598,- DM kosten. Eine interne Speichererweiterung für den Amiga 1000, die mit 2 MByte ausgeliefert wird und auf 8 MByte erweiterbar ist, soll ebenfalls für 598,- DM erhältlich sein.

Eine Beschleunigerkarte, die Tornado für die Amiga 500, 1000 und 2000, ist von Roßmüller für 598,- DM erhältlich. Durch die Karte wird der Amiga zirka zweimal schneller. Einen Test dieser Karte präsentieren wir in einer der nächsten Ausgaben.

Info:
Roßmüller Handshake GmbH
Neuer Markt 21
5309 Meckenheim
Tel: 02225-2061/62/63
Fax: 02225-10193

Haben Sie ein paar Minuten Zeit für uns?

Die Amiga DOS hat schon viele Freunde gewonnen. Wir wollen die AmigaDOS für Sie aber so interessant wie nur möglich gestalten. Dazu müssen wir von Ihnen wissen, wie Sie Ihren Amiga bevorzugt nutzen, um Sie in Zukunft noch besser informieren zu können. Um Ihnen das Ausfüllen des Fragebogens zu versüßen, haben wir einige interessante Preise zusammengestellt.

Nicht vergessen:

Einsendeschluß ist der 12.09.1990 (Poststempel)

So können Sie AMIGA DOS mitgestalten:

Sie brauchen lediglich die gestellten Fragen durch Einkreisen des betreffenden Buchstabens zu beantworten. Natürlich interessieren uns auch Ihre Ideen, die Sie in den liniierten Feldern niederschreiben können (pro Person bitte nur einen Fragebogen). Um einen der tollen Preise zu gewinnen, senden Sie einfach den ausgefüllten Bogen an:

DMV-Verlag

Redaktion AMIGA DOS

Postfach 250

3440 Eschwege

Der Rechtsweg ist ausgeschlossen. Mitarbeiter des DMV-Verlages und des Redaktionsbüros Everts & Hagedorn sowie deren Angehörige dürfen nicht teilnehmen.



Unter allen Einsendern verlosen wir:

- 1 Drucker Epson LQ-400
- 3 Zweitlaufwerke für den Amiga
- 10 Abonnements der AMIGA DOS
- 10 Databoxen dieser Ausgabe
- 10 Sampler mit den besten Programmen der AMIGA DOS

AMIGA-DOS Leserbefragung 09/90

1. Welchen Amiga Rechner benutzen Sie?
 - a Amiga 500
 - b Amiga 1000
 - c Amiga 2000
 - d Amiga 2500
 - e Amiga 3000
2. Welchen Rechner hatten Sie vor Ihrem Amiga in Betrieb?
 - a Keinen
 - b C-64
 - c Commodore VC20, C16
 - d Commodore 128
 - e Sinclair
 - f Schneider CPC
 - g Atari
 - h Texas Instruments
 - i Andere
3. Welche Zusatz-Hardware betreiben Sie an Ihrem Amiga?

a Zweitlaufwerk	Typ:
b Harddisc	Typ:
c Drucker	Typ:
d Speichererweiterung	Typ:
e PC-Karte	Typ:
f Turboboard	Typ:
g Sonstiges	Typ:
4. In welcher Sprache programmieren Sie?
 - a AmigaBASIC
 - b GFA-BASIC
 - c Assembler
 - d Modula2
 - e C
 - f Pascal
 - g Sonstige: _____
 - h Überhaupt nicht
5. Wie würden Sie Ihre Programmierkenntnisse einstufen?
 - a Einsteiger
 - b Fortgeschrittener
 - c Professionell
6. Wie würden Sie die einzelnen Rubriken in der AMIGA DOS und Ihre Aufmachung benoten? (Noten von 1 - 6)

a Amiga News	_____
b Software-Test	_____
c Hardware-Test	_____
d Tips & Tricks	_____
e Kurs	_____
f Listing	_____
g Werkstatt	_____
h Public Domain	_____
i Spiele	_____
j Rubriken	_____
k Titelbild	_____
l Umfang	_____
m Gliederung	_____
n Aufmachung	_____
o Gesamteindruck	_____
7. Welcher Artikel hat Ihnen am besten in diesem Heft gefallen?
 - a _____
 - b _____
 - c _____
8. Welcher Artikel hat Ihnen überhaupt nicht gefallen?
 - a _____
9. Warum hat Ihnen dieser Artikel nicht gefallen?
 - a _____
 - b _____
10. Über welche Themen würden Sie gerne mehr in der AMIGA DOS lesen?
 - _____
 - _____

Leserbefragung

11. Welche Programme tippen Sie bevorzugt ab?

- a _____
- b _____

12. Wo liegt Ihr persönlicher Schwerpunkt bei der Arbeit mit dem Amiga?

- a Textverarbeitung
- b Programmierung
- c Spiele
- d Datenbank
- e Grafik/Animation
- f Musik
- g andere _____

13. Welche Themen sind ausreichend behandelt?

- a Computerbasteleien
- b Programmierung/Listing
- c Kurse, Thema:
- d Software-Tests
- e Hardware-Tests
- f Tips & Tricks
- g Workshops
- h Unterhaltung
- i Public Domain / Shareware
- k Sonstige Bereiche:

14. Welche Themen sind für Sie unwichtig?

- a Computerbasteleien
- b Programmierung/Listing
- c Kurse, Thema:
- d Software-Tests
- e Hardware-Tests
- f Tips & Tricks
- g Workshops
- h Unterhaltung
- i Public Domain / Shareware
- k Sonstige Bereiche:

15. Die AMIGA DOS sollte über folgende Themen ausführlicher berichten:

- a Computerbasteleien
- b Programmierung/Listing
- c Kurse, Thema:
- d Software-Tests
- e Hardware-Tests
- f Tips & Tricks
- g Workshops
- h Unterhaltung
- i Public Domain / Shareware
- k Sonstige Bereiche:

16. Kaufen Sie Software und Zubehör aufgrund

- a Tests in Zeitschriften?
- b Anzeigen?
- c private Empfehlungen?
- d Händlerberatung?

17. Wo würden Sie die "Schmerzgrenze" bei der Anschaffung von Soft- und Hardware setzen?

- a Software: DM:
- b Hardware: DM:

18. Wie haben Sie AMIGA DOS kennengelernt?

- a Beim Zeitschriftenhändler
- b durch Anzeigen
- c Durch Bekannte
- d Beim Computerhändler

19. Sammeln Sie die AMIGA DOS?

- a Ja
- b Nein

20. Wie lange lesen Sie durchschnittlich an einer Ausgabe der AMIGA DOS?

_____ h, _____ Min

21. Welcher Berufsgruppe gehören Sie an?

- a Schüler/Auszubildende
- b Studenten
- c Berufstätige
- d Selbständige
- e Sonstige

22. Welche Schulbildung haben Sie?

- a Hauptschule
- b Hauptschule und abgeschlossene Berufsausbildung
- c Lehre
- d Abitur
- e Hochschule

23. Welche Freizeitinteressen haben Sie?

- a Hifi
- b Video
- c Literatur
- d Urlaub/Reisen
- e Essen und Trinken
- f Foto
- g Auto
- h Ski
- i Sonstige _____

24. Welcher Altersgruppe gehören Sie an?

- a unter 15
- b 15 bis 18
- c 19 bis 25
- d 26 bis 30
- e 31 bis 40
- f über 40

25. Welcher Gehaltsgruppe gehören Sie an?

- a kein eigenes Einkommen
- b bis 1000,- DM Netto
- c bis 2000,- DM Netto
- d bis 3000,- DM Netto
- e bis 4000,- DM Netto
- f bis 5000,- DM Netto

26. Welche Investitionen wollen Sie in nächster Zeit tätigen?

- a Floppy-Disc
- b Festplatte
- c Drucker
- d Monitor
- e Sonstige _____

27. Welche Zeitschriften lesen Sie außer AMIGA DOS? (Mehrfachnennungen möglich)

- a Amiga
- b Kickstart
- c Amiga Power
- d Amiga Aktiv
- e Amiga World
- f Amiga Welt
- i Toolbox
- k 68000
- m Sonstige:

28. Welche Zeitschriften haben Sie abonniert?

- a _____
- b _____
- c _____
- d _____

29. Wie kommen Sie in den Genuß der veröffentlichten Programme?

- a abtippen
- b Databox

30. Wie hat Ihnen die Aufmachung der Databox gefallen?

- a gut
- b durchschnittlich
- c nicht so gut
- d Verbesserungsvorschläge:

Vorname: _____

Name: _____

Straße: _____

PLZ/Ort: _____

Wir versichern, daß die hier gemachten Angaben vertraulich behandelt werden.

GFA für AMIGA

GFA-BASIC 3.5 Interpreter Amiga

Weiterentwicklung des GFA-BASIC 3.0 Interpreter mit 35 zusätzlichen Befehlen aus der linearen Algebra und Kombinatorik. Außerdem verbesserte Editor-Eigenschaften (Funktionen falten und Suche in Kopfzeilen gefalteter Funktionen bzw. Prozeduren)

DM 228,- *neu*

GFA-BASIC 3.5 Compiler

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 139,- *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,-



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,-**

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,-

**HÄNDLERANFRAGEN
ERWÜNSCHT!**
Bitte fordern Sie unser
Händler-Info an!
Tel.: 0211 - 55 04-0

GFA Systemtechnik GmbH
Heerdter Sandberg 30
D-4000 Düsseldorf 11
Tel. 0211/55 04-0 Fax 0211/55 04 44





Der Amiga hört aufs Wort

Sprachliche Kommandoeingabe mit »VorecOne«

Ein Amiga, der versteht, was der Benutzer von ihm möchte – das wäre eine feine Sache. Genau das wurde ja schon im Editorial der AMIGA DOS 7/90 angesprochen. Daß sich solche Wünsche realisieren lassen, stand allerdings damals noch in den Sternen. Wie schnell sich das Blatt wenden kann, zeigt eine neue Art der Kommandoeingabe – »VorecOne«.

»VorecOne« kommt als "Blackbox", als schwarzer, nicht identifizierbarer Kasten daher, der auf der einen Seite in einem neunpoligen Sub-D-Stecker endet und auf der Oberseite eine Klinkenbuchse besitzt. Dazu findet man eines der handelsüblichen Kleinmikrofone, wie Sie auch bei preiswerten Kassettenrecordern zu finden sind. Richtig geraten – der Stecker des Mikros gehört in die Klinkenbuchse der "Blackbox". Diese wiederum wird am Joystickport des Amiga angeschlossen.

Stellen Sie sich vor, sie sagen "LIST" – und der Amiga hört zu

Nachdem das ganze elektrisch miteinander verbunden wurde (während der Rechner aus-

geschaltet war, versteht sich von selbst), startet man nun die Software von »VorecOne« – und steht im ersten Moment ratlos da. Das Mikrofon in die Hand genommen und geredet, was das Zeug hält, ist ja ganz gut und schön, nur leider reagiert der Amiga darauf überhaupt nicht.

Des Rätsels Lösung ist einfach. Bevor der Computer was sagt, müssen wir was sagen, nämlich "was Sache ist". Im Klartext, »VorecOne« ist ein Editor, der eine eigene Computersprache verwaltet. Diese Computersprache ist dem BASIC recht ähnlich, Einsteiger kommen also auch gleich mit dem System klar, ohne noch großartig dazulernen zu müssen.

Diese Sprachelemente müssen sinngebend zusammengestellt werden, damit der Computer vom Benutzer gespro-

chene Befehle ausführen kann. Der Editor teilt sich in zwei Bereiche auf, zum einen in einen Spracherkennungsteil, zum anderen in den Ausführungsteil. Im Spracherkennungsteil werden die Wörter eingetragen, die später per Mikrofon dem Computer übergeben werden. Diese Wörter müssen nämlich erst einmal gesprochen werden, damit der Amiga sie überhaupt erkennt. Zu diesem Zweck findet man in der Menüleiste des Programms »Voice«, so heißt der Editor, die Funktion »Train«. Bei der Anwahl von »Train« muß jedes Wort, das im Spracherkennungsteil steht, dem Computer zweimal vorgesprochen werden (bei fehlender Erkennung auch manchmal öfter).

Diese Stimmerkennung ist wichtig, wird das Wort von einer anderen Person gespro-

chen, ist eine Erkennung kaum möglich. »VorecOne« braucht eine klare Aussprache, um seine Arbeit zu tun. Übrigens, das Programm ist komplett in Englisch gehalten und vielleicht nicht gleich ohne Verständnisprobleme zu bewältigen, man sollte allerdings bei entstandenen Fehlern nicht gleich das Mikrofon ins Korn werfen. »VorecOne« ist kein Programm für das Fünfminuten-Lernziel, es bedarf eines intensiven Trainings, um es effektiv nutzen zu können. Zum Glück waren sich die Hersteller nicht zu schade, dem Käufer ein ausführliches Handbuch (allerdings ebenfalls in Englisch) und diverse Demo-Dateien mit auf den Weg zu geben. Anhand dieser Demo-Dateien sollte man sich erst mal mit der Spracherkennung vertraut machen, die manchmal auch ihre Tücken hat, doch dazu später mehr.

Die Reihenfolge: sprechen, erkennen, ausführen – meistens jedenfalls

Kommen wir zurück zum Editor. Während auf der linken Seite die Kommandos, die vom Benutzer gesprochen über das Mikro in den Rechner gelangen, quasi als Vergleichstabelle stehen, sind rechts die eigentlichen Kommandos einzusetzen, welche die entsprechende Aktion ausführen, die durch das Sprachkommando erfolgen soll. »VorecOne« kennt dabei eine Menge an Kommandos oder besser Befehle, mit denen ein ganzes Programm geschrieben werden kann. Dabei sind Grafikoperationen und die Einbindung des Systems ebenso gewährleistet wie bedingte Verzweigungen mit »IF« und »ELSE«.

Zwei Grafikscreens werden von »VorecOne« verwaltet, zwischen beiden kann während des Programmablaufs hin- und hergeschaltet werden. Zu diesen Screens gehören bestimmte Grafikbefehle, die durch die Sprachkommandos abgerufen werden können. So können einfache Grafiken unter Zuhilfenahme von Kreisen, Rechtecken oder Linien gezeichnet werden, wobei die Parameter ähnlich denen unter AmigaBASIC sind. Durch das Sprechen des Kommandowortes wird dann



Bild 1. Das ist »VorecOne«: Ein Analog-Digitalwandler, ein Mikrofon und das Programm

die jeweilige Grafikroutine abgearbeitet. Mit den Kommandos läßt sich zum Beispiel zwischen beiden Grafikschränken hin- und herschalten. Für Präsentationszwecke könnte man sich »VorecOne« als Hilfsprogramm schon mal gut vorstellen.

Spracheingabe zwischen Wollen und Können

Wenn man »VorecOne« beurteilen will, muß man mehrere Faktoren beachten:

1.) Spracheingabe-Systeme sind auf dem Amiga – oder besser auf Computern – noch ziemlich unbekannt, wenn man in erster Linie den Privatbereich sieht. Meistens sind sie dazu recht teuer und für den »Otto Normalamiganer« nicht unbedingt lebensnotwendig. Der Preis von »VorecOne« stand leider zu Redaktionsschluß noch nicht fest, eine Kosten-Nutzenanalyse kann daher nur hypotheti-

schen Wert haben. Schaut man sich das Handbuch zu »VorecOne« an, so fällt einem die Stelle ins Auge, wo gesagt wird, daß die Spracherkennung zu 98% funktioniert. Bei unseren Tests mußten wir deutliche Abstriche machen, prozentual kommen wir nur auf einen Wert von 75%. Aber auch das ist für Spracheingaben schon recht hoch.

»VorecOne« ist also kein Blender, der mehr verspricht, als er hält. Als mildernde Umstände muß man auch immer wieder darauf hinweisen, daß englischsprachige Software (hier im wahrsten Sinn des Wortes) in deutschen Ländern immer ein bißchen mit Schwierigkeiten kämpfen muß. Unsere Versuche zeigten, daß Wörter mit annähernd gleichem Klang (»good«, »blue«, »load«) mit fünfzigprozentiger Sicherheit falsch verstanden wurden. Das Abfangen von Programmierfehlern ist ebenfalls nicht gewährleistet, eine falsche Pfadangabe, und schon ist es aus mit dem gegenseitigen Verständnis. Man kommt

weder in den Editor zurück, noch läßt sich die Spracheingabe deaktivieren. Das Ergebnis? Die Anwendung des berühmten Dreifingergriffs wird zur letzten Kommunikation zwischen Mensch und Maschine. Das waren die sogenannten Negativpunkte. Kommen wir jetzt zu den positiven.

Dafür, daß Spracheingaben noch Neuland sind, macht »VorecOne« eine gute Figur. Durch die Implementation einer eigenen Computersprache ist der Anwender nicht gezwungen, sich an starre Kommandofolgen zu gewöhnen, sondern kann sich seine Spracheingabe-Umgebung selbst zusammenstellen. Durch die Vielzahl der Funktionen und Befehle, die sogar Grafik und Sound zulassen, ist »VorecOne« eines der interessantesten Eingabemedien neben Maus, Joystick und Tastatur. Für wen und welchen Geldbeutel, muß jeder selbst entscheiden, der Verwendungszweck ist ebenfalls selbst zu finden.

(jb)

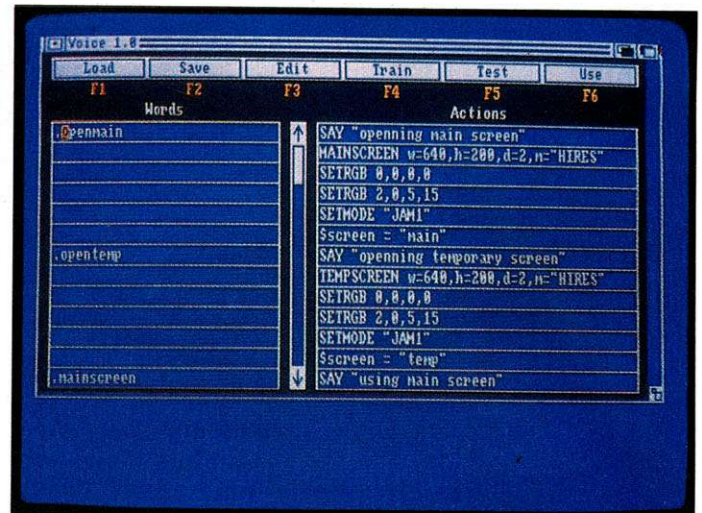


Bild 2. Der Editor von »VorecOne«. Hier werden die Kommandozeilen eingegeben

AMIGA DOS Blitzlicht

Name: VorecOne
Hersteller: Impulse
Anbieter: Pulsar
Funktion: Spracheingabe-System
Preis: 328,- DM

Positiv:

- leicht erlernbar
- schnelle Installation
- nicht kopiergeschützte Software, direkt auf Festplatte übertragbar
- nützliche Hilfsprogramme und hilfreiche Demos
- Sonderpunkt für originelle Idee

Negativ:

- nicht immer treffsicher, vor allem bei gleichlautenden Wörtern
- Anschlußkabel der Wandlerbox recht kurz
- bei falschen Verzeichnispfaden »Reise ohne Wiederkehr«

CD , CIRCLE , CKEY , CLOSE , CLOSEMAIN , CLOSETEMP , DEACTIVATE ,
DELAY , DRAW , ELLIPSE , ELSE , ELSEIF , ENDIF , EXIT , FKEY ,
FUNCTIONS , GADGETDOWN , GADGETUP , GFXMAIN , GFXTEMP , IF ,
KEYSEQ , LABEL , LKEY , LDOWN , LUP , LOADPIC , LOADBRUSH ,
LOADPALETTE , LOADWORDS , MAINSCREEN , MAINTOBACK , MAINTOFRONT ,
MENU , MOUSE , MOVE , NEWWORDS , POP , POPTO , RELMOUSE , RKEY ,
RAWKEY , RECON , RECTANGLE , RUP , RDOWN , REXX , SAY , SCROLL ,
SETSPPEECH , SETAPEN , SETBPEN , SETRGB , SETFONT , SETMODE , SOUND ,
TEMPSCREEN , TEMPTOBACK , TEMPTOFRONT , TEXT , TITLEOFF , TITLEON ,
WAITMOUSE , \$EXTRACT , \$FILETYPE , \$INPUT , \$LENGTH , \$SLOWER ,
\$MOUSEX , \$MOUSEY , \$POSITION , \$REPLACE , \$REXXFN , \$BSTRING ,
\$UPPER , \$VALUE

Abb. 1. Der Sprachschatz von »VorecOne«. Die Befehls- und Funktionsvielfalt erlaubt komplette Programme

Der CLI ist, was Optionen im Umgang mit Disketten anbelangt, in vieler Hinsicht nicht ausreichend. Daher werden immer wieder Programme veröffentlicht, die das Arbeiten mit Daten von externen beziehungsweise internen Laufwerken schöner gestalten.

Wir haben die Programmsammlung Disk-Utilities Nr. 2 etwas genauer unter die Lupe genommen. Auf der Diskette befinden sich insgesamt fünf Programme, die sich mit der Diskettenbehandlung beschäftigen: »Disk Ed V6«, »Select Copy«, »DCopy II«, »Check« und »Boot Girl Plus«. Das Programm »DiskEd« erlaubt das Einlesen und Modifizieren von Disketten. Der Diskettenmonitor besitzt ein Eingabefeld, das nahezu den ganzen Bildschirm einnimmt. Nur am unteren Rand sind die Gadgets zur Steuerung des Programms in einer Menüleiste angebracht. Diese Optionen erreicht man durch Anklicken per Maus. Hier wird auch das aktuelle Laufwerk gewählt, das von DF0: bis DF3: reichen kann. Nachdem der Sektor, der Zylinder und der Head definiert worden sind, liest das Programm den gewählten Bereich der Diskette aus und zeigt die Daten im Hex-Modus in dem Editorfeld an. Hier können Sie den Text sowohl in Hexadezimal- als auch im ASCII-Format eingeben. Mit der Write-Option wird der Sektor abgespeichert.

Eine besondere Option stellt der Map-Befehl dar. Nach Aktivierung dieses Menüpunktes bauen sich zwei Felder auf, die Kästchen für 80 Zylinder x 11 Sektoren beinhalten. Belegte Sektoren sind schwarz und freie Sektoren braun gekennzeichnet, so daß man die Möglichkeit hat, die jeweils eingelegte Diskette einfacher zu analysieren. Der Computer errechnet dann die freien und belegten Sektoren.

Bei dem Programm »Check« handelt es sich um ein kleines, jedoch recht nützliches Helferlein. »Check« vergleicht nach einem Kopiervorgang das Source-File und das Destination-File. Dies ist vor allem dann nötig, wenn in komplexeren Programmen, die auf Unterprogramme zurückgreifen, unerwartet Fehler auftreten. Nach dem Start wird das aktuelle Verzeichnis der in dem Laufwerk DF0: befindlichen Diskette aufgelistet.

Jürgen Seibel

Fleißige Helfer Disk-Utilities Nr. 2

Wer wünscht sich nicht einen unkomplizierten Umgang mit seinen Disketten? Auf dem Software-Markt gibt es eine Vielzahl nützlicher Utilities. Was leisten sie?

Nach Anklicken des zu vergleichenden Files ist das Programm aktiviert. Falls das Destination-File nicht mit dem Source-File übereinstimmen sollte, öffnet sich ein Fenster mit einer Fehlermeldung. Nun können Sie zwischen zwei Gadgets mit der Aufschrift »weiter« oder »Abbruch« wählen. Natürlich können nicht nur Files, sondern auch Disketten miteinander verglichen werden.

Das Kopierprogramm »DCopy II« glänzt mehr durch Schlichtheit als durch eine Vielzahl an Gadgets. Doch dies soll keinesfalls ein Manko sein, es stellt eher seine Benutzerfreundlichkeit und Effektivität unter Beweis. Das Programm wird aus dem CLI oder der Workbench gestartet. Danach zeigt sich ein kleiner Bildschirm, der vier Optionen enthält: »Change Parameters«, »Begin Copy«, »Statistics« und »Quit«. Diese Optionen können jeweils über die Funktionstasten aktiviert werden. Interessant ist der Menüpunkt »Change Parameters«. Hier werden die Voreinstellungen vorgenommen. So können Sie beispielsweise »Verify« einbeziehungsweise ausschalten, oder gar die Anzahl der Schreibversuche beim Auftreten eines Verify-Errors angeben. Mit »StartTrack« und

»EndTrack« definieren Sie den zu kopierenden Bereich. »HEAD« gibt an, welche Diskettenseite kopiert werden soll, entweder nur Kopf Null, nur Kopf Eins oder beide Köpfe.

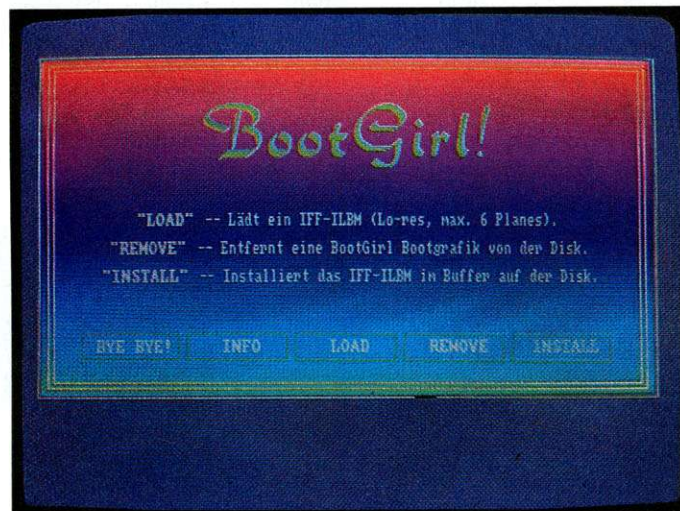
»Multiple Copy« ermöglicht das mehrmalige Schreiben auf nur ein Laufwerk, aber nicht auf verschiedene Laufwerke. Zusätzliche Destination-Parameter gestatten es, die Quelldiskette auf mehrere, maximal vier Laufwerke gleichzeitig zu kopieren. Die Beschleunigung des Kopiervorgangs wurde mit einem Trick realisiert. Denn während das Programm Track x auf die Zieldiskette schreibt, liest es bereits Track x+1. Hierdurch wird eine Beschleunigung von nahezu dreißig Prozent gegenüber anderen Kopierprogrammen erreicht. Nach der Aktivierung des Menüpunktes »Statistics« baut sich auf dem Bildschirm eine Grafik auf, die den vorherigen Kopiervorgang darstellt. Hier kann man nun ablesen, wo sich ein Verify-Error, ein Read-Error oder noch freier Speicherplatz befindet. Bei den Grafikkünstlern unter den Amiganern sollte das Programm »BootGirl V1.28« auf großes Interesse stoßen. Hier handelt es sich um ein Utility, mit dem es möglich ist, IFF-

Bilder in den Bootvorgang mit Musikuntermalung einzubinden. Hierzu steht Ihnen ein recht einfaches Gadget-Menü zur Verfügung. Nach Anklicken des Load-Gadgets öffnet sich ein Requester, in dem der Name der einzuladenden Grafik einzusetzen ist. Die Option »Remove« dient dazu, die alte Bootgrafik zu entfernen. Anschließend kann mit der Option »Install« die neue Grafik installiert werden. In das Programm ist zusätzlich eine SCA-Virusscanner-Funktion eingebunden, die, wie der Name schon sagt, die Diskette auf mögliche SCA-Viren untersucht.

Ein weiteres Kopierprogramm bietet sich mit »Select-Copy« an. Vorab muß jedoch gesagt werden, daß dieses Programm nur lauffähig ist, wenn zwei Laufwerke angeschlossen sind. Ein besonderes Feature dieses Programms: Es kann Fremdformate wie zum Beispiel das einseitige Atari-Format einlesen und kopieren. Mit seinen vielen Gadgets ist das Programm zudem sehr benutzerfreundlich. Das Gadget »Bamcopy« kopiert nur die belegten Zylinder, die freien Stellen werden auf der Zieldiskette lediglich formatiert. Hierdurch wird eine Beschleunigung erreicht. »Nibblecopy« kopiert alles, was dem Laufwerk »untergeschoben« wird. Denn die Option berücksichtigt keine Diskfehler. Das Gadget »ST/IBM-Copy« kopiert, wie oben schon angesprochen, sowohl MS-DOS-Formate, als auch Atari-Formate.

Insgesamt scheint die Diskette eine sinnvolle Utility-Sammlung zu sein. Das täuscht aber nicht darüber hinweg, daß Public-Domain-Programme in der Anschaffung günstiger sind als kommerzielle Programme.

(Jürgen Seibel/vb)



Grafisch nett aufbereitet: das Programm »BootGirl«

AMIGA DOS Blitzlicht

Name: Disk Utilities Nr. 2
Vertrieb: Markt & Technik
Preis: 49,- DM

Positiv:

- fünf Utilities
- kopiert Fremdformate (PC, ST)
- übersichtliche Menüs
- Modifizieren von Daten auf Diskette per ASCII-Code möglich

Negativ:

- Select-Copy nur mit zwei Laufwerken lauffähig

Der musikalisch ambitionierte Amiga-User sieht sich inzwischen einer großen Auswahl von Musikprogrammen gegenüber. Das wichtigste Auswahlkriterium beim Kauf ist wohl die Art der Noteneingabe. »The Musical Enlightenment« geht hier den herkömmlichen Weg. Nahezu alle Eingaben erfolgen im hexadezimalen Zahlenformat. Die Möglichkeit, Noten direkt über die Tastatur einzugeben (RECORD), fehlt hier völlig. Wer bisher mit dem »Soundtracker« komponiert hat, wird hierbei kaum Schwierigkeiten haben. Anfänger stehen vor einem Problem, das auch das mitgelieferte Handbuch kaum zu lösen vermag. Dieses Handbuch, in englischer Sprache verfaßt, glänzt durch auf ein Minimum zusammengefaßte Informationen. Dies erhöht zwar die Übersicht, der Lerneffekt bleibt jedoch auf der Strecke.

In allen anderen Punkten ist der Track-Editor vorbildlich gestaltet. Jeder Parameter eines Songs ist übersichtlich auf dem Bildschirm dargestellt. Über die Menüleiste sind die restlichen Module von »The Musical Enlightenment« schnell erreichbar.

Grafische Hüllkurven

Echtes Synthesizer-Feeling offenbaren die Klangbearbeitungsmenüs. Für jedes der 32 verwendbaren Instrumente werden die Hüllkurve und die Lautstärke grafisch auf dem Bildschirm dargestellt. Die Werte der Hüllkurven können auf einfache Art mit Schieberreglern manipuliert werden und erlauben so weiträumige Klangveränderungen. Wie bei einem professionellen Sampler kann der Anwender die Anschlagshärte (ATTACK), die Dauer bis zur größten Lautstärke (DECAY), Haltevolumen (SUSTAIN) und Ausklingzeit (RELEASE) einstellen. Jedes Instrument kann außerdem einen Effekt verwenden, der vom leichten Vibrato über Tremolo bis zum Spezialeffekt reicht. Das Editieren der Effekte findet zusammen mit den Arpeggios auf einem separaten Bildschirm statt. Hier wird, außer zum Verändern des LFO (Oszillator mit niedriger Frequenz), die Tastatur als Eingabegerät verwendet. Mit der

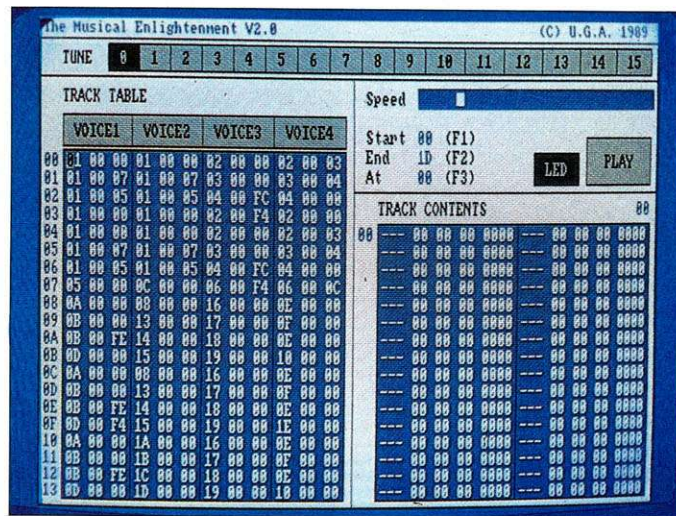


Bild 1. Das Track-Menü von »The Musical Enlightenment«

Musikalische Erleuchtung !?!

Viel Musik für wenig Geld

Das holländische Software-Haus UGA hat es sich zum Ziel gemacht, preiswerte Software zu produzieren. Als Einstieg präsentieren sie ein Musikprogramm, das sich hören lassen kann.

Maus können 16 verschiedene LFOs erzeugt werden, welche später die Wellenform modulieren.

Samples mühelos verändern

Das Menü zum Editieren der Klangquellen hält einige

Überraschungen bereit. Als Rohmaterial können neben IFF-Samples auch selbst erzeugte Wellenformen Verwendung finden, die mit der Maus auf den Bildschirm gezeichnet werden. Dieses Klangmaterial kann anschließend verstimmt, transponiert oder gemischt werden. So ent-

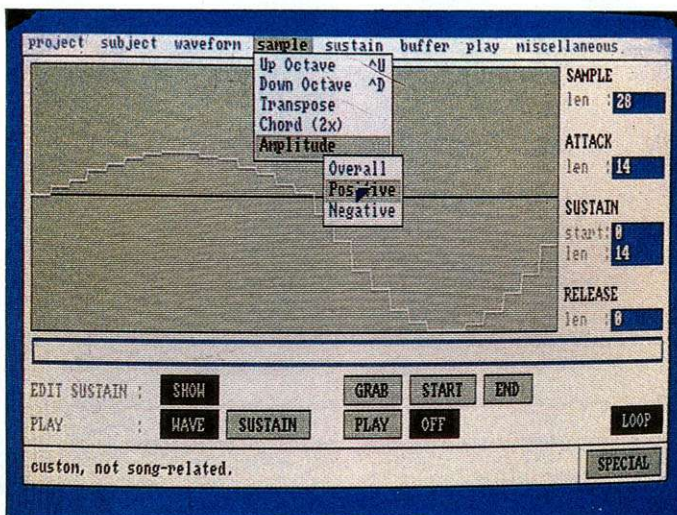


Bild 2. Das Sample-Menü gestattet vielfältige Klangveränderungen

stehen komplexe Wellen, die dem Klang eines natürlichen Instruments verblüffend nahe kommen. Instrumente lassen sich auf Diskette abspeichern und finden anschließend in selbsterstellten Musikstücken Verwendung.

Diese Musikstücke werden im Track-Menü erstellt und können dort auch sofort zur Kontrolle abgespielt werden. Wenn der Song zur Zufriedenheit des Komponisten ausgefallen ist, wird die Arbeit auf Diskette gesichert. Ein gepacktes abgespeicherter Song kann mit einem beigefügten Abspielprogramm vom CLI aus gestartet werden. Ungepackte Songs eignen sich zur Weiterverarbeitung in eigenen Programmen. Leider schweigt sich das Handbuch zu diesem Thema aus und erwähnt in keiner Weise, in welchem Format die Daten vorliegen. Somit bleibt es den versierteren Programmierern überlassen, Songs weiterzuverwenden.

Preiswert und gut

Die Firma UGA hat mit »The Musical Enlightenment« ein Programm auf den Markt gebracht, das trotz einiger Mängel ein gut durchdachtes Produkt darstellt und sein Geld allemal wert ist. Es bleibt zu wünschen, daß bei weiteren Produkten aus diesem Hause mehr Wert auf detailliertere Dokumentation gelegt wird. Auf jeden Fall darf man auf weitere Programme von UGA gespannt sein.

(tb)

AMIGA DOS Blitzlicht

Name: The Musical Enlightenment

Hersteller: UGA Software Development, PO BOX 881, 3700 AW Zeist, Holland

Vertrieb: UGA Software Development

Preis: 59,- DM

Positiv:

- verwendet IFF-Samples
- günstiger Preis

Negativ:

- spärliches Handbuch in Englisch
- keine Einbindungshinweise für eigene Programme
- unkomfortable Eingabe

Bernd Rudolf

Schnappschuß vom Video

Praxistest des Digitizers Snapshot

In die Reihe der Digitizer hat sich seit einiger Zeit ein Gerät eingereiht, das vor allem durch sein profihaftes Aussehen auffällt – der Snapshot. Ob bei der Leistung ebenfalls hohe Ansprüche gestellt werden können, haben wir in der Praxis getestet.

Meine Güte, sooo viele Knöpfe! Das ist der erste Eindruck, den einem das Snapshot Studio+ beim Auspacken vermittelt. Doch das anfängliche Erschrecken angesichts der vielen Einstellmöglichkeiten ist schnell überwunden. Dann präsentiert sich dieser Digitizer als ein Universal-Genie, das einen schnell die Zeit vergessen läßt, hat man erst einmal mit dem Digitalisieren begonnen. Im Lieferumfang sind der Digitizer, das Verbindungskabel zum Amiga, zwei Cinch/Cinch-Verbindungskabel, zwei BNC/BNC-Verbindungskabel, ein Vierfach-Cinch/Cinch-Kabel und vier Adapter (BNC/Cinch – Cinch/BNC – BNC/BNC – Cinch/Cinch) enthalten. Dies ist besonders vorteilhaft, da alle Videoanschlüsse des Digitizers auf BNC ausgelegt sind und nicht jeder Käufer

solche Kabel sofort zur Verfügung hat. Die ausführliche Anleitung wurde in Deutsch verfaßt und ist gut verständlich. Das Snapshot Studio+ kann die verschiedensten Signale verarbeiten: vom BAS-Signal (schwarzweiß) einer Überwachungskamera über das FBAS-Signal (farbig) eines Camcorders oder Videorecorders, das Y/C-Signal (SW- und Farbanteil getrennt) eines S-VHS-Recorders oder einer Hi-8-Kamera bis hin zum RGBS-Signal (Rot-, Grün-, Blau- und Synchron-Signal) einer Studiokamera. Schon daran erkennt man, für welche Art von Anwendungen der Digitizer konstruiert wurde.

Es ist kein Problem, alle eben genannten Bildquellen gleichzeitig an das Studio anzuschließen, denn es verfügt neben den vier softwaremäßig schaltbaren Kanälen über



Bild 1. Der "Snapshot Studio + "-Digitizer

zwei dreistufige Schalter an der Gehäusefront, mit denen man jederzeit auf eine andere Signalquelle umschalten kann. Um das Auffinden bestimmter Szenen aus einem Film zu erleichtern, wird zu jedem Digitizerkanal ein Audiokanal mitgeschaltet. Der interne RGB-Splitter sorgt für eine optimale Trennung der Mischsignale. Dabei können die Farbanteile einzeln nachgeregelt werden. Ebenso kann hardwaremäßig die Helligkeit, der Kontrast und die Farbintensität eingestellt werden. Eine sehr nützliche Einrichtung ist die Einstellmöglichkeit der Wandlergrenze für schwarz und weiß. Sie dient dazu, die Bildqualität durch sehr feine Justierung der Drehregler stark zu optimieren. Ein Regler, mit dessen Funktion man anfangs überhaupt nichts anzufangen weiß, ist der Lace-Regler. Er ist

dazu da, die Halbbilderkennung zu regeln. Im Klartext heißt das, daß ein stark zitterndes Interlace-Bild der Vergangenheit angehört, sofern der Regler richtig eingestellt ist. Interessant wird es jedoch, wenn der Regler "falsch" positioniert ist. Dann kann es unter Umständen passieren, daß man nach dem Digitalisieren ein Bild erhält, welches aus zwei verschiedenen Halbbildern zusammengesetzt wurde. Je nach Originalszene kann das zu erstaunlichen Effekten führen, so zum Beispiel einer scheinbar durchsichtigen Person, die im Bild zweimal in unterschiedlicher Position dargestellt wird, da sie gerade in Bewegung war. Was nun noch fehlt, ist ein Ausgang für einen Kontrollmonitor, aber auch der ist beim Snapshot vorhanden und läßt sich zwischen den verschiedenen Eingangskanä-

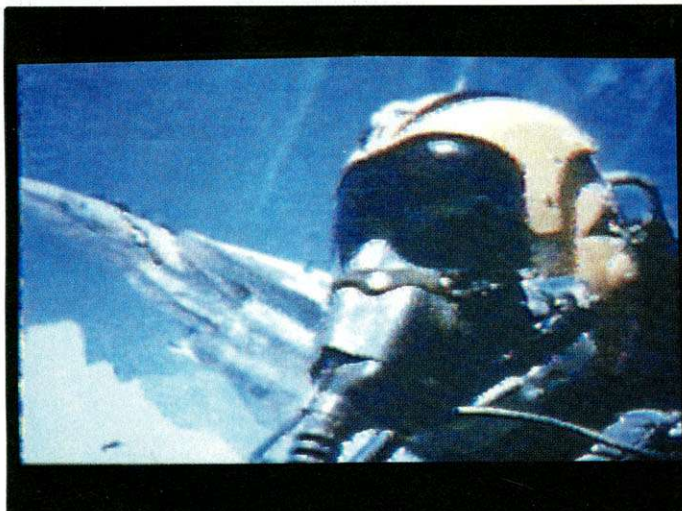


Bild 2. Standbild eines VHS-HQ-Videorecorders, digitalisiert in einer Auflösung von 320x256 Punkten und 4096 Farben und Vierfach-Oversampling



Bild 3. Von einem Video8-Camcorder direkt digitalisiertes HAM-Bild

176*136*16	G	1/50 s
352*276*16	G	1/50 s
704*552*16	G	4 s
176*138*4096	F	4/5 s
352*276*4096	F	1.7 s
704*552*16	F	4 s

Tabelle: Digitalisierungszeiten inklusive Bildaufbau

len umschalten. Die Software ist übersichtlich gehalten und somit einfach zu bedienen. Sehr vorteilhaft ist es, daß sich diverse Einstellungen des Digitizers mit der Maus vornehmen lassen. Auf diese Weise kann man einen der vier Kanäle auswählen, den Color- oder Smooth-Filter für die einzelnen Kanäle ein- oder ausschalten oder festlegen, welches der beiden Halbbilder (even/odd) gelesen werden soll, wenn die vertikale Auflösung 276 Zeilen beträgt.

Zum Snapshot die Software mit "Format"

Bei dieser hohen Anzahl von Zeilen ist man zu Anfang recht erstaunt, doch muß man beachten, daß das Programm im Overscan-Modus arbeitet. Die "normalen" Auflösungen betragen horizontal 160 oder 320 Bildpunkte in der Auflösung »LoRes«, 320 Bildpunkte und 640 Punkte im »HiRes«, vertikal 128 und 256 Bildpunkte normal, 256 im »Interlace« und 512 Punkte. Die Anzahl der Farben liegt je nach Auflösung bei 2, 4, 8, 16, 32, 64 (Extra Halfbright) oder 4096

(Hold and Modify). Hier ist es sehr interessant, einmal näher auf die Digitalisierungszeiten (siehe Tabelle) einzugehen, da sie neben der Qualität eines der Hauptmerkmale bei Digitizern sind, die auch hervorragend zum Vergleich herangezogen werden können: Bilder mit einer Auflösung bis 352x276 Punkten und nicht mehr als 16 Graustufen werden in Echtzeit digitalisiert, das heißt in diesen Fällen ist kein Standbild vonnöten. Es ist nur unheimlich schwierig, wirklich das gewünschte Bild zu treffen. In der derzeitigen Version ist es auch noch nicht möglich, eine gesamte Szene einzulesen und zu animieren. Diese Option ist aber bereits in Arbeit.

Es bleibt allerdings der Nachteil, daß ein Videorecorder ein störstreifenfreies Standbild und eine Einzelbildweitschaltung benötigt, um für das Digitalisieren von Szenen geeignet zu sein. Für diesen Fall ist auch die Funktion "Remote" gedacht. Eine noch in der Entwicklung befindliche "Fernsteuerung" (Interface) wird es irgendwann ermöglichen, den Videorecorder per Maus über die Software zu bedienen.

Eine besonders gute Funktion ist »Multichannel«. Sie ermöglicht das gleichzeitige darstellen aller vier Kanäle auf dem Bildschirm. Schade ist nur, daß der Bildaufbau trotz SW-Bild und niedrigster Auflösung sehr lange dauert, so daß es kaum noch möglich ist, die Szenen der einzelnen Bildquellen zu verfolgen. Da es laut Hersteller bisher Speicherprobleme gab, wird ein Programm, mit dem man un-

abhängig von der eigentlichen Digitizer-Software Animationen erstellen kann, mitgeliefert. Leider unterstützt es nur die eine Auflösung von 160x128 Punkten in 16 Graustufen. Auch hier ist wieder die Einzelbildweitschaltung des Videorecorders gefragt, da die Bilder zwar in Echtzeit in den Speicher des Digitizers (96 KByte) eingelesen werden, von dort aus aber einige Zeit bis zur Darstellung auf dem Bildschirm benötigen. Abschließend bleibt noch anzumerken, daß neben dem Studio+ auch noch der pure Digitizer, ein extra RGB-Splitter und RGBS-Farbadapter erhältlich sind. Zusammengekommen ergeben diese drei Einheiten zwar prinzipiell das Studio, sie sind jedoch noch extern zu verkabeln und verfügen nicht über alle Funktionen des Studios.

Insgesamt gesehen handelt es sich bei dem Snapshot Studio+ um einen Digitizer, der seinen Einsatz im professionellen Bereich findet. Er liefert eine hervorragende Qualität und ist gut zu bedienen. Schade ist nur, daß er trotz seiner Professionalität und seines stolzen (aber durchaus angemessenen) Preises nicht in der Lage ist, bei einer Auflösung von 320x256 Punkten in 16 Graustufen die Geschwindigkeit eines Vidi-Grab zu erreichen und somit Animationen zu ermöglichen.

(jb)



Bild 4. Standbild eines Video8-Camcorders, digitalisiert in einer Auflösung von 320x256 Punkten und 4096 Farben

CSV-HIGHLIGHTS

Commodore	
Speichererweiterung Commodore 1050 (256 kB)	39,-
Commodore Farbmonitor 1084 Stereo	589,-
Commodore Amiga 500	799,-
Amiga 500 + Farbmonitor 1084 S	1369,-
Speichererweiterung auf 1 MB mit Uhr	189,-
Commodore Amiga 2000	1799,-
Amiga 2000 + Farbmonitor 1084 S	2369,-
Amiga 2000 + PC/XT-Karte + 40 MB Filecard	
Western Digital + Epsomdrucker LX 400	3599,-
3 1/2"-Laufwerk intern für Amiga 2000	199,-
PC/XT-Karte mit 5 1/4"-Laufwerk	629,-
AT-Karte mit 5 1/4"-Laufwerk	1699,-
SCSI Controller Commodore A 2090 A	599,-
20-MB-Festplatte für Amiga 2000 mit SCSI	
Controller Comm. 2090 A (autobootend)	949,-
40-MB-Festplatte mit Controller 2090 A	1299,-
40-MB-Filecard autobootfähig	1199,-
20-MB-Filecard (Seagate, 40 ms) für	
A 2000 mit PC-Karte oder A 1000 / Sidecar	599,-
30-MB-Filecard (Seagate, 40 ms)	749,-
40-MB-Filecard (Western Digital, 29 ms)	829,-
50-MB-Filecard (Seagate, 40 ms)	949,-
2-MB-RAM-Erweiterungskarte für A 2000, aufrüstbar bis 8 MB	799,-

Atari	
Festplatte Atari Megafile 30	869,-
Festplatte Atari Megafile 60	1279,-
1040 STFM + Monochrommonitor SM 124	1179,-
1040 STFM + SM 124 + Megaflo 30	2029,-
Atari STE + Monochrommonitor SM 124	1399,-
Atari Computer Mega ST 1 mit Maus +	
Monochrommonitor SM 124	1479,-
Mega ST 1 + SM 124 + Megaflo 30 MB	2329,-
Atari Mega ST 2 + Monochrommonitor SM 124	2169,-
Atari Mega ST 2 + SM 124 + Megaflo 30	2999,-
Atari Mega ST 4 + Monochrommonitor SM 124	3149,-
Atari Mega ST 4 + SM 124 + Megaflo 30	3999,-
Supercharger für Atari ST	649,-
80-MB-Festpl. Seagate ST 1096 N (SCSI)	999,-

Epsomdrucker (dt. Handbücher)	
LX 400	399,-
LQ 400 (24-Nadel-Drucker)	629,-
LQ 550 (24-Nadel-Drucker)	779,-
Tintenstrahlendrucker IX 300 (9 Düsen, NLQ, max. 240 Zeichen/Sekunde)	499,-

Stardrucker (dt. Handbücher)	
LC-10 mit Centronicsinterface	429,-
LC-10 Color, Farbdrucker mit Centronics	569,-
LC 24-10 mit Centronicsinterface	649,-
XB 24-10 mit Centronicsinterface	1349,-

NEC-Drucker (dt. Handbücher)	
NEC P 6 Plus 1099,-; Farboption	249,-
Einzelblatteinzug für NEC P 7 Plus	399,-
NEC P 60 1499,-; NEC P 70	1899,-

NEU: IBM-kompat. AT (12 MHz, 512 kB)	1049,-
Druckerkabel 5 m lang für Amiga, ST	29,-
EGA-Karte (800x600) + 14"-EGA-Farbmonitor	649,-
VGA-Karte Optima 16 Bit, 512 kB	299,-
VGA-Karte Trident 16 Bit, 512 kB	269,-
Multisynch. Farbmon. (0,28 mm, 1024x768)	979,-
VGA-Farbmonitor (Aufl. 1024x768)	779,-

Versandkostenpauschale: Inland DM 12,-; Ausland DM 40,- je Paket. Lieferung nur gegen NN oder Vorauskasse; Ausland nur Vorauskasse. Preise gültig ab 13.8.90.

CSV RIEGERT GmbH

Gärtnersstr. 4, 7320 Göppingen
Tel. 07161/13591, Fax 07161/13587

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

AMIGA DOS Blitzlicht

Name: Snapshot Studio +
Anbieter: Videotechnik Diezemann, Dammstr.42, D-2300 Kiel 1, Tel.: 0431/94424
Funktion: Video-Digitizer
Preis: 2795,- DM

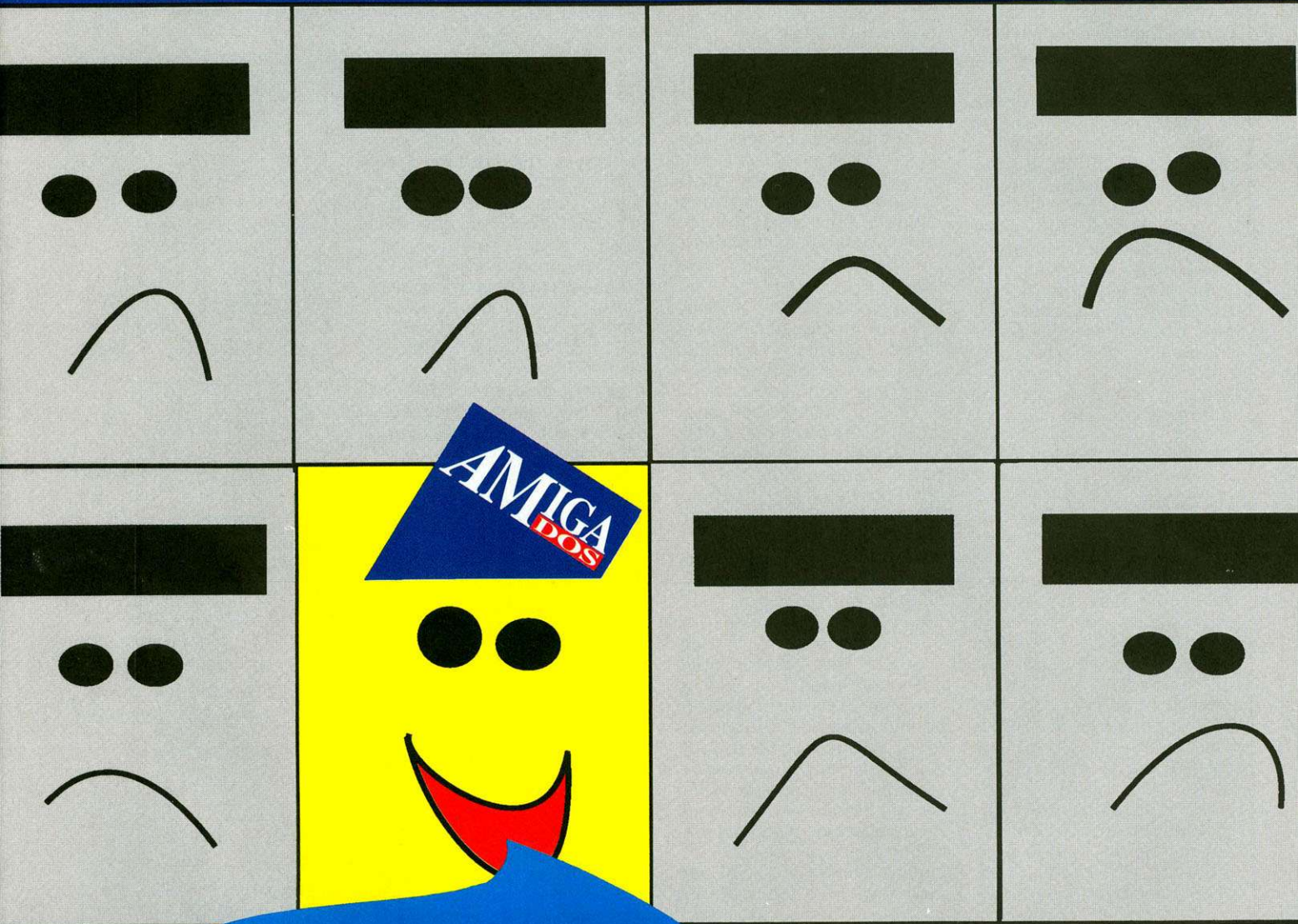
Positiv:

- alle Auflösungen möglich
- Bildqualität fein einstellbar
- Software anwenderfreundlich
- Videokabel im Lieferumfang enthalten

Negativ:

- Übertragung zu langsam
- Handbuch nicht ausführlich genug

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



PageSetterII / TransScript

Desktop Publishing (kurz DTP) wird immer wichtiger beim Herstellen von Zeitungen oder sonstigen Publikationen. Mittlerweile sind viele Programme auf dem Markt, die jedoch nicht alle ganz billig sind.

GoldDisk möchte nun auch DTP für den Heimanwender interessant machen. Es handelt sich um PageSetterII, welches 198,- DM kostet. Dazu wird noch die Textverarbeitung TransScript angeboten, die noch einmal mit 99,- DM zu Buche schlägt. Ich möchte Ihnen dieses Komplettpaket für weniger als 300,- DM in diesem Artikel vorstellen.

existieren noch zwei Arten von Grafikboxen, eine für Bitmapgrafik und eine für Vektorgrafik (siehe Ausg. 7). Für jede Boxart gibt es vielfältige Gestaltungsmöglichkeiten.

Die Textbox

Um Text in eine Box einfließen zu lassen, muß einfach eine Box mit dem Schreibtool aktiviert werden. In der Box erscheint dann ein Cursor, und der Anwender kann mit der Schreibarbeit beginnen. Möchten Sie nicht den ganzen Text in PageSetterII erfassen (was natürlich nicht gerade sehr schnell ist, da Text als Grafik dargestellt wird) ist es sinnvoll, den Text mit einer anderen Textverarbeitung zu erfassen und dann zu importieren. PageSetterII unterstützt hier alle gängigen Textformate von WordPerfect bis

Textcraft. Sollte Ihre Textverarbeitung nicht dabei sein, so speichern Sie ihre Texte einfach als ASCII-Text ab und importieren Sie ihn dann. Leider gehen so aber die Formatierungen verloren. Wünschenswert wäre es noch, daß PageSetterII das Textformat automatisch erkennt. Leider muß der Anwender das Format vor dem Einladen per Hand einstellen. Auch das Importieren von Beckertext-Texten sollte noch implementiert werden. Besonders unterstützt werden noch die Anwender von TransScript, der Textverarbeitung aus dem Hause GoldDisk. Startet man nämlich TransScript im Hintergrund, so kann man über den Menüpunkt TransScript den aktuellen Text direkt ohne etwas speichern zu müssen, nach TransScript transportieren. Dort kann er dann

viel einfacher bearbeitet werden. Ist der Text fertig, so kann er über den Menüpunkt Text zurückschicken wieder direkt nach PageSetterII importiert werden. Somit wird die Texterfassung sehr einfach, zumal beide Programme die gleichen Steuercodes besitzen. Ist der Text erfaßt, so muß er natürlich layoutet werden. Wichtig ist hier natürlich zuerst einmal die Schriftart. PageSetterII nutzt die Compugraphic Outlinefonts, wovon die beiden Schrifttypen Times und Helvetica mitgeliefert werden. Hierbei handelt es sich um sehr schöne Vektorfonts, die beliebig ohne Qualitätsverlust vergrößert und verkleinert werden können. Hier sind alle Größen bis zu 127 Punkt möglich. Ab sechs Punkt erscheint eine Schrift leserlich. Das dürfte für die meisten Anwendungen ausreichen. Allerdings können Schriftgrößen nur in ein-Punkt-Schritten verändert werden, was aber für den Heimbereich ausreichen sollte. Neben den Compugraphic-Schriften wird noch ein PD-Utility namens MakeMetric mitgeliefert, welches Amiga-Fonts auch für PageSetterII zugänglich macht. Allerdings traten hiermit Fehlfunktionen auf. Außerdem sollte man von den Amiga-Schriften sowieso nicht Gebrauch machen, da deren Schriftqualität nicht gut ist. Natürlich kann man auch die Textattribute ändern. Zur Verfügung stehen fett, unterstrichen, kursiv, umrandet (outline), schatten, hoch- und tiefgestellt. Hier sind noch zusätzliche Attribute wie doppelt-unterstrichen oder durchgestrichen wünschenswert. Texte müssen nicht immer schwarz erscheinen. PageSetterII bietet die Möglichkeit, Buchstaben mit verschiedenen Mustern zu füllen. Dabei kann der Anwender die Muster selber umdefinieren. Leider besteht aber nicht die Möglichkeit, Farbverläufe zu erzeugen, was sehr schöne Effekte ergibt. Hier muß PageSetterII anderen DTP-Programmen Tribut zollen. Auch der Zeilenabstand kann mit PageSetterII frei gewählt werden.

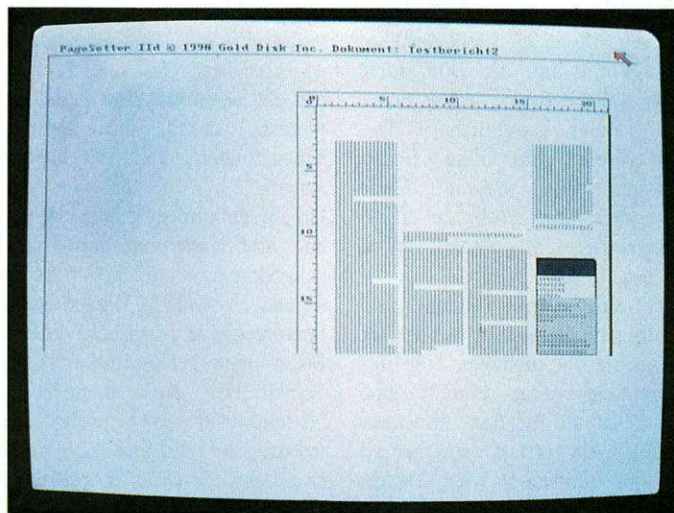


Bild 1: Der PageSetter Desktop. Rechts befindet sich diese Seite in 50%iger Vergrößerung. Links ist die Ablage.

Um es gleich vorweg zu nehmen.

PageSetterII ist in dieser Preisklasse bisher einmalig. Das liegt sicher auch daran, daß PageSetterII nicht eine neue Version des alten PageSetters ist, sondern eine abgespeckte ProfessionalPage-Version. Man hat lediglich einige Dinge wie PostScriptausgabe, Farboption und Trennhilfe entfernt. Widmen wir uns zuerst PageSetterII, dem Herz unseres Systems.

PageSetterII wird auf zwei Disketten und mit einem deutschen Handbuch ausgeliefert. Das Handbuch ist gut gegliedert und führt in die Arbeit mit PageSetterII gut ein. Auf den Disketten befindet sich das Programm PageSetterII und eine Utilities. Auf der zweiten Diskette befinden sich die beiden Fonts Times und Helvetica, die beiden Stammschriften in der Satztechnik.

Ist das Programm installiert, so kann man PageSetterII starten. Es öffnet sich ein Screen, der Desktop. Zuerst muß eine Seite erstellt werden. Dazu wählt man den entsprechenden Menüpunkt an. Es erscheint ein Requester, in dem man zwischen einigen Standardformaten auswählen kann oder wahlweise ein eigenes Seitenformat angibt. Möchte man auf einer Seite mehrere Spalten darstellen, so kann das hier gleich in diesem Requester eingestellt werden. Natürlich können auch mehrere Seiten auf einmal erstellt werden. Hat man diese Einstellungen betätigt, so erstellt PageSetterII die Seiten und stellt die erste Seite auf dem Desktop dar. Leider wird immer nur eine Seite auf dem Desktop angezeigt, obwohl mitunter Platz für zwei Seiten wäre.

PageSetterII arbeitet boxorientiert, d.h. alle Elemente, egal ob Grafik oder Text, werden in einer Box dargestellt. Um das Layout des Dokumentes zu ändern, muß der Anwender die verschiedenen Boxen abändern. PageSetterII bietet drei Arten von Boxen an. Zum einen wäre da die Textbox. Wie der Name schon sagt, wird in ihr der Text eines Dokumentes dargestellt. Neben den Textboxen

Auch der Abstand zwischen zwei Paragraphen kann frei bestimmt werden. Wahlweise können auch Absätze ein oder ausgerückt werden, so daß die erste Zeile immer hervorgehoben wird. Möchten Sie den ersten Buchstaben eines Absatzes einrücken, so können Sie auch dies mit PageSetterII tun. Allerdings gelten solche Einstellungen immer für das ganze Dokument.

Kommen wir noch einmal zurück zu den Textboxen. Logischerweise paßt ein Text nicht immer in eine Box. Um ein Abschneiden des Textes zu vermeiden, können in PageSetterII verschiedene Boxen auch über mehrere Seiten verbunden werden. So können einzelne Textboxen verändert und gelöscht werden, ohne das deren Text verloren geht. PageSetterII formatiert den Text immer neu.

Das Erstellen von Boxen ist übrigens sehr einfach. Entweder man gibt beim Erstellen einzelner Seiten direkt die Anzahl der Spalten an und läßt diese direkt erzeugen, oder man gibt die Anzahl der Spalten an und erstellt dann die gewünschte Box per 'Autoklick'. PageSetterII erstellt dann die Box so, daß Sie genau so groß ist wie eine Spalte des Dokumentes.

Grafik

Natürlich wird ein Dokument nicht nur Text, sondern auch Grafik enthalten. Hier hat der Anwender die Möglichkeit, sowohl Bitmapgrafiken als auch Vektorgrafiken im Vektorformat Draw-Format einzuladen. Bitmap-Grafiken können beliebig vergrößert und verkleinert werden, allerdings wird bei verkleinerten Bitmapgrafiken die Ausdruckqualität nicht verbessert. Die Grafik kann auch innerhalb der eigenen Box verschoben werden. So wird es möglich, nur einen Teil einer Grafik darzustellen. Übrigens werden alle Formate von LoRes bis HAM verarbeitet und in Graustufen konvertiert. Ebenso können Vektorgrafiken eingelesen und verändert werden.

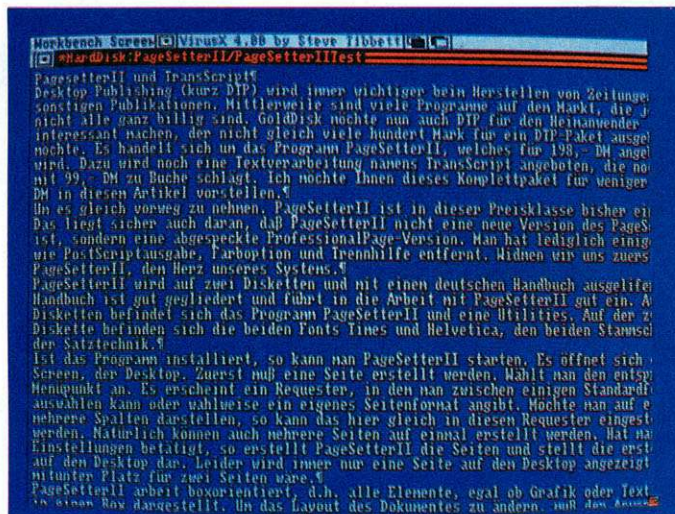


Bild 2: TransScript: Einfach, aber als Editor für PageSetterII gut geeignet, da das Programm schnell ist und einige Sinnvolle Funktionen beinhaltet.

Dazu bietet PageSetterII noch einige Grundfunktionen zum Erstellen von Vektorgrafiken innerhalb des Programms. Dazu dienen die Grundfunktionen Kreis, Ellipse, Rechteck, Quadrat und Polygon. Diese Figuren können mit frei bestimmbar Linienstärken und Füllmustern erstellt werden. Diese Tools reichen zwar aus, um elementare Grundzeichnungen zu erstellen, zum Erstellen von komplexeren Grafiken sind sie allerdings nicht geeignet. Hier muß der Anwender auf ProDraw zurückgreifen.

Somit nun zu den drei Boxentypen. Natürlich ist es auch wichtig, wie die drei Typen harmonisieren, und wie aufwendig die Erstellung eines Dokumentes ist. PageSetterII bietet die Möglichkeit, Text um Grafikboxen herumfließen zu lassen. Allerdings fließt der Text immer nur um die Box, nicht aber um die Grafik selber. So ist es nicht möglich, Text kreisförmig um eine Grafik fließen zu lassen. Diese Grundfunktion sollte ein DTP-Programm eigentlich schon bieten.

Ist eine Box einmal zur Zufriedenheit erstellt worden, so kann Sie vor jeglichen Zugriffen geschützt werden. Eine 'locked-Box' ist so für den Anwender nicht mehr abänderbar, bis sie wieder 'unlocked' wird. Da der Text- und Grafikaufbau beim Verschieben einer Box einiges an Zeit braucht, kann der Inhalt

einer Box wahlweise nicht dargestellt werden. Der Bildaufbau erfolgt dann wesentlich schneller.

In dem Menü "Einstellungen" kann der Anwender noch einige zusätzliche Dinge einstellen. Hier sind vier Vergrößerungsfaktoren (33%, 50%, 100% und 200%) einstellbar. Der Faktor kann aber nicht selber eingestellt werden. Auch kann man hier festlegen, ob der Screen im Interlace-Modus geöffnet werden soll oder nicht. Auch ein Raster kann in einem Requester aktiviert werden. Das Lineal am Rand eines Dokumentes kann wahlweise ein- oder ausgeschaltet werden. Der Anwender kann übrigens frei festlegen, ob er in Pica, Zentimeter oder Inch arbeiten möchte. Leider lassen sich diese Voreinstellungen nicht abspeichern. Sie werden lediglich mit dem Dokument abgespeichert. So muß immer erst ein leeres Dokument geladen werden, damit die eigenen Einstellungen zur Verfügung stehen.

Ist ein Dokument fertig erstellt, so muß es natürlich noch ausgedruckt werden. Hier ist zu sagen, daß PageSetterII hervorragende Ausdrucke liefert. Sogar eine sechs Punkt große Schrift ist auf einem 24-Nadeldrucker noch sehr gut lesbar. Der Treppeneffekt tritt kaum auf, die Drucker werden in ihrer maximalen Auflösung unterstützt. Natürlich sind auch Probedrucke in niedrigen

Auflösungen möglich. Das spart Zeit und schon das Farbband.

TransScript

Eben habe ich bereits erwähnt, daß es möglich ist, Text direkt von und nach TransScript zu portieren. Wer noch keine Textverarbeitung besitzt, sollte TransScript in seine Überlegungen mit einbeziehen, denn so wird die Texterfassung im Verbund mit PageSetterII wirklich zum Kinderspiel. TransScript bietet viele Funktionen, die andere Textverarbeitungen auch bieten, allerdings arbeitet es nach einem etwas anderem Prinzip. Auf dem Bildschirm werden Steuercodes direkt angezeigt. Somit ist kein WYSIWYG möglich. Lediglich Textattribute werden direkt am Bildschirm angezeigt. Dadurch ist ein sehr schnelles Arbeiten möglich.

Neben den Grundfunktionen einer Textverarbeitung wie Suchen/Ersetzen und den gängigen Blockoperationen bietet TransScript noch einige Zusatzfunktionen. So lassen sich beispielsweise Serienbriefe oder Indizes erstellen. Allerdings geschieht das erst beim Ausdruck des Textes, der auch wahlweise in eine Datei erfolgen kann. Somit hat man keine direkte Kontrolle über sein Dokument. Auch getrennt wird erst beim Ausdruck. Trennfehler lassen sich so nicht mehr korrigieren. Solche Mängel sollte man umgehen können.

Wer sich den Ausdruck vor dem Druck einmal anschauen möchte, für den ist die Previewfunktion nützlich. Der Text wird in Originalgröße auf dem Bildschirm dargestellt und kann per Maus gescrollt werden.

Einmalig in dieser Preisklasse ist wohl die Macrofunktion. Mit ihr können einfache kleine Programmabläufe erstellt werden, die das Arbeiten erheblich vereinfachen. Eine Rechtschreibkorrektur ist nicht implementiert, allerdings existiert ein Port zu einem Spellchecker, der ebenfalls bald erscheinen wird. Somit lassen sich dann Texte in TransScript direkt korrigieren.

Fazit

Das Gespann PageSetterII mit TransScript ist im Preis/Leistungsverhältnis sehr gut. Der Anwender erhält ein DTP-Programm, welches für den Heimbereich vollkommen ausreicht. Die Schriften sind hervorragend und werden auch sehr gut auf den Drucker ausgegeben. Leider besitzt das Programm noch einige kleine Mängel, wie beispielsweise das fehlende asynchrone Umtreiben von Text um Grafik. Auch die Texterfassung wird dank TransScript und dem direkten Textimport sehr einfach. Wer allerdings schon eine andere Textverarbeitung besitzt, braucht sich TransScript nicht mehr zuzulegen. Leider hat PageSetterII noch einen Fehler. Manchmal wird der Text nicht mehr umgebrochen. Sollte das passieren, so müssen einfach Trennhilfen angebracht werden.

Hoffentlich wird dieser Fehler bald behoben, er kann nämlich sehr nervend sein. Natürlich hat PageSetterII seine Grenzen. Wer nicht auf PostScriptausgabe, automatischer Trennhilfe, Farbe, Macros und Drehen von Text und Grafik verzichten kann, der sollte direkt mehr investieren und sich ein leistungsfähigeres DTP-Programm zulegen. Wer dazu noch umfangreiche Grafiken erstellen möchte, kommt auch um ein Zeichenprogramm nicht herum. Für DM 298,- sind übrigens noch weitere Fonts erhältlich, die ebenfalls eine sehr gute Qualität bieten. Übrigens: Dieser ganze Bericht ist komplett mit PageSetterII und TransScript erstellt und auf einem NEC P2200 (24-Nadeln) ausgedruckt worden. So können Sie sich ein Bild von den Möglichkeiten dieses Gespanns machen.

AMIGA DOS

Blitzlicht

Name: PageSetterII
Hersteller: GoldDisk
Vertrieb: Fachhandel
Preis: DM 198,-

Positiv:

- Preis-/Leistungsverhältnis
- hervorragende Ausgabe
- direkter Import von/nach TransScript
- gute Fonts

Negativ:

- keine UNDO-Funktion
- Funkt. teilw. eingeschränkt
- fehlerhaftes einbinden der Amiga-Fonts
- keine Unterstützung von Beckertext

AMIGA DOS

Blitzlicht

Name: TransScript
Hersteller: GoldDisk
Vertrieb: Fachhandel
Preis: DM 99,-

Positiv:

- preiswert und schnell
- Macros möglich
- Serienbrieffunktion
- Indexerstellung
- direkter Import nach/von PageSetterII

Negativ:

- kein WYSIWYG
- Trennen/Serienbriefe erst beim Druck
- teilweise unflexibel

AMIGA DOS Info

Wie testet man Desktop-Publishing-Programme?

Wenn neue Programme oder Produkte in der Redaktion der AMIGA DOS ankommen, so ist ihre weitere Bestimmung schon vorgezeichnet. Sie werden auf Herz und Nieren geprüft, und danach wird der Testbericht geschrieben und veröffentlicht. Bei DTP-Programmen ist dies zwar auch gang und

gäbe, aber kann der Leser durch Aufzählen von Details voll über das Programm informiert werden?

Wir dachten uns, daß es vielleicht besser wäre, gleich zu zeigen, was DTP-Programme können, und so entstand dieser Artikel, der komplett mit dem Testkandidaten erstellt wurde. Der Ausdruck entspricht dem eines 24-Nadel-Druckers, eines Gerätes also,

welches heute schon an den meisten Amiga-Arbeitsplätzen zu finden ist. Unser Autor Andreas Polk hat den Artikel in Heimarbeit zusammengesetzt, um ihn so in das Heft zu bringen, wie es seinen Vorstellungen entsprach.

Eventuelle Rechtschreibfehler bitten wir zu entschuldigen; um die Aktualität zu bewahren, mußte das fertige Layout direkt einmontiert werden.

Wir glauben jedoch, daß der Test einer DTP-Software ohne ein Beispiel wesentlich schlimmer ist als ein durch die Eile entstandener Verdreh. Schauen Sie sich die Seiten in aller Ruhe an, und sie werden feststellen, daß sie mit etwas Geduld und Mühe selbst eine professionelle Zeitschrift mit selbstgestellten Layoutseiten sehen lassen kann. (jb)



Für den schnellen Check der Disketten beim Einlegen haben sich Utilities wie »VirusX« aus der Public Domain recht gut bewährt, aber es gibt immer noch mehr Dinge zwischen Bootblock und Startup-Sequence, als sich unsere Schulweisheit träumen läßt.

Weisheit ist jedoch gefragt, denn der Anwender steht nun vor der Qual der Riesenauswahl – das Beste scheint, den Arbeitsspeicher voller Virenkiller zu halten oder den Rechner gleich aus zu lassen. Mit den richtigen Werkzeugen kann man jedoch ein gewisses Maß an Datensicherheit erreichen.

Keine Macht den Viren!

Viruscope besteht aus drei Teilen: Die ersten beiden Teile haben sich dem Kampf gegen die üblichen Bootblock- und Link-Viren verschrieben, der dritte Teil dient zur komfortablen Verwaltung von Bootblöcken. Der erste Teil liest die Bootblöcke von bis zu vier Diskettenlaufwerken und untersucht deren Inhalte auf "populäre" Viren. Wird ein alter Bekannter gefunden, so kann er auf Wunsch von der Diskette verjagt werden. Es werden dabei nicht nur "Klassiker" ihres Genres gefunden, sondern auch "Newcomer" wie der »Lamer Exterminator 3« (hier mit guten Rettungschancen bei einer Früherkennung) oder der »MGM-89«. Auch Link-Viren haben schlechte Chancen, da es ihnen mit dem zweiten Programmteil an den Kragen geht. Disketten und Festplatten können nach den bislang bekannten Vertretern dieser Spezies durchsucht werden, wahlweise in einem Schnelldurchlauf, der nur die wahrscheinlichsten Stellen für eine Infektion untersucht, oder in einem wesentlich zeitaufwendigeren Sicherheitscheck, der alle Dateien eines Laufwerks umfaßt. Dabei werden nicht nur ältere Vertreter wie der »IRQ« aufgespürt, sondern auch neuere Untaten wie »BGS9«, »XENO« oder »Revenge of the Lamer Exterminator«.

Natürlich können nur solche Viren mit Sicherheit erkannt werden, die bei der Entstehung des Programms schon bekannt waren. Der Wahn der Virenprogrammierer bringt je-

Michael Anton

Unter dem Viruscope

Neues Antivirusprogramm im Test

Mit »Viruscope« tritt ein neuer Retter der Disketten vor der Virenpest auf den Plan. Seine Chancen im Kampf gegen den Krampf sind gar nicht so schlecht...

doch öfters "Neuigkeiten" hervor, die anonym bleiben.

Aber nicht jeder unbekannte Bootblock ist gleich ein neuer Virus, es gibt durchaus auch harmlose Vertreter dieser Art. Bei den meisten Spielen steht hier ein Lader, der auf keinen Fall entfernt werden sollte, es gibt auch Bootblöcke, die andere sinnvolle Aufgaben erfüllen. Erkennt Viruscope einen Bootblock mal nicht, so besteht die Möglichkeit, den Inhalt genauer unter die Lupe zu nehmen. Dazu wird der Inhalt nach verdächtigem Code, wie Befehlen zum Schreiben, durchsucht und gegebenenfalls entsprechende Warnungen ausgegeben. Dies geht aber nur bei Bootblock-Viren, vor "neuen" Link-Viren steht auch Viruscope noch machtlos da. Es ist zu hoffen, daß in zukünftigen Versionen auch eine Überwachung von Veränderungen an Dateien möglich sein wird – wohl zur Freude aller Festplattenbesitzer. Viruscope erkennt nicht nur Viren, die sich auf den Datenträgern niedergelassen haben, sondern auch solche, die bereits im Speicher stehen. Die

Überprüfung verbogener Vektoren ist zwar hilfreich, kann jedoch auch falschen Alarm auslösen. Das gilt nicht nur für harmlose resetfeste Programme, wie den Treiber für RAD:, sondern auch für manche Hardware-Erweiterungen: Einige Autoboot-Platten verbiegen ebenfalls etliche Vektoren oder verändern Libraries, so daß solche Warnungen mit Vorsicht genossen werden sollten.

Viruscope verwaltet auch Bootblöcke

Vorbeugen ist besser als Heulen – in diesem Sinne stellt Viruscope auch eine Sammlung von hilfreichen Zusätzen zur Arbeit mit Bootblöcken zur Verfügung. "Wertvolle" Bootblöcke, von kopierschutzgeschützten Spielen beispielsweise, können archiviert werden. Darüber hinaus können Disketten mit praktischen Bootblöcken versehen werden. Eine kleine Sammlung wird gleich mitgeliefert, unter anderem Bootblöcke zum Schutz vor Viren, Abschalten

von Speichererweiterungen oder dem Laufwerkklappen oder zur Bildschirmsteuerung. Eine persönliche Note kann man seinen Disketten ebenfalls geben, da Scroller oder ganze Auswahlmenüs ebenfalls mit diesem Tool erstellt werden können. Mit einem Editor kann sogar direkt an den Bootblöcken gearbeitet werden – wobei natürlich Vorsicht angeraten ist...

Hilft der Blick durchs "Viruscope"?

Die Bedienung von Viruscope ist recht einfach und durch das gute Handbuch und die nett gemachte grafische Oberfläche problemlos. Leider wird dadurch das Programm auch etwas unhandlich, da öfters auf Diskette oder Platte zugegriffen wird, nur um ein Bildchen nachzuladen. Eine Installation auf Festplatte ist problemlos möglich, leider konnte man es sich nicht verkneifen, das Programm in "unsichtbaren" Dateien unterzubringen, was nicht gerade ein feiner Zug ist. Auch kann Viruscope nicht im Hintergrund betrieben werden, wie beispielsweise »VirusX«. Dadurch verliert Viruscope etwas an Universalität, macht dies jedoch durch die umfangreichen Möglichkeiten des Bootblock-Utility wieder wett. An »VirusX« wird wohl auch in Zukunft kein Weg vorbeiführen, »Viruscope« ist jedoch eine wertvolle Ergänzung im Waffenarsenal gegen Viren. Ein Allheilmittel ist es nicht, kann es aber auch nicht sein. (Die Maßnahmen, die hierfür notwendig wären, sollen lieber nicht beschrieben werden...) (jb)



Mit »Viruscope« auf Virensuche

AMIGA DOS Blitzlicht

Name: Viruscope
Vertrieb: Maxon, Schwalbacher Str. 52, 6236 Eschborn
Funktion: Virenprüfer
Preis: 59,- DM

Positiv:

- erkennt alle bekannten Boot- und Link-Viren
- Analyse von Bootblöcken
- umfangreiche Werkzeuge zum Bearbeiten von Bootblöcken
- akzeptabler Preis

Negativ:

- etwas unhandlich
- kein Hintergrundbetrieb
- keine Erkennung völlig neuer Link-Viren

PULSAR

Computer Vertriebs GmbH • Erlanger Str. 8 – 10 • 5 Köln 91 • Tel: (0221) 87 33 59 / 87 36 57 / 87 43 21 • Fax: (0221) 87 41 89

EXKLUSIV – VERTRIEB FÜR DEUTSCHLAND • HÄNDLERANFRAGEN ERWÜNSCHT

Mega Midget	Die 68030 Karte sowohl für den Amiga 500 als auch für den Amiga 2000. Extrem hohe Rechenleistung durch asynchron getakteten 68030 Prozessor. Optional 68882 Coprozessor und 512 KB 32 – Bit RAM. Incl. Turbo Silver (Raytracing Programm)	ab 1498, – DM
Caligari	DAS 3D Konstruktions – und Animationsprogramm der Spitzenklasse. (Professional – Version auf Anfrage) ☆ Alle Befehle (Move, Rotate, Copy usw.) werden in Echtzeit ausgeführt. ☆ Animationen können genauso einfach direkt in 3D erstellt, und in Echtzeit abgespielt werden ☆ Alle Bilder können in Video Auflösung (PAL) berechnet werden und dies bis zu 16.7 Mio. Farben ! ☆ Sehr schneller 3D Editor, mit dem Sie perspektivisch arbeiten.	528, – DM
Vo Rec One	Spracherkennung für den Amiga ! Unterscheidet bis zu 70 verschiedene Wörter und bietet folgende Features: ☆ Alle Amiga Grafikfunktionen können gesteuert werden. ☆ Scroll Bars können gesteuert werden ☆ Digitalisierte Sounds können abgespielt werden. ☆ Zeigt und "liest" Texte ☆ Wird mit deutscher oder englischer Anleitung ausgeliefert ☆ Eigene Software, die nur die VOICE REC LANGUAGE benutzt, kann erstellt werden	328, – DM
Saxon Publisher	Projektion von Text auf Kugeln, Würfeln u.a. Objekten möglich. Der Text läßt sich außerdem problemlos stauchen, strecken und stufenlos rotieren. Textgröße frei wählbar bis 2000 Punkt. Die Ausgabe erfolgt unter Postscript.	799, – DM
Pic Magic	Clipart für alle Amigas. Sämtliche 250 Bilder werden in EPS, Videobitmap und Superbitmap – Format geliefert.	169, – DM
Supercard	Abschaltbare Kopierhardware mit durchgeschliffenem Bus. Erlaubt es jeden Kopierschutz zu kopieren. Für alle Amigas. Für den A2000 auch intern	169, – DM
Mousemaster	Ermöglicht den Anschluß von zwei Geräten am "Maus – Port" (Practical Solution)	65, – DM
Kabellose Maus	Jetzt können Sie Ihren Amiga über 3 m Entfernung mit der Infrarotmaus "fernbedienen" (Practical Solution) Der Anschluß von Modems ist in der BRD verboten.	198, – DM

8 MB Speichererweiterung für A2000 0MB unbestückt *	478, – DM	A1000 SCSI Hard Drive Controller *	579, – DM
8 MB Speichererweiterung für A2000 2MB bestückt *	785, – DM	A1000 40MB Hard Drive Quantum 11 ms. SCSI Contr. *	1798, – DM
8 MB Speichererweiterung für A2000 4MB bestückt *	1185, – DM	A1000 80MB Hard Drive Quantum 11 ms. SCSI Contr. *	2398, – DM
8 MB Speichererweiterung für A2000 6MB bestückt *	1635, – DM	A2000 40MB Hard Card Quantum 11 ms. SCSI Contr. *	1449, – DM
8 MB Speichererweiterung für A2000 8MB bestückt *	1949, – DM	A2000 80MB Hard Card Quantum 11 ms. SCSI Contr. *	2198, – DM
Insider II 1.5MB Speichererweiterung für A1000 unbestückt	525, – DM	A2000 105MB Hard Card Quantum 11 ms. SCSI Contr. *	2449, – DM
Insider II 1.5MB Speichererweiterung für A1000 1.5MB	874, – DM	A500 40MB Hard Drive Quantum 11 ms. SCSI Contr. *	1698, – DM
Baseboard 4MB Ram für A500 intern unbestückt	329, – DM	A500 80MB Hard Drive Quantum 11 ms. SCSI Contr. *	2298, – DM
U.S. Robotics HST 14.400 Modem	1489, – DM	44MB Wechselplatte Syquest mit Cartridge.	1998, – DM
2400 Baud Modem extern *	298, – DM	44MB Syquest Cartridge	268, – DM
2400 Baud Modem intern für A2000 *	306, – DM	Mouse Stick (programmierbarer Joystick – Gravis)	146, – DM
		Gravis Joystick	79, – DM

Alle mit * gekennzeichneten Artikel sind von SUPRA.

Bilder wie im Original!

In Sekundenbruchteilen von Video und Kamera eingelesen.

SNAPSHOT!

VIDEO-DIGITIZER

... unentbehrlich bei:

- Grafik
- Animation
- Desktop Video
- Präsentation



Herstellung und Vertrieb
sowie kostenlose
Information bei:

**VIDEOTECHNIK
DIEZEMANN**

Dammstraße 42 • D-2300 Kiel 1 • Telefon (0431) 9 44 24 • Fax 9 24 32

Festplatten sind ein beliebtes Speichermedium, wenn es darum geht, große Mengen von Daten schnell zur Verfügung zu haben. Meistens ist die Übertragungsgeschwindigkeit eine Frage des Preises. Wer keine allzu großen Ansprüche stellt, kann eine 40-MByte-Festplatte für zirka 1200 DM einsetzen und bekommt dafür eine Übertragungsrate von 300 bis 450 kByte pro Sekunde geboten. Werden jedoch höhere Geschwindigkeiten oder größerer Speicherplatz benötigt, wie es zum Beispiel bei Werbepräsentationen nötig ist, ist ein tiefer Griff in den Geldbeutel unumgänglich. In diese Lücke springt die AT-Bus-Festplatte der Firma FSE, die zu einem günstigen Preis wahre Wunder verspricht.

Rekordsprinter

Die zum Test vorliegende Platte wird über ein Interface angesprochen, das dem 16-Bit-Bus des IBM AT entspricht. Dies verspricht mit der eingebauten Quantum-Harddisk eine mittlere Zugriffszeit von 19 ms (die Zugriffszeit sagt aus, wie lange der Schreib-/Lesekopf braucht, um einen Datenblock zu erreichen). Mit Hilfe eines 64 kByte großen Cache-Speichers (eine Art Pufferspeicher) wird eine Datentransferrate von 750 kByte/s erreicht. Leider gilt dieser Wert nicht für das AmigaDOS. Hier wurde im Test "nur" ein Wert von 620 kByte/s ermittelt. Dies ist immer noch eine geradezu atemberaubende Übertragungszeit! Doch trotz dieser Schwerarbeit läßt die Platte kein Ächzen hören. Die Harddisk ist sehr leise, und auch im Dauerbetrieb ist vom Lüfter kaum mehr als ein gedämpftes Brummen zu vernehmen.

Äußerlichkeiten

Vom Design her macht die AT-Bus-Festplatte eher einen schlichten Eindruck. Ein amigafarbenes Metallgehäuse, das sich als Monitoruntersatz eignet, verbirgt sämtliche technischen Meisterleistungen. Nur eine Power-LED und eine Funktions-LED zeugen davon, daß sich etwas drinnen befinden muß. Im Gehäuse ist viel Platz, und eine Möglichkeit zur Aufnahme einer weiteren Harddisk ist bereits vorgesehen. Späteren Erweiterungen steht also nichts im Wege. Etwas zu kurz geraten ist das Anschlußkabel, das sich in Form eines Flachbandkabels präsentiert. In der Serienversion wird dieses Kabel durch ein 80 cm langes, abgeschirmtes Rundkabel ersetzt. Die Montage der Festplatte gestaltet sich denkbar einfach.

Schneller, höher, weiter!

Eine AT-Bus-Festplatte für den Amiga 500

Die Firma FSE hat eine Festplatte konstruiert, die über ein 16-Bit-Interface am Amiga 500 betrieben wird. Als Treiber-Software wird ein neuartiges modulares Konzept verwendet, das keine Wünsche offen läßt und sich problemlos an jede Art externen Speichers anpassen läßt. Die Hardware besteht aus einer Quantum-Festplatte. Die sich dadurch ergebenden Vorteile sind außergewöhnlich. Wir hatten die Möglichkeit, ein Vorabmodell für Sie zu testen.

ungen steht also nichts im Wege. Etwas zu kurz geraten ist das Anschlußkabel, das sich in Form eines Flachbandkabels präsentiert. In der Serienversion wird dieses Kabel durch ein 80 cm langes, abgeschirmtes Rundkabel ersetzt.

Die Montage der Festplatte gestaltet sich denkbar einfach.

Aufwärm lauf

Es wird lediglich die Abdeckklappe zum externen Bus des Amiga 500 entfernt, die sich an der linken Gehäuseseite befindet. Hier wird der Bustreiber aufgesteckt, der den Amiga 500 um zirka 10 cm verbreitert. Jetzt muß nur noch eine freie Steckdose für die Stromversorgung der Platte gesucht werden, und schon ist die Harddisk einsatzbereit.

Installiert wird die Festplatte mit der mitgelieferten Installations-Software. Auch hier treten keine Probleme auf, da die mitgelieferte Dokumentation auch unerfahrenen Anwendern auf verständliche Art die Funktion und Auswirkung sämtlicher Tools deutlich macht. Als Treiber-Software wird »BOIL! 3« eingesetzt, das sich allen Anforderungen gewachsen zeigte. »BOIL! 3« ist ein Festplattentreiber von etwas ungewöhnlicher Bauart.

Es handelt sich dabei um eine Art Software-Interface, das in etwa mit den amigaspezifischen Devices zur Steuerung externer Geräte (»Printer.device«, »Serial.device«) vergleichbar ist. Dieser Treiber verfügt über eine modulare Struktur, die es ermöglicht, jede Art von Festplatte und andere externe Speichermedien

individuell anzupassen. Die Hauptanlaufstelle ist das »BOIL.device«. Dieses Device nimmt alle Aufträge entgegen und verteilt sie auf die maximal 800 möglichen Laufwerke. Die eingehenden Befehle werden in Hardware-unabhängige Standardkommandos umgesetzt. Dadurch ist es möglich, mehrere Controller-Typen zu verwalten, die normalerweise über sehr unterschiedliche Befehlssätze verfügen.

Das Ansprechen der Hardware wird nun von einem der beliebigen »Nucleus.devices« übernommen. Das »Nucleus.device« nimmt die standardisierten Kommandos entgegen und übersetzt diese in Hardware-spezifische Controller-Kommandos. Ein einzelnes »Nucleus.device« ist lediglich in der Lage, diese Controller-Kommandos auszuführen und kann infolge dessen sehr kompakt gehalten werden. Für jedes externe Gerät wird also ein spezifisches »Nucleus.device« benötigt. Geräte gleicher Bauart können jedoch von ein und demselben »Nucleus.device« angesprochen werden.

Das »BOIL.device« und ein einzelnes »Nucleus.device« reichen schon aus, um ein spezielles Speichermedium anzusteuern. Unter AmigaDOS wird jedoch zusätzlich »BOILMount« verwendet. »BOILMount« übernimmt die Anmeldung der einzelnen Partitionen und die Vorbereitung zum Reboot beziehungsweise Autoboot (unter Kickstart 1.3). Die auf der Platte stehenden Informationsdaten werden analysiert und so aufbereitet, daß sie von AmigaDOS verarbeitet werden können. Somit entfällt das Erstellen einer Mountlist für unter »BOIL! 3« laufende Geräte. »BOILMount« erlaubt somit ein sehr differenziertes Bootverhalten der einzelnen Geräte und sogar einen Passwortschutz, entweder von ganzen Platten oder auch nur spezieller Partitionen.

Doping gestattet!

Auf den Installationsdisketten befinden sich auch einige sehr gut durchdachte Tools, die es auch dem unerfahrenen User ermöglichen, seine Harddisk problemlos zu verwalten.

Als erstes ist das Programm »BOILInstall« zu erwähnen. Hiermit kann die Platte Low-

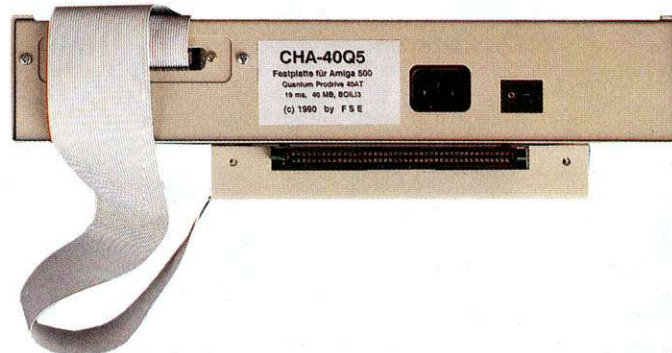


Bild 1. Die AT-Bus-Platte präsentiert sich in schlichtem Design...

Level-formatiert werden, das heißt, die Festplatte wird physikalisch formatiert. Hierbei werden alle auf der Platte vorhandenen Daten endgültig und unwiderruflich gelöscht, es ist also mit Vorsicht anzuwenden. »BOILInstall« übernimmt auch das Partitionieren der Harddisk. Partitionieren bedeutet, daß die Platte in mehrere logische Geräte (zum Beispiel DH0:, DH1:, DH2:, etc.) eingeteilt wird. Dies kann aus Gründen der Übersichtlichkeit sehr sinnvoll sein. Wer jedoch über einen geringen RAM-Speicher verfügt, sollte bedenken, daß jede Partition (genau wie zusätzliche Diskettenlaufwerke) zirka 30 kByte Speicher verbraucht. Wieviel Partitionen nun eingerichtet werden, richtet sich nach dem Geschmack des Besitzers beziehungsweise dem vorhandenen Rechnerspeicher. Beim Einrichten der Partitionen sollte unbedingt das »Fast-File-System« verwendet werden, da das Standard-Filesystem relativ langsam arbeitet und die maximale Partitionsgröße auf 51 MByte begrenzt ist. Mit »BOILInstall« läßt sich auch die Größe des Cache-Speichers manipulieren. Wer über wenig Rechnerspeicher verfügt, kann den Cache-Speicher weglassen. Dies wirkt sich jedoch nachteilig auf die Übertragungsrate aus; die Platte wird folglich langsamer.

Das zweite Programm, »BOILPass«, kann ebenfalls zum Belegen von Partitionen verwendet werden. Sein Haupteinsatzgebiet ist jedoch das nachträgliche Ändern beziehungsweise Einrichten von Passwörtern. Hier ist es möglich, ganze Platten oder einzelne Partitionen vollständig zu sperren oder aber nur Schreib- und Lesezugriffe einzuschränken.

»BOILCheckDrive« dient zur Überprüfung der Festplatte auf physikalische Fehler. Es existiert keine Festplatte, die nicht irgendwo einen oder mehrere physikalische Fehler vorweisen kann. Diese Fehler werden meist mit einem Aufkleber auf dem Festplattengehäuse kenntlich gemacht und dem System mitgeteilt, indem sie in einer Fehlerliste eingetragen werden. Für fehlerhafte Spuren muß nun eine Alternativspur gesucht werden. Bei der vorliegenden Quantum-AT-Platte entfällt dies jedoch ganz, da dieser Plattentyp seine Alternativspuren selbst

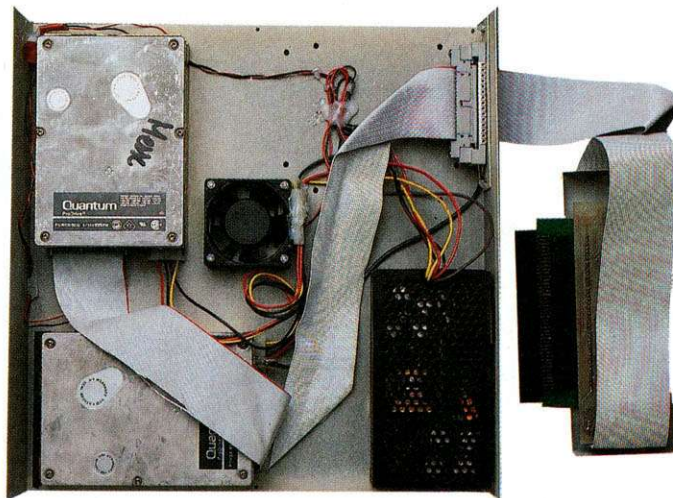


Bild 2. ...aber sie hat es in sich!

einrichtet, sofern dies noch möglich ist.

»BOILWhy« ist ein nützliches Hilfsprogramm, das die Fehlererkennung sehr erleichtert. Meldet sich der Rechner einmal mit »Volume XYZ has a read/write error«, so kann mit »BOILWhy« einfach festgestellt werden, was denn da nun schiefgegangen ist.

»BOILExists« dient zur Steuerung von Batch-Dateien (wie zum Beispiel der »Startup-Sequence«). Es ähnelt dem »IF EXISTS« des AmigaDOS und gibt einen Fehlercode zurück, wenn das »BOIL.device« nicht gefunden wurde.

Fliegender Start

Mit dem Programm »BOIL-Specials« können nun Autopark und die Transferraten unabhängig von »BOILInstall« eingestellt werden. Autopark bedeutet, daß die Schreib-/Leseköpfe nach einer bestimmten Zeit, in der kein Zugriff auf die Platten erfolgt, in eine Position gebracht werden, die sie von den Platten entfernt. Dies ist zum Beispiel unbedingt notwendig, wenn die Platte transportiert werden soll. Die »Flughöhe« der Schreib-/Leseköpfe auf der Platte beträgt wenige tausendstel Millimeter. Wenn die Köpfe über der nicht mehr in Drehung befindlichen Platte schweben und infolge einer Erschütterung die Platte berühren, ist eine physikalische Beschädigung der Platte unausweichlich. Dieser Vorgang, der mit dem Verlust meist wichtiger Daten verbunden ist, wird im Fachjargon »Head-crash« genannt und trifft den

Nagel im wahrsten Sinn des Wortes auf den (Schreib-/Lese-)Kopf.

Selbstverständlich ist die AT-Bus-Platte autobootfähig ab Kickstart 1.3 oder bei Kickstart 1.2 mit einem Autoboot-Controller. Ist die Harddisk erst einmal installiert, kann der Rechner zukünftig von der Platte aus gebootet werden.

Erwähnenswert ist auch, daß nach einem Reset oder Kaltstart der Autobootvorgang durch Druck auf die Taste [Help] unterbunden werden kann. Dies ist zum Beispiel nötig, wenn eine Diskette gebootet wird und der Verdacht auf Viren besteht oder ein Programm gestartet wird, das eine Festplatte »nicht mag« (einige wenige Spiele zum Beispiel). Über diese Taste kann auch festgelegt werden, ob beim Booten der Platte eine Passwortabfrage stattfinden soll. Eine solche Abfrage verhindert, daß sich Unbefugte an der Harddisk vergreifen können. Dies setzt jedoch voraus, daß das Passwort nicht allgemein bekannt gemacht wird.

Innere Werte

Die Festplatte selbst besteht aus einer Quantum-Harddisk vom Type Prodrive 40AT mit 40 MByte, die bereits unter IBM-Rechnern ihre Zuverlässigkeit bewiesen hat. Die Platte hat einen Durchmesser von 3,5 Zoll, 3 Schreib-/Leseköpfe und 2 Disks. Die Schreibdichte der einzelnen Disks liegt bei 1000 tpi (Tracks per Inch, also Spuren auf ein Inch Diskbreite). Laut Quantum widersteht die Harddisk einem Schlag

bis zu 60G, das heißt, dem sechzigfachen der Erdbeschleunigung. Es wird jedoch angeraten, dies nicht nachzuprüfen. Anstelle der Quantum 40AT-Harddisk kann auch eine Quantum 80AT verwendet werden, die dann 80 MByte zur Verfügung stellt. Dank des noch vorhandenen Platzes im Gehäuse kann das System bis auf 160 MByte erweitert werden, was auch den speichererschleißendsten Festplattenbesitzer zufriedenstellen dürfte.

Die Interface-Hardware und die Platte befinden sich in einer baulichen Einheit. Dadurch kann die Harddisk optimal an den Rechner angepaßt werden. Die Elektronik des Interfaces und der Plattenansteuerung sind selbst bei der vorliegenden Testversion sehr sauber gearbeitet.

Die AT-Bus-Festplatte der Firma FSE stellt ein gut durchdachtes Produkt dar, das sich technisch auf dem neusten Stand befindet. Gute Konzeption, solide Verarbeitung und durchdachte Software sind hier zu einem fairen Preis erhältlich. Laut Angaben des Herstellers wird dieser wahrscheinlich noch fallen. Die Ausbaufähigkeit des Systems läßt auch bei späteren Erweiterungen alle Möglichkeiten offen. Wünschenswert wäre nun noch diese Festplatte mit eingebauter Speichererweiterung, um den knappen Platz im Amiga 500-Gehäuse etwas zu entlasten.

(tb)

AMIGA DOS Blitzlicht

Name: AT-Bus Festplatte mit BOIL! 3-Treiber
Hersteller: FSE Frank Strauß Elektronik
 Schmiedstraße 11
 6750 Kaiserslautern
Preis: 42 Byte - 1498,- DM;
 84 MByte - 1898,- DM

Positiv:

- leichter Einbau
- schneller Zugriff
- Gehäuse als Monitor-stander
- durchdachte Installations-Software
- beliebig erweiterbar

Negativ:

- nichts Negatives anzumerken

Wer jetzt noch einen Atari ST-Emulator benötigt, braucht ab sofort mit seinem Wunsch nicht mehr hinter dem Berg zu halten. Einige Lösungen im Bereich Emulation von Fremdrechnern sind ja recht brauchbar, wie beispielsweise das PC-Board, während C64-Emulationen doch recht fragwürdig sind. Wie wird sich also der neue Atari-ST-Emulator einführen?

Handelt es sich hier wirklich um eine brauchbare Version oder um eine Spielerei? Wenn man die Verpackung öffnet, ist man erst einmal erstaunt; nicht eine große Steckkarte strahlt einen an, sondern ein gerade mal 2 Zentimeter hohes Kärtchen, auf dem sich zwei PAL-Bausteine (programmierbare Logikschaltkreise) befinden. Diese beiden PALs sind für die Kommunikation mit der angeschlossenen Hardware zuständig.

Der eigentliche Emulator befindet sich in Form eines Programms auf der mitgelieferten Amiga-Diskette. Zuerst muß das System also installiert werden. Dazu wird die Karte in einen der freien Slots gesteckt und die Diskette gebootet. Nach kurzer Zeit wird der Screen grau, und man muß eine TOS-Diskette ins Laufwerk legen. Übrigens: Das TOS (DOS des Atari ST) wird nicht mitgeliefert, es muß für ein paar Mark im Laden gekauft werden.

Nach kurzer Ladezeit erscheint der Desktop (Arbeits-tisch des Atari ST) auf dem Bildschirm. Um die hohe monochrome Auflösung des ST nutzen zu können, arbeitet der Emulator im Interlace-Modus. Das hat natürlich ein merkliches Flimmern zur Folge. Um dieses Flimmern etwas zu mildern, wird ein "Accessory" mitgeliefert. Dieses besteht aus Programmen, die von jedem anderen Programm aus aufgerufen werden können. So hat der Atari-Anwender trotz fehlendem Multitasking dennoch das Gefühl, daß zwei Programme parallel ablaufen. Mit diesen Accessory kann der Anwender die Anzahl der Scanlines herabsetzen. Das hat zur Folge, daß der Elektronenstrahl des Monitors nicht mehr so viele Zeilen darstellen muß und somit die Bildwiederhol-frequenz heraufgesetzt werden kann. Durch eine Tastenkombination kann dann zwischen den beiden Werten umge-

Verwandlungs-künstler

Der Amiga-ST-Emulator

Mittlerweile scheint es ja schon Mode zu sein, Emulatoren für den Amiga herauszubringen. Angefangen beim PC-Board, welches eigentlich kaum noch Emulator genannt werden kann, da sich auf der Erweiterungskarte ein ganzer PC befindet, geht es weiter bis hin zum C64-Emulator und dem Macintosh-Emulator AMax.

schaltet werden – und siehe da, das Flackern ist zwar noch nicht verschwunden, hat sich jedoch deutlich verringert.

Mit diesem Accessory kann aber noch mehr eingestellt werden. So kann der Anwender festlegen, welche Amiga-Laufwerke in der Emulation zum entsprechenden Atari-Laufwerk "umgewandelt" werden sollen. Dies ist jedoch nicht alles. Wie eingangs schon erwähnt, kann durch die mitgelieferte Steckkarte die gesamte Hardware mitbenutzt werden. Sind Sie im Besitz einer Festplatte, so können Sie sich eine Partition für den Atari reservieren. Die nötigen Programme werden für den ST mitgeliefert. Sind Sie außerdem im Besitz einer RAM-Erweiterung, so können Sie diese auch mit dem Atari nutzen. Natürlich können auch Drucker oder ein eventuell angeschlossenes MIDI-Interface mitbenutzt werden. Wer also seinen Amiga schon um einiges erweitert hat, braucht seinen "Pseudo-Atari" nicht mehr auszubauen.

Auch wer seine Festplatte nicht mit dem Atari teilen möchte, kann dennoch Nutzen von ihr ziehen. Installiert man nämlich den Emulator auf der Festplatte, so kann man mit dem mitgelieferten Installationsprogramm eine TOS-Version erstellen, die auf der Amiga-Platte abgespeichert wird. Startet man dann den Emulator von der Platte, verhält sich der Atari so, als hätte er ein ROM-TOS eingebaut. So muß nicht eine Bootdiskette eingelegt werden, sondern nach zirka 40 Sekunden lädt der Emulator das TOS automatisch. Allerdings muß man immer erst die Zeit abwarten, was jedoch systembedingt ist und nicht dem Emulator zugesprochen werden kann.

Mit dem Utility können noch einige andere Einstellungen gemacht werden. So läßt sich hier schon auf Amiga-Seite festlegen, wie viele Laufwerke der Atari benutzen soll. Diese Einstellungen werden direkt im Emulationsprogramm gespeichert. So kann der An-

wender mehrere verschiedene Konfigurationen erstellen und muß bei Bedarf nur das entsprechende Icon anklicken. Das Handbuch ist zwar sehr knapp gehalten, reicht aber aus. Neben einigen kleinen Mängeln (die Tastenkombination für einen Reset ist übrigens [Ctrl] + [Alt] + [1], [2] oder [3]) erwähnt es auch nicht, was die Info-Files bedeuten, ohne die der Emulator nicht startbar ist. Die kurze Einführung in die Arbeit mit dem Atari ST ist allerdings sehr lobenswert. Auf der Atari-Diskette befindet sich übrigens noch eine RAM-Disk, die der Anwender installieren kann.

Kommen wir nun zu der Emulation selbst. Auch hier setzt sich das positive Bild fort, das wir von dem Emulator haben. Der Anwender wird kaum Geschwindigkeitsnachteile gegenüber einem "normalen" Atari bemerken. Allerdings ist der Bildschirmaufbau manchmal etwas träge. So öffnen sich die Fenster gegenüber einem normalen Atari doch etwas langsamer. Trotzdem läßt sich mit dem Emulator sehr gut arbeiten. Die schnelle Emulation beruht auch darauf, daß während der Emulation kein Multitasking möglich ist. Um wieder zum Amiga zurückzukehren, ist immer ein Reset möglich. An einer neuen Version, bei der dann der ST auf einem besonderen Screen emuliert wird, wird aber gearbeitet. Hier wird übrigens ein kostenloser Update-Service, der genau ein Jahr dauert, angeboten. Das sollten sich andere Firmen mal zum Vorbild machen.

Ein weiteres wichtiges Thema ist natürlich die Kompatibilität, also die Frage, wie sich der Emulator mit der Software verträgt. Hier kann man eine ganz klare Antwort geben. Handelt es sich bei einem Programm um kein kopiergeschütztes, welches direkt auf die Hardware zugreift, so ist es mit großer Wahrscheinlichkeit lauffähig. So liefen die von mir getesteten Anwenderprogramme wie »Beckertext«, »Stad«, »Calamus« und »BeckerCAD« einwandfrei. Man konnte sehr gut mit ihnen arbeiten. Allerdings muß man hier einige Einschränkungen hinnehmen. Müssen geschützte Programme erst installiert werden, wie beispielsweise Beckertext, so muß diese Installation auf einem originalen ST gesche-



Bild 1. Der Atari ST-Emulator – Platine und Diskette

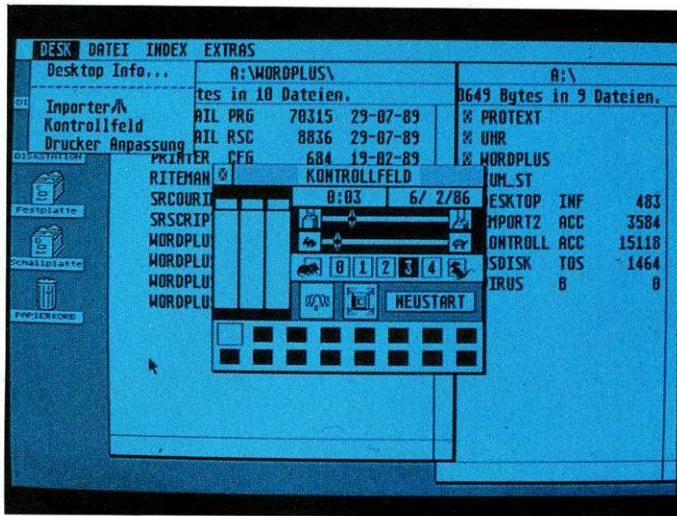


Bild 2. Die meisten Anwendungen für den ST laufen problemlos

hen, da hierbei direkt auf die Hardware zugegriffen wird. Programme, bei denen dies nicht der Fall ist, können natürlich auch problemlos auf dem Emulator installiert werden. Im Zweifelsfall können Sie die Software sicherlich bei einem Freund oder direkt beim Händler installieren.

Ist der Amiga nun ein ST oder nicht?

Anders sieht es bei Spielen aus. Der Emulator kann zwar auch in den Farbmodus umgeschaltet werden, allerdings ist die Chance, daß ein Spiel läuft, sehr gering. Dies ist unserer Meinung nach aber auch kein großer Nachteil, da der Amiga sowieso die besseren Spielmöglichkeiten bietet. Man wird sich kaum dem Emulator zulegen, um Spiele auf ihm laufen zu lassen.

Probleme gab es auch mit Zweitemulatoren, wie beispielsweise »PC-Ditto«. Hier zeigt die Emulation ihre Grenzen.

Der Atari ST-Emulator ist wirklich gelungen. Die Kombination aus Hard- und Software ist sehr sinnvoll. Auch die Ausnutzung der Amiga-Hardware ist vorbildlich. Probleme gab es nur beim Erstellen einer Hardcopy. Hier war dem Drucker nur Müll zu entlocken. Allerdings wurde der Drucker sonst gut genutzt. Sind Sie im Besitz eines 68020/30, so müssen Sie vorerst auf den ST verzichten. Die aktuelle Version des TOS, TOS 1.4, ist mit diesen Prozessoren nicht lauffähig. Hier muß auf das TOS 1.6 gewartet werden, welches dann auch mit diesen Prozessoren "fertig wird".

Somit ergibt sich ein klarer Anwendungsbereich für den Emulator. Diejenigen, die die guten Textverarbeitungen und MIDI-Programme des Atari benutzen möchten, machen mit "Medusa" einen guten Kauf. Wer allerdings auf dem Emulator spielen möchte, der hat ihn umsonst gekauft, denn dafür ist er nicht gebaut worden. Übrigens, wer möglichst schnell mit dem Atari arbeiten möchte, dem sei hier das Atari-ST-Handbuch von Data Becker empfohlen. Es erklärt nicht nur die Arbeit mit dem Desktop, sondern geht auch ausführlich auf die vorhandene Soft- und Hardware ein. Somit ist das Buch für den Umsteiger sehr gut geeignet.

(jb)

AMIGA DOS Blitzlicht

Name: Medusa
Anbieter: 3-State Computer Technik, Schaumburgstraße 17, 4350 Recklinghausen, Tel.: 02361/16207
Preis: 498,- DM

Positiv:

- fast so schnell wie ein original Atari
- gute Unterstützung der vorhandenen Hardware
- hohe Bildqualität
- alle Bildschirmodi werden unterstützt

Negativ:

- Probleme bei Programmen mit Kopierschutz
- Dokumentation teilweise unzureichend

Festplatten

in 16 Bit Technologie mit Boil3 exklusiv bei

FSE von **Quantum**

für A500/1000 :

AutoBoot, 2 Jahre Garantie, BOIL3, 750² KB/S, nach Diskperf 620 KB/S 19¹ ms, sehr leise und zuverlässige

Quantum - Festplatten

42 MB 1498.- 84 MB 1898.- 210³ MB 2998.-

Die Preialternative :

66 MB 1398.-

28¹ ms, 450² KByte/S, 1 Jahr Garantie, 8 Bit, OMTI - Controller, AutoBoot, BOIL3, durchgeführter Bus, NEC

SCSI für A2000

Flexibler SCSI - Controller mit Anschlußmöglichkeiten für Festplatten, Wechselplatten, Streamer, Laserdrucker usw. Leistungsdaten wie bei 16 Bit Interface; siehe oben

Filecards Quantum - Festplatten mit 2 J. Garantie : 42 MB 1398.- 105 MB 1998.- 210³ MB 2898.-

Wechselplatte SyQuest SQ555, inkl. Medium : 44 MB, 25¹ ms 1798.- Medium SQ 400, 44 MB 198.-

1) Herstellerangabe 2) Datentransferrate auf dem Bus 3) Bei Schwierigkeiten von Quantum, können auch Rodime HDDs geliefert werden.

Diskettenlaufwerke

Vollkompatibel, anschlussfertig, abschaltbar, inkl. durchgeführtem Bus, amigafarbenem Metallgehäuse, 880 KB, beige Frontblende. Wir verwenden nur Markenlaufwerke von **TEAC**. Bei allen 5.25" Stationen 40/80 Track schaltbar. Wir gewähren 1 Jahr Garantie auf alle **TEAC** Diskettenlaufwerke. Mit durchgeführtem Bus :

3.5": 198.-

5.25": 228.-

Neu : BOIL3
Der ultimative Festplattentreiber, lauffähig mit AT-Bus- und OMTI.

DM 98.-

Frank Strauß Elektronik
Schmiedstraße 11
6750 Kaiserslautern
Tel.: (0631) 67096 - 98
Fax: 60697

Lieferung erfolgt ab Lager mit UPS oder Post per Nachnahme. Donnerstags bis 20.30 h geöffnet.

Ottmar Röhrig

Die Schatzkiste

Public Domain für den Amiga



Auf kaum einem Computertyp hat die Public-Domain-Szene einen solch großen Stellenwert wie auf dem Amiga. Grund genug für uns, Sie mit einem Schwerpunkt zu diesem Thema umfassend zu informieren.

Die Idee der Public-Domain-Software – auch kurz PD genannt – ist noch relativ neu und hat erst in den letzten Jahren einen gewissen Ruhm erlangt. Daß PD-Programme frei kopiert und benutzt werden dürfen, das ist wohl der Hauptunterschied zu „anderer“, der sogenannten kommerziellen Software. Doch was ist PD eigent-

lich genau? Viele reden zwar davon, doch an eindeutigen Definitionen mangelt es bis jetzt. Beginnen wir also ganz am Anfang, bei der Erklärung des Begriffs Public Domain. Das Schwierige bei diesem Unterfangen besteht darin, daß es keinerlei rechtliche oder gar gesetzliche Erklärungen zu diesem Thema gibt. Wie sollte man auch von Politikern verlangen, daß sie sich

mit der PD-Software und deren Urheberrechten befassen, wenn es bis heute nicht einmal zu einer offiziellen Aufnahme von allgemeinen Computer-Programmen (Software) in das Urheberrechtsschutzgesetz gelangt hat. Zwar gibt es einige Grundsatzurteile, die Software ebenso als „schützenswert“ bezeichnen, wie beispielsweise Musikstücke oder schriftliche Ar-

beiten, doch sind all dies eben Urteile bestimmter Gerichte, nicht jedoch in einem Gesetzestext verankerte Rechte beziehungsweise Pflichten.

Wirrwarr der Begriffe

Doch soweit nur ein kurzer Ausflug. Wenden wir uns wieder der PD zu. Wie bereits ge-



auf einer der PD-Reihen (Ansammlungen zahlreicher PD-Programme auf katalogisierten Disketten) läßt sich problemlos in eine dieser drei Kategorien einordnen, wenn auch in letzter Zeit ein neuer Begriff namens "Tryware" aufgetaucht ist.

Was ist eigentlich was?

Das Verwunderliche dabei ist jedoch, daß lediglich ein Bruchteil der PD-Programme wirklich Public Domain ist. Um weitere Verwirrungen zu vermeiden, möchte ich ab nun die Abkürzung "PD" als Vertreter für den erwähnten Oberbegriff, das ausgeschriebene "Public Domain" jedoch als Spezifizierung einer der drei erwähnten Gruppen verwenden. Deklariert der Autor eines Programms sein digitales Werk nämlich als Public Domain, so verliert er sämtliche Rechte an diesem Programm. Das heißt dann nicht nur, daß Sie als Benutzer es sooft kopieren und verwenden können, wie's beliebt. Vielmehr ist auch nichts dagegen einzuwenden, wenn Sie ein solches Programm auf eine Diskette kopieren und für einen immensen Preis verkaufen würden. Sie können also mit einem einwandfrei als Public Domain deklarierten Programm erlaubterweise Gewinn mit dem geistigen Eigentum anderer machen. Gott sei Dank ist so etwas jedoch selten. Eines der wenigen, laut Dokumentation wirklichen Public-Domain-Programme ist beispielsweise "SetFont" von Dave Haynie, das in der Version 2.5 auf Fish-Disk 182 erschienen ist. Dieses Programm dient dem Austausch des Systemzeichensatzes und ist in mehreren, ständig verbesserten Ausführungen erschienen.

Die Sache mit den eigenen Rechten

Doch was ist dann der wohl größte Teil der PD-Software? Hier kommt der zweite Begriff ins Gespräch, die sogenannte "Freeware". Freeware sind Programme, die Sie zwar kostenlos weitergeben und benutzen dürfen, doch behält sich der Programmautor sämtliche Rechte vor. So wäre es denkbar, daß er nach einiger Zeit sein Programm dem PD-Pool (so der Fachbegriff für

die Ansammlung aller Public Domain-, Freeware-, Shareware- und Tryware-Programme) wieder entnimmt und das Programm – auch in vollkommen unveränderter Form – kommerziell vertreibt. Auch wenn Sie ein solches Programm schon seit langer Zeit benutzen, wären Sie in diesem Moment nicht mehr berechtigt, das Programm weiterhin kostenlos zu verwenden, geschweige denn weiterzukopieren.

Natürlich ist das nur ein hypothetisches Beispiel (mir ist zumindest ein solcher Fall noch nicht zu Ohren gekommen), doch kann diese leichte Übertreibung lediglich das Verständnis für die Freeware festigen. Der Autor eines Freeware-Programms behält sich also insbesondere das Copyright der Software vor, kann den "Status" und die Vertriebsbedingungen derselben also jederzeit ändern.

Mit der Freeware ist also ein recht guter Begriff für den Großteil der PD-Programme gefunden. Dabei bezieht sich alles hier besprochene nicht nur auf PD-Software für den Amiga. Auch für MS-DOS-, Macintosh- oder Apple-II-Rechner gibt es genügend Software, die sich in diese zwei Kategorien einordnen läßt.

Vor zirka zwei Jahren kam dann eine Bewegung auf, in der Programmautoren versuchten, einerseits dem Grundgedanken der PD-Software (der freien Kopierbarkeit und der Kostengünstigkeit) treu zu bleiben, andererseits jedoch mit ihrer geistigen Leistung auch Geld zu verdienen. Daraus entwickelte sich eine Software-Art, die wir heute als Shareware bezeichnen.

Ein kleiner Obulus tut's auch

Shareware ist im eigentlichen Sinne (auch im Sinne des Oberbegriffs) keine PD mehr, sondern kommerzielle Software mit ganz speziellen Vertriebsbedingungen. Der Grund, warum Shareware auch auf PD-Reihen vertrieben wird, ist allein darin zu suchen, daß die Software auf diesem Wege eine große Zahl von Computerbenutzern erreicht, die ansonsten keinerlei Kenntnis von solchen Programmen erlangen würden. In letzter Zeit hat sich – vor allem durch die

allen Programmen beiliegenden Disclaimer (doch dazu später mehr) folgende Definition durchgesetzt: Shareware sind Programme, die zwar frei kopiert und weitergegeben werden dürfen. Sollten Sie jedoch als Benutzer nach einiger Zeit der Verwendung und Prüfung des Programms davon ausgehen, daß Sie es gebrauchen können, so sind Sie verpflichtet(!), einen angemessenen Geldbetrag an den Programmautor zu überweisen.

Wie hoch dieser Betrag ist, wird meistens in den Dokumentationsdateien zum jeweiligen Programm angegeben. Ist das nicht der Fall, so überweisen Sie, was Sie für ein solches Programm auch an einer Ladentheke bezahlen würden und als vernünftig empfinden. Im Gegensatz zu einer weitverbreiteten Meinung ist die Bezahlung dieses Betrages Pflicht(!), also keine wohlwollende Geste. Bezahlen Sie für ein solches Programm nichts und verwenden es weiterhin, so machen Sie sich moralisch und rechtlich (siehe dazu oben) ebenso strafbar, wie Sie es bei der Verwendung von Raubkopien "echter" kommerzieller Software tun.

Update-Service inbegriffen

Als Vorteil einer Bezahlung des Programms ist die offizielle Registrierung bei dem Programmautor gegeben. Neben der neuesten Version des Programms erhalten Sie nach der Registrierung meist auch noch ein gedrucktes Handbuch und den Service, daß der Autor Sie bei der Erscheinung neuer Versionen oder Programme sofort informiert. Der Autor eines Shareware-Programms behält sich somit alle Rechte an seinem Werk vor, genauso, wie das bei kommerzieller Software der Fall ist. Lediglich das Kopieren und Weitergeben, sowie das Ausprobieren (Verwenden) des Programms für einen angemessenen Zeitraum sind erlaubt, was die Shareware auch eindeutig von kommerzieller, über die Ladentheke verkaufter Software unterscheidet.

Ausprobieren ist angesagt

Als letztes Mitglied hat sich seit noch kürzerer Zeit (etwa einem Jahr) die Tryware den

sagt, ist eine Begriffsdefinition des Pudels Kern. PD ist nämlich einmal eine ganz bestimmte Gattung von Software (dazu gleich mehr), andererseits im Sprachgebrauch der Oberbegriff von Software dreier beziehungsweise vierer Gattungen.

Näher betrachtet gibt es also nicht nur PD-Software, sondern auch Freeware und Shareware. Jedes Programm

Weg auf die PD-Reihen gebahnt. Tryware sind prinzipiell kommerzielle Programme, die jedoch einiger wichtiger Funktionen oder Fähigkeiten beraubt wurden. So kann dies eine Textverarbeitung sein, bei der die Druckfunktion fehlt, oder ein Raytracing-Programm, das auf die Bearbeitung von Szenen mit maximal 100 Punkten begrenzt ist. Mit solchen Programmen läßt sich also genauso arbeiten wie mit dem Original-Programm, das Sie bei fast jedem Software-Laden kaufen können.

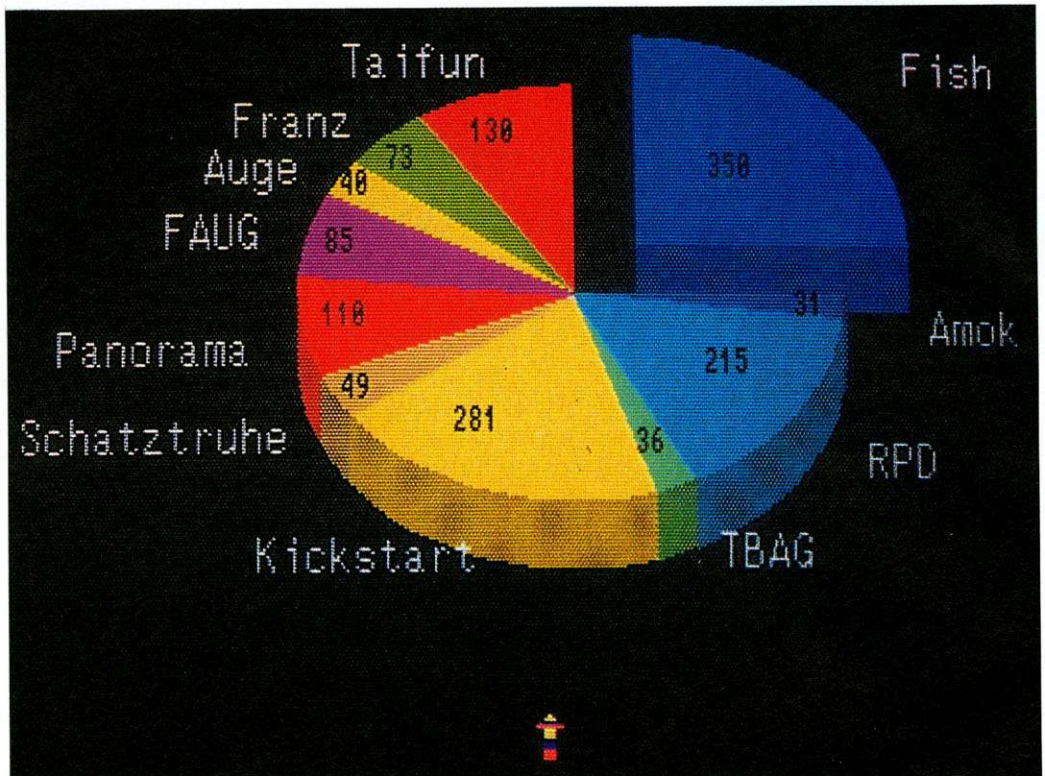
Der Vorteil dieser Idee liegt darin, daß Sie ein solches Programm zu Hause in Ruhe eine beliebige Zeit lang Ihrem ganz persönlichem Test unterziehen können. Sie können so sehr schnell erkennen, ob Ihnen das Programm zusagt oder nicht. Dann gehen Sie in ein Geschäft und kaufen es mit der Gewißheit, daß nicht genau die Funktion fehlt, die Sie für am wichtigsten erachten.

Mit dieser zusätzlichen Vertriebsart von Programmen können die Software-Häuser einem der Hauptargumente der Kritiker kommerzieller Programme entgegenreten. Diese behaupten nämlich, ein Grund für das Unwesen der Raubkopiererei sei neben den immer wieder zu hohen Preisen auch die Tatsache, daß man Software praktisch als "Katze im Sack" kaufen müsse und auf die Testberichte in Fachzeitschriften angewiesen sei.

Wenn wir also in Zukunft von PD sprechen, so meinen wir paradoxerweise in den meisten Fällen gar nicht Programme, die Public Domain sind, sondern Programme, die sich einer der vier angeführten Definitionen "unterwerfen" lassen.

Disclaimer, was ist das eigentlich?

Sollten Sie einmal im Zweifel über den genehmigten Vertriebsweg oder über die Art der genehmigten Nutzung eines Programms sein, so ist jedoch – Begriffdefinition hin und her – der sicherste Ort für definitive Informationen immer noch der sogenannte "Disclaimer" eines Programms. Meist besteht das Vertriebspaket (die Anzahl und Länge der Dateien, die unbedingt zusammen mitein-



Die bekanntesten PD-Serien nach Marktanteilen grafisch aufbereitet

ander weiterkopiert werden müssen, damit die Verbreitung legal bleibt) eines PD-Programmes aus mindestens zwei oder drei Dateien.

Da hätten wir einmal die eigentliche Programmdatei und zum anderen entweder eine Anleitungdatei, in der der Disclaimer bereits untergebracht ist, oder zusätzlich eine dritte Datei, die lediglich den Disclaimer enthält. Der Disclaimer ist meist vom Autor selbst geschrieben (obwohl es in der Zwischenzeit bereits vorformulierte Texte gibt) und gibt an, unter welchen Umständen welche Teile seines Programms weiterverbreitet werden dürfen. Allein dies ist die einzig moralisch haltbare Quelle über zuverlässige Informationen zur Weitergabe eines Programms. Fehlt sie, so können Sie fast davon ausgehen, daß dieses PD-Programm bereits mindestens einmal auf unkorrekte Art und Weise kopiert wurde – nämlich nicht mit allen dazugehörigen Dateien auf einmal. Auch dieser Umstand verbietet jedem Anwender eigentlich die Nutzung des Programms, ob nun PD oder nicht!

Meilensteine

Warum hat nun aber die PD-Software gerade auf dem Amiga einen solchen Erfolg? Dies

hat mit Sicherheit mehrere Gründe, die hier nach und nach erörtert werden sollen. Dabei sind die Ursachen sicherlich vornehmlich in den Anfängen der Amiga-Familie zu suchen. Zum ersten gab es die wohl bekannteste PD-Reihe für den Amiga – die offiziell "Freely Redistributable Software Library For The Amiga" genannte, den meisten jedoch lediglich als Fish-Disk bekannte praktisch seit Gründerzeit der Amiga-Familie in den Jahren 1985/86. Fred Fish, auch liebevoll "Mr Public Domain" genannt, ist der Sammler und Herausgeber dieser Reihe. Er hat den immensen Aufwand nicht gescheut, sich mit der Katalogisierung und Zusammenstellung jeder einzelnen PD-Diskette aus seiner Reihe selbst zu befassen. War anfangs das Hauptproblem noch darin zu sehen, überhaupt genügend Programme zusammenzubekommen, liegt jetzt das Problem eher in einem Überangebot, das bisweilen sogar schon zu einer Auswahl der Programme zwingt, die in diese Reihe mit aufgenommen werden.

In den Gründerzeiten stammten die meisten Programme übrigens von den Entwicklern des Amiga selbst, die auf diese Art und Weise ihr Wissen einer breiten Öffentlichkeit von Amiga-Benutzern zu-

gänglich gemacht haben. Sicherlich haben viele Programmierer von heute ihr erstes Wissen über die Software-Erstellung auf dem Amiga auch aus diesen Quellen bezogen. Andererseits sind viele Autoren von PD-Programmen entweder auf diesem Wege von "großen" Software-Häusern entdeckt worden, oder aber sie haben sich nach einiger Zeit selbst dazu entschlossen, ihr geistiges Können kommerziell zu vertreiben.

Neues System – fehlende Info

Der zweite Punkt liegt wahrscheinlich in der Tatsache, daß der Amiga bei seiner Einführung in den besagten Jahren ein vollkommen neuartiges System darstellte. Verständlicherweise gab es zu Beginn kaum Software und, was noch viel schlimmer war, auch kaum Dokumentation, sieht man einmal von der vor Fehlern nur so strotzenden ersten Ausgabe der offiziellen Dokumentation (ROM-Kernel-Manuals und ähnlichem) ab. Dieses Problem erkannten viele Programmierer der ersten Stunde. Und da sie wohl meist als "echte Freaks" von Maschinen wie dem C64 oder dem Atari XL her kamen, gaben sie ihre Kenntnisse über die Programmierung des Ami-

ga auf diese Art und Weise preis. Damit konnten Sie sich einerseits einen Namen machen und andererseits Leuten in ähnlichen Situationen hilfe reich unter die Arme greifen. Nachdem die PD-Reihen auf dem Amiga dann einen durchschlagenden Erfolg hatten, kamen auch die kommerziellen Anbieter von Programmen auf die Idee, daß sich damit doch eigentlich ein ideales Medium für die Verbreitung, sagen wir mal etwas suggestiver Werbung bot. Damit ist noch lange nicht die schon erwähnte Tryware gemeint, sondern vielmehr die "Produkte" kommerzieller Programme. Wer kennt sie nicht, die sagenumwobene Demonstration des "Jugglers" (Fish-Disk 47, 57 und 97, jeweils in neuen Versionen). Und wie sehr waren alle diejenigen, die sich schon damals mit dem Amiga beschäftigten, davon überrascht, daß mit den kommerziellen Programmen »Sculpt 3D« und »Sculpt/Animate 3D« ein jeder solch fantastische Grafiken zu Hause erstellen können soll. Zugegeben, heute reißt diese Grafik-Demo kaum noch jemanden vom Hocker. Doch damals stand man einfach sprachlos und mit offenem Mund vor dem Computer und wollte kaum glauben, daß der Amiga tatsächlich zu solchen Leistungen fähig ist.

In diesem Zusammenhang sollen auch die zahlreichen anderen, vornehmlich mit den Programmen »Sculpt/Animate 3D«, »Videoscape 1.0 bis 2.1« und »Director« erstellten Demos bleiben. Sie alle verhalfen den Autoren und Erstellern nicht nur zu einem recht großen Bekanntheitsgrad. Namen wie Allan Hastings, Leo Schwab und Joel Hagen sind für jeden Amiga-Insider ein Begriff. Sie förderten sicherlich auch den Kauf der verwendeten Software.

PD auf Erfolgskurs

Der jedoch meines Erachtens wichtigste Punkt für den Erfolg der PD-Software auf dem Amiga ist die Tatsache, daß es sich bei diesem Computer um ein – für diesen Preisbereich – ungewöhnlich komplexes System handelt. Und komplexe Systeme bringen es nun mal mit sich, daß sie auch viel Raum für Verbesserungen beinhalten, will nicht sagen konzeptionelle Fehler aufweisen. Um es mit den Worten

des erwähnten Leo Schwab zu sagen "There are NO bugs in MY code. However, there may be any number of undesirable features".

Und solche Probleme lassen sich meist am besten durch kleine Hilfsprogramme, sogenannte Utilities lösen. Ob es sich nun um einen Patch einer Betriebssystemfunktion (beispielsweise »Overscan« von Fish-Disk 133) oder um komplexere Hilfsprogramme wie den Dateiretter »DiskSalv« (in der neuesten Version 1.42 auf Fish-Disk 251) handelt, immer wieder können auch kleine Programme große Probleme lösen. Dabei ist die Software von ihren "Ausmaßen" her jedoch oft so gering und hat ein solch begrenztes Anwendungsgebiet, daß sich eine kommerzielle Vermarktung nicht lohnen würde.

Genug damit, warum PD-Software auf dem Amiga so erfolgreich ist. Im Endeffekt wird Ihnen jeder User andere Gründe dafür nennen, doch dürfen die zuvor aufgeführten wohl die wichtigsten für den Erfolg sein. Jetzt soll uns vielmehr die Verbreitung und der Vertrieb von PD-Software interessieren.

Viele Jäger sind des Hasen Tod – oder?

Die Zahl der PD-Reihen ist auf dem Amiga schon einige Zeit lang nicht mehr überschaubar. Eine Durchforstung zahlreicher Zeitschriften, Mailboxen und anderer Medien ergab eine Liste von fast 30 Reihen, die jeweils mehr oder minder groß sind. Die wichtigsten und in Deutschland am weitesten verbreiteten sind dabei in einer Grafik (siehe Bild) aufgeführt – jeweils mit der Anzahl der Disketten, wie Sie bei der Erstellung dieser Zeilen aktuell waren. Andere, in der Grafik nicht aufgeführte Reihen sind beispielsweise ACS, Amicus, Amuse, Austria, Cactus, Chiron Conception, ES, Kiss, Poseidon, RHS, RMS, RW, Safe, Tornados, PP und UKAug. Es soll jedoch **ausdrücklich** erwähnt werden, daß diese Liste keinerlei Anspruch auf Vollständigkeit erhebt. Fast täglich tauchen neue PD-Reihen auf, andere werden eingestellt, weil der Verteiler nach einiger Zeit merkt, daß eine solche Aufgabe doch viel mehr Arbeit aufwirft, als anfänglich angenommen wurde.

Wie kommen nun aber die einzelnen Disketten der PD-Reihen-Hersteller schließlich bis zu Ihnen, dem Endverbraucher? Hier stellt sich uns das derzeit wohl meistdiskutierte Thema im Bereich der PD-Software. Vom Grundgedanken her sollten solche Programme ja kostenlos sein, was für eine vornehmliche Verbreitung der Disketten per "Mund-Propaganda" sprechen würde. Also, eventuell interessante Disketten vielleicht bei einem Freund sehen und diese dort auf mitgebrachten, eigenen Disketten vervielfältigen?

PD-Preiskarussell

Leider funktioniert dieses Vorgehen in Deutschland bei weitem nicht so gut, wie beispielsweise in den USA, wo es kaum kommerzielle PD-Vertreiber wie in Deutschland gibt und die Verbreitung der PD-Software trotzdem einwandfrei klappt. Aufgrund dieses Nicht-Funktionierens haben sich einige findige Geschäftsleute auf diese Marktlücke gestürzt und vertreiben PD-Disketten zum Stückpreis zwischen 2,50 DM und 10,- DM – je nach Anbieter mehr oder weniger – professionell.

Betrachtet man die reinen Herstellungskosten (Leerdiskette, Geräteabnutzung, Steuern, Miete, Gehalt und so weiter), so kann man sich prinzipiell schon vorstellen, daß ein Vertrieb von PD-Disketten zum Stückpreis von 2,50 DM eigentlich – vor allem steuerlich – wohl kaum korrekt funktionieren kann. Jedoch, ob der Preis deshalb gleich bei 10,- DM pro Diskette angesetzt werden muß, ist äußerst fraglich.

Dennoch halte ich dieses Problem für weit weniger umfangreich, als dies anscheinend eine Vielzahl deutscher PD-Autoren tut. So kann man die meisten Argumente damit abtun, daß wir nun mal im Kapitalismus leben, in dem – übertrieben ausgedrückt – ein jeder versuchen sollte, mit möglichst wenig Aufwand einen möglichst großen Gewinn zu erzielen und niemand verpflichtet ist, PD-Disketten zu solch "immensen" Preisen zu kaufen. Ein anderes

Problem ist jedoch die Tatsache, daß hier Menschen mit dem geistigen Eigentum anderer Gewinn machen, was bei den angesprochenen hohen Diskettenpreisen nicht ganz ausgeschlossen ist.

Schwarze Schafe – weiße Schafe

Zur Klärung dieses Themas wäre es doch einmal – zumindest hypothetisch – interessant, wenn ein PD-Vertreiber Einblick in seine Bücher bieten oder gar einem entsprechend geschulten PD-Autor eine Buchprüfung genehmigen würde. Da dies jedoch wahrscheinlich nicht geschehen wird, kann man wohl davon ausgehen, daß sich der Streit in Zukunft weder beilegen noch verschärfen wird. Beide Seiten (kommerzielle Verteiler und PD-Autoren) haben gute Argumente, die jedoch so weit voneinander entfernt sind, daß eine gütliche Einigung außer Sichtweite liegt. Bei all diesen Vorgängen kann man lediglich hoffen, daß damit die PD-Szene auf die Dauer gesehen keinen "Knacks" bekommt, worunter letztendlich Sie als Endanwender leiden müßten. Damit wollen wir den Grundlagenteil über PD-Software beenden. Auf den folgenden Seiten wird es konkreter. Wir stellen Ihnen die sogenannten **Highlights** der PD-Szene sowie weitere interessante Fakten vor.

(vb)



Ottmar Röhrig

PD-Tophits

The best of...

So beginnen üblicherweise die Plattentitel bekannter Interpreten, deren Label sich nun endlich dazu entscheiden konnte, die bekanntesten Stücke auf einem einzigen Tonträger zu veröffentlichen. Hier geht es jedoch nicht um Musik, sondern um Public-Domain-Software, von der wir Ihnen in diesem Rahmen die bekanntesten "Meisterstücke" vorstellen möchten.

Die Zahl der PD-Disketten mit Programmen für den Amiga nimmt langsam unüberschaubare Größen an. Genaue Zahlen kann niemand nennen, doch kann davon ausgegangen werden, daß mindestens 1500 bis 2000 PD-Disketten für den Amiga vorhanden sind. Und alle sind meist randvoll mit mehreren Programmen bespielt, so daß die Anzahl der PD-Programme sicherlich in die Zehntausende geht.

Trotzdem sollte man bei diesen Zahlen nicht zu euphorisch werden. Es stellt sich nämlich immer wieder heraus, daß gerade kleinere PD-Reihen meist nur Kompilationen der größeren sind. Das heißt, der Ersteller dieser neuen Reihe hat einfach die seiner Meinung nach besten Programme von anderen Disketten genommen und auf einer neuen Diskette zusammengestellt. Dies ist nicht verboten und unter Umständen sogar durchaus sinnvoll, doch kann so schnell Verwirrung entstehen. Diese endet schlimmstenfalls in dem Ärger über den Lieferanten der PD-Software, wenn Sie sich mehrere Disketten bestellt haben und dann feststellen müssen, daß ein Großteil der gelieferten Programme identisch ist.

Dies ist der Grund, weshalb Sie sich genauestens informieren sollten, was auf jeder einzelnen Diskette einer PD-Reihe abgelegt ist. Sinnvollerweise verwendet man dafür die sogenannten Katalog-Disketten, die fast jeder PD-Anbieter kostengünstig verschickt. Diese enthalten meist die kompletten Inhaltsverzeichnisse aller Disketten der von diesem Vertreiber angebotenen PD-Reihen. Doch ist meist auch das noch nicht genug, um eine Entscheidung über die Nützlichkeit eines

Programms zu treffen. Deshalb sollen hier die bekanntesten und nützlichsten Programme aus der PD-Szene in einer fundierten Übersicht vorgestellt werden. Einige haben längere Beschreibungen, andere kürzere, doch allen ist gemeinsam, daß sie die von ihnen erledigte Aufgabe besser machen als andere PD-Programme aus derselben Sparte – ja, oftmals sogar besser als vergleichbare kommerzielle Programme. Die Auswahl der Programme beruht dabei übrigens auf Reaktionen von Benutzern und dem weitverbreiteten "Ruf" eines Programms, den wir natürlich zuvor überprüft haben. In der Übersicht wurden alle Programme der Fish-Disks bis zur Diskette 350 berücksichtigt.

Es sollte Sie übrigens nicht wundern, daß die Anzahl der vorgestellten Programme gegen Nummer 350 stetig zunimmt. Das liegt nicht daran, daß früher erschienene Programme schlechter waren, doch benötigt die Weiterentwicklung eines guten Pro-

gramms oft sehr viel Zeit und auf den Fish-Disks finden sich zahlreiche Versionen ein und derselben Software, jeweils mit entsprechenden Verbesserungen. Natürlich wurde immer nur die aktuellste Version mit ihrer Disketten-Nummer vorgestellt, was die erwähnte Tatsache erklären dürfte. Ich möchte hier nicht weiter darauf eingehen, ob ein Programm Public Domain, Freeware oder Shareware ist; das entnehmen Sie bitte den jeweils beiliegenden Dokumentationen beziehungsweise auch der in den letzten Heften bereits gestarteten Reihe unserer PD-Beschreibungen.

Lediglich die Tryware ist hier als ein separater Teil aufgeführt worden; die Übersicht in Tabelle 1 ermöglicht Ihnen eine schnelle Lokalisierung aller auf den Fish-Disks als Demoverionen veröffentlichten kommerziellen Programme. Die Tabellen 2 bis 4 geben Ihnen Auskunft darüber, welche Programme im Laufe dieser Seiten vorgestellt werden und in welchem Bereich sie

zu finden sind. Wir haben bei dieser Übersicht sämtliche Grafik-Demos, sowie die zahlreichen Display-Hacks (kleine Programme, die meist recht lustigen Unfug im System treiben, indem sie Mauszeiger rotieren, Bildschirme zuschneiden, Fenster zerschmelzen oder beim Diskettenwechsel Töne ausgeben) nicht berücksichtigt. Letztere vor allem, weil es viel interessanter ist, per Zufall auf ein solches Gag-Programm zu stoßen und es dann zu starten, ohne zu wissen, was geschehen wird. Des weiteren wurden all die Programme ausgelassen, die zwar auf PD-Disketten das erste Mal veröffentlicht wurden, dann jedoch mit der Betriebssystemversion 1.3 offiziell den Weg auf die Workbench gefunden haben. Dazu gehören etwa solche Programme wie »Cmd«, »FastFonts« oder »Xicon«.

Alle hier vorgestellten Programme lassen sich in die Kategorien Utilities, Applikationen und Spiele unterteilen. Zwischenüberschriften informieren Sie jeweils über den gerade besprochenen Bereich. Der Großteil der Beschreibungen bezieht sich übrigens auf Programme der Fish-Disks, da diese Reihe die wohl am meisten verbreitete ist. Lediglich einige besondere Programme, die ausschließlich auf anderen Disketten zu finden sind, wurden am Ende des jeweiligen Bereiches noch mit aufgenommen.

PD-Utilities

Ist Ihnen der Workbench-Bildschirm auch manchmal zu klein? Ist anscheinend nicht genügend Platz für alle Fenster inklusive der zahlreichen Inhaltsverzeichnisse? Wenn Sie nicht in der glücklichen Lage sind, in diesem Fall auf

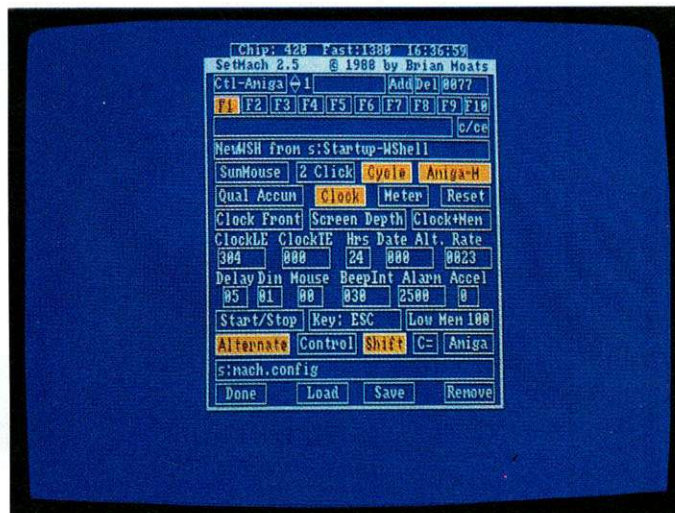


Bild 1. »Mach II« vereint zahlreiche Hilfsprogramme (Fish 254)

den Interlace-Modus umschalten zu können (was nur Besitzern einer Flickerfixer-Karte angeraten sei), dann kann Ihnen »MoreRows« von **Fish 54** sicherlich helfen. Es erhöht zwar nicht direkt die Auflösung einer Workbench, doch kann es die Anzahl der Pixel in der Vertikalen und Horizontalen doch erheblich vergrößern. Bildschirmgrößen von etwa 670 x 270 ohne Interlace-Modus sind problemlos möglich. Das einzige, was Sie tun müssen, ist, den Monitor nach einer solchen Veränderung wieder richtig zentriert zu justieren.

Einem anderen Problem nimmt sich »FillDisk« von **Fish 61** an. Es überschreibt alle nicht belegten Blöcke einer Diskette explizit mit Dummy-Bytes, so daß Ihnen beim Weitergeben von Disketten garantiert ist, daß der Empfänger nicht mit einem Dateirettungsprogramm über die Diskette geht und so vielleicht noch die vor Monaten zwar gelöscht, aber vom AmigaDOS nicht überschriebenen Passwortdateien Ihrer Mailboxen findet.

»DosKwik« von **Fish 129** besteht aus zwei Teilprogrammen »ReadKwik« und »WriteKwik«. Es versetzt Sie in die Lage, den Inhalt eines Verzeichnisses in einem DosKwik-eigenen Diskformat auf einer oder mehreren Disketten abzulegen (je nachdem, wie viele Daten geschrieben werden, fordert das Programm automatisch neue Disketten) und zu archivieren. Der Vorteil dieser Methode besteht darin, daß Sie beispielsweise den Inhalt der RAM-Disk genauso abspeichern und am nächsten Tag wiederherstellen können wie das mit einem »copy ram: df0: all clone« möglich wäre, doch dauert das erstens 4 bis 5 mal länger als mit »DosKwik«, und zweitens ist es so nicht möglich, Dateien mit einer Länge von mehr als 880 kByte zu sichern, was für »DosKwik« kein Problem ist. Wer regelmäßig Termine vergißt oder bestimmte Aufgaben vom System zu definierten Zeiten oder Zeitabständen automatisch erledigen lassen möchte, der sollte sich mit »AmiCron« von **Fish 139** befassen. Es ermöglicht das Anlegen einer Ereignistabelle, in der verschiedene Befehle zu beliebigen Zeiten oder in fest definierten Zeitabständen (jeden zweiten Montag um 15.30 oder ähnliches) eingetragen und

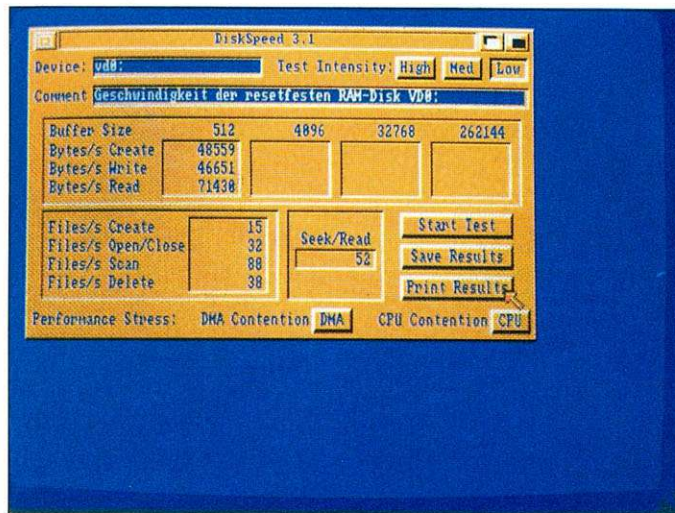


Bild 2. »DiskSpeed« (Fish 329) testet nicht nur reine Schreib-/Lese-Geschwindigkeiten

beim Hochfahren des Rechners automatisch ausgeführt werden, ohne daß Sie sich darum kümmern müssen. »TurboBackup« ist ebenfalls auf **Fish 139** zu finden und ist das schnellste und wohl sicherste Kopierprogramm für ganze Disketten, das in der PD verfügbar ist. Es kopiert jedoch ausschließlich DOS-Disketten ohne irgendwelche "Besonderheiten", wie Lese-Fehler oder Kopierschutztechniken verschiedenster Art. Neben einer Kopie inklusive Verify-Durchgang in gut 100 Sekunden kann es auch mit vier Laufwerken drei Kopien einer Diskette gleichzeitig erzeugen, was zusätzliche Zeitgewinne bringt.

Für Programmierer interessant ist das kleine Hilfsmittel »Surveyor« auf **Fish 151**. Es öffnet ein kleines Fenster auf dem jeweils vorliegenden Intuition-Screen und zeigt

fortlaufend die Koordinaten des Mauszeigers an. So können Sie schnell und einfach die Koordinaten für die Position eines eigenen Gadgets oder einer Grafik feststellen, was ansonsten meist mehrere Compiler-Durchgänge erfordern würde. »ScreenX« von **Fish 158** sorgt dann dafür, daß Sie mausgesteuert zwischen den im System vorhandenen Bildschirmen hin- und herspringen können, oder es versetzt Sie in die Lage, Intuition-Bildschirme als IFF-Grafiken auf Knopfdruck abzuspeichern. Auch das – jedoch nicht systemkonforme – Schließen von Fenstern und Screens ist mausgesteuert möglich, sollte aber nur von Programmierern angewendet werden, die sich der möglichen Folgen solcher Aktionen bewußt sind.

Fish 164 enthält wiederum zwei interessante Programme.

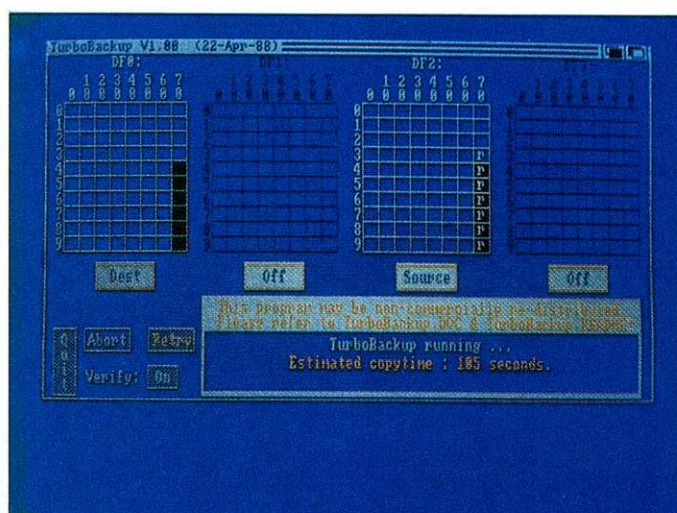


Bild 3. Das schnellste Kopierprogramm der PD hier im Einsatz (»TurboBackup«, Fish 139)

»NewZap« lädt Dateien ein, zeigt sie als hexadezimale sowie als ASCII-Notierungen an und ermöglicht es dem Benutzer, einzelne Bytes komfortabel zu ändern; ein ideales Programm für das Patchen anderer Software. »Zoo« hingegen ist auf zahlreichen Rechnern in immer wieder angepaßten Versionen zu finden. Ähnlich wie das Programm »LHarc« (**Fish 312**) ist »Zoo« ein Dateiarchivierer (nicht zu verwechseln mit einem Backup-Programm), der beliebige Dateien komprimiert und eine frei wählbare Anzahl solcher Dateien in einer Archivdatei verbindet. So lassen sich eine Vielzahl von Dateien beispielsweise für die Übertragung an eine Mailbox zusammenbündeln, komprimiert, das heißt mit verkürzter Übertragungszeit, uploaden und vom Empfänger wieder komfortabel entpacken.

Den gleichen Funktionsbereich deckt auch »PKAZip« (**Fish 318**) ab, eine Adaption des von MS-DOS her bekannten »PKZip«. Im Gegensatz zu »LHarc« und »Zoo« arbeitet es jedoch vollkommen mausgesteuert und bietet eine Intuition-Oberfläche, wo die anderen Programme lediglich Kommandozeilenparameter bieten. Für die Kompression ganzer Disketten sind diese Programme jedoch meist nicht zu verwenden, da sie Bootblöcke und ähnliches nicht mitkomprimieren können. Doch bietet sich für diesen Zweck »LHWarp« an, das der geneigte Benutzer ebenfalls auf **Fish 318** findet. Es komprimiert Disketten trackweise und erlaubt so die "1-zu-1-Übertragung" ganzer Disketten per DFÜ bei vertretbarem Aufwand, da alle Daten komprimiert sind.

Einen der Diamanten der PD enthält **Fish 165**. Hier findet sich das Programm »Conman«, das nicht nur ein vollwertiger Ersatz für die neuen Funktionen der Shell gegenüber denen des CLI ist, sondern zudem noch eine Vielzahl anderer Verbesserungen beinhaltet. Auf jeder Eingabezeile können zuvor eingetippte Zeilen dank der History-Funktion wieder aufgerufen, Buchstaben wortweise gelöscht und zahlreiche andere Editierfunktionen vorgenommen werden. Es ist beispielsweise sogar möglich, den Einfügemodus zeitweise oder dauernd durch den Überschreibmodus zu ersetzen

und so weiter. **Fish 180** bietet Ihnen mit dem Programm »**Browser**« die Möglichkeit, sich auf der Workbench Dateien nicht nur als Icons, sondern auch als Texte im Fenster anzeigen zu lassen. Somit werden nun von der Workbench aus endlich auch Dateien ansprechbar, die keinen Komplementäreintrag mit der Erweiterung ».info« besitzen, ähnlich, wie das auch bei Workbench 2.0 der Fall ist. Kombiniert man dieses Tool noch mit dem Programm »**Set-Font**« von **Fish 182**, so kann das Aussehen der Workbench im groben Stil fast dem der neuen Version 2.0 angepaßt werden, ermöglicht »SetFont« doch die Veränderung eines jeden Zeichensatzes. So können Sie für Fenstertitel beispielsweise einen größeren, für die Iconbeschriftungen jedoch einen kleineren Zeichensatz – auch mit einem anderen Schrifttyp als »Topaz« – einsetzen.

Wer sich als Programmierer oder interessierter Anwender für den IFF-Standard begeistert, kann auf der gesamten **Fish 185** so ziemlich alle Informationen beziehen, die bis heute von Commodore und Electronic Arts (den Initiatoren dieses Standards) veröffentlicht wurden. Die »IFF-Disk« enthält dabei nicht nur Beschreibungen der wichtigsten Formate, sondern auch zahlreiche Beispielprogramme und Unterroutinen, die die Implementation in eigene Programme erleichtern. Besonders interessant, gerade für Programmierer, ist dann auch **Fish 348**, auf der sich die sogenannte »**ILBMLib**« befindet. Dies ist eine diskettenge-



Bild 4. Wie schnell ist das System? »Performance« von Kickstart 250 zeigt differenzierte Werte grafisch an

stützte Funktionsbibliothek, die zahlreiche Funktionen zur Be- und Verarbeitung von IFF-Dateien beliebigen Typs enthält und von jeder Programmiersprache aus ansprechbar ist. Baut man dann in sein eigenes Programm noch einen fertigen und äußerst komfortablen Datei-Requester ein, steht professioneller Software nichts mehr im Wege. Ein guter Weg, dies zu bewerkstelligen, ist »**FileWindow**« von **Fish 336**, das einen schönen Datei-Requester zur Verfügung stellt – und zwar mit Source und der Erlaubnis, ihn auch in kommerziellen Programmen verwenden zu dürfen, ohne daß Sie dazu die Erlaubnis des Autors einholen müssen.

Weiter geht's mit dem »**Key-MapEd**« von **Fish 193**. Er ermöglicht es Ihnen, die Tastaturbelegung komfortabel und ohne Kenntnisse über den

Aufbau der Tastaturreiber vollkommen mausgesteuert zu verändern und den ganz persönlichen Bedürfnissen anzupassen. Schreiben Sie also beispielsweise oft französische Texte, so können die französischen Sonderzeichen auf den numerischen Ziffernblock oder andere Tasten gelegt werden, was komplizierte Fingerakrobatik zur Erreichung der gleichen Zeichen mittels der Alt- und Shift-Tasten erspart.

Den Viren Herr zu werden, ist Aufgabe des Programms »**VirusX**«, in der Version 4.0 auf **Fish 287** zu finden. Es erkennt alle bis dato bekannten Bootblock- und – mittels eines anderen Hilfsprogramms – Link-Viren. Doch kann die Version 4.0 auf einigen Rechnern zu Problemen führen. Sollte das bei Ihnen der Fall sein, so verwenden Sie am besten die Version 3.2 von **Fish**

216. Ein anderes Link-Viren-Erkennungs-Tool ist »**LVR**« von **Fish 331**, das sogar darauf ausgelegt ist, infizierte Dateien automatisch zu löschen oder mit einem Dateikommentar zu versehen.

»**SetCPU**« von **Fish 223** ist nur hilfreich, wenn Sie einen 68020/68030-Prozessor besitzen. Es ermöglicht das Abschalten der einzelnen Caches, das Kopieren der Kickstart in das 32-Bit-RAM und andere systemspezifische Sachen. Besitzen Sie auch noch einen passenden mathematischen Coprozessor des Typs 68881/68882, so können Sie mit »**Mathtrans**« von **Fish 319** die normale Mathematik-Funktionsbibliothek »mathtrans.library« gegen eine spezielle für mathematische Coprozessoren austauschen. Damit steigert sich die Rechengeschwindigkeit, auch von nicht speziell für Coprozessoren zugeschnittenen Programmen, um mindestens das Vierfache.

»**MyMenu**« (**Fish 225**) hängt an die Titelleiste der Workbench auf Wunsch selbst definierte Menüpunkte an. So können Sie – nach korrekter Installation – eigene Programme oder Applikationen per Menü von der Workbench aus aufrufen.

Für Festplattenbesitzer ist mit »**KwikBackup**« (**Fish 234**) ein Backup-Programm auf dem PD-Markt, das zwar nicht so viele Funktionen besitzt wie beispielsweise »**MRBackup**« von **Fish 327**, jedoch im Gegensatz zu diesem ein eigenes Diskettenformat benutzt und so sehr schnelle Backup-Zeiten erzielt, die sich durchaus mit denen kommerzieller Pro-

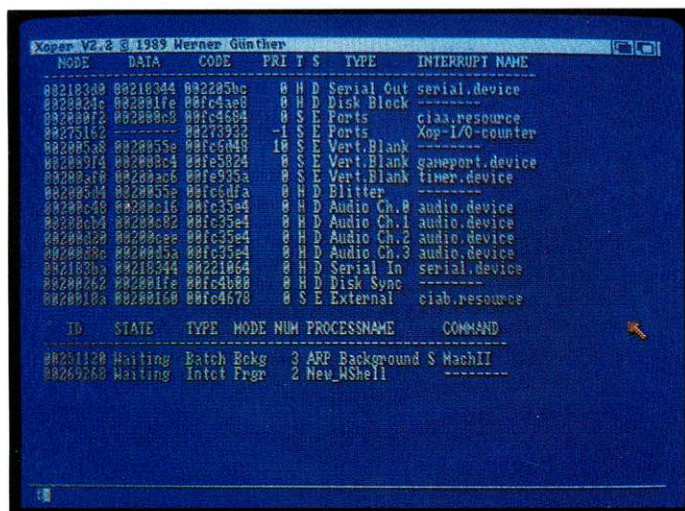


Bild 5. Was im System los ist, kann man mit »Xoper« am besten beobachten (**Fish 318**)

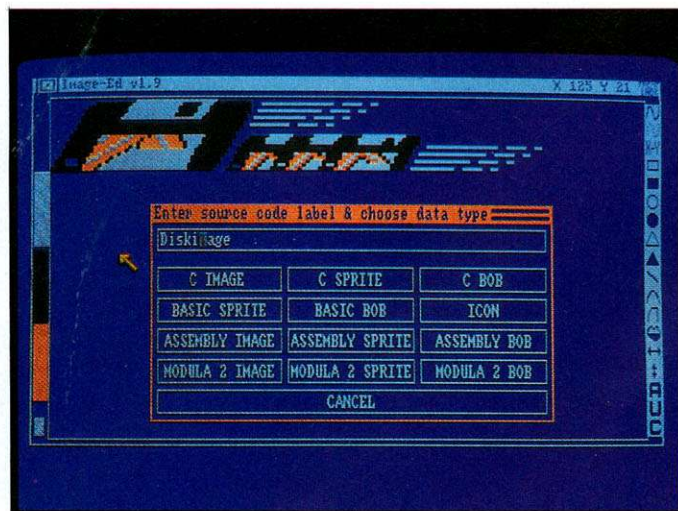


Bild 6. »Image-Ed« (**Fish 242**) kann Grafikdaten auch als Source-Code für verschiedenste Programmiersprachen ausgeben

Name des kommerziellen Programms	Fish-Disk Nr.	Unterschied zur kommerziellen Version
AegisDraw	29	Save-Funktion stillgelegt
TxE	31	Search/Replace stillgelegt, maximale Dateilänge 10 kByte
Business Management	43	beschränkte Dateigröße
PerfectSound	50	Es fehlt der Digitizer, und die Programmversion ist nicht die neueste
Shanghai	55	Nur Solitaire-Version, nur ein Spielaufbau
SetKey	70	Save-Funktion stillgelegt
PowerWindows	71	Erzeugung von Source-Code stillgelegt
Disk-2-Disk	82	Copy-Funktion stillgelegt
TeX	83	Maximal 10 Seiten-Dokument, nur kleine Dateien, wenig verschiedene Zeichensätze
FileIISG	86	Save-Funktion stillgelegt
MicroFicheFiler	89	Save-Funktion stillgelegt
CygnusEd	95	Maximale Dateilänge 5 kByte
Disk Cataloguer	98	Maximal 100 Einträge
M2Amiga	113	Limitierte Anzahl von IMPORTs, limitierte Code-Länge
ExpressPaint 1.1	117	Save-Funktion stillgelegt, begrenzte Anzahl von ausführbaren Funktionen, Ausdruck gestört
C-Light	146	Maximal 10 Objekte pro Szene
Dynamic Sound Machine	168	Save-Funktion stillgelegt
ReSource	232	Save-Funktion stillgelegt
Comal	296	Save-Funktion stillgelegt
SceneGen	299	Lediglich Vorführung, keine Mausaktivitäten möglich
Midi Sample Wrench	307	Keine Sampler-Treiber-Software vorhanden, es kann also nur mit der mitgelieferten Welle gearbeitet werden
ANSIED	324	Save-Funktion stillgelegt
Boing	335	Maximale Spieldauer 5 Minuten
SpeakerSim	340	Nur ein Lautsprecher, kein Interlace-Modus, Plot-Funktion stillgelegt

Tabelle 1. Diese Liste gibt Ihnen Auskunft darüber, welche kommerziellen Programme bereits als Tryware-Versionen auf den Fish-Disks veröffentlicht wurden

gramme vergleichen lassen. Das erwähnte »MRBackup« hat jedoch den Vorteil, daß es bei der Archivierung die anfallenden Daten automatisch komprimiert, also mit weit weniger Disketten auskommt als andere Backup-Programme. Auch »FmsDisk« von **Fish 294** ist ausschließlich für Festplattenbesitzer von Interesse. Durch einen speziellen Handler ist dieses Programm in der Lage, auf einer Festplatte einzelne Disketten so zu emulieren, daß Sie sogar einen Diskcopy-Befehl auf Teile der Platte ausführen können oder Vertriebsdisketten auf der Festplatte (mit entsprechend höherer Geschwindigkeit) zusammenstellen und diese danach mittels »DiskCopy« vom CLI aus auf eine echte Diskette übertragen können.

Auf **Fish 240** befindet sich ein wohl den meisten bekanntes Programm namens »RunBack«. Es stellt dem Benutzer die Möglichkeit zur Verfügung, nun auch aus dem CLI Programme als Background-Tasks starten zu können, ohne

daß dadurch das CLI-Fenster bis auf alle Ewigkeit hinaus (das heißt bis zum nächsten Reset oder der Beendigung des Background-Programms) nicht geschlossen werden kann, da die Textausgabe des Programms normalerweise immer noch in dieses Fenster geleitet wird. »RunBack« umgeht dieses Problem, indem es die Ausgabe – einfach gesagt – ins Nichts umleitet. Der Vorteil dieser Version gegenüber anderen, weitverbreiteten ist die Tatsache, daß hier der Suchpfad des CLI mit beachtet wird.

Ebenso bekannt dürfte die »ASDG-rrd« genannte VD0: RAM-Disk von **Fish 241** sein. Dies ist eine resetfeste RAM-Disk, die – im Gegensatz zur RAD: der Workbench – nur jeweils soviel Platz im Speicher belegt, wie auch gerade vom Inhalt der RAM-Disk benötigt wird. Ein äußerst hilfreiches Utility, das bei niemandem fehlen sollte.

Ein unbedingtes Muß für jeden Anwender stellt das Programm »DiskSalv« dar, zum

letzten Mal auf **Fish 251** erschienen. Es ist das derzeit wohl beste Programm zur Rettung defekter Dateien oder durch Schreib-Lese-Fehler zerstörter Disketten oder Festplatten. Es können zwar nicht alle erdenklichen Fehler repariert werden, doch bietet das Programm in den allermeisten Fällen eine große Chance, versehentlich oder durch einen Unfall gelöscht Daten wiederzubekommen. Ist der Grund für »verlorene« Daten jedoch darin zu suchen, daß auf der Festplatte einfach keine Ordnung herrscht und Sie verzweifelt eine Datei in den unzähligen Unterverzeichnissen suchen, so kann »WhereIs« von **Fish 321** helfen.

Es durchsucht beliebige Bereiche einer Diskette oder vorzugsweise einer Festplatte nach Dateinamen oder -namensmustern und zeigt Verzeichnis und Unterverzeichnis an, in dem eine passende Datei gefunden wurde.

Das Programm »MuchMore« – vom gleichen Autor wie »KwikBackup« – ist ein voll-

wertiger Ersatz für den Textanzeiger »More« der Workbench. Doch hat es den Vorteil, daß zum einen das Scrolling des Textes pixelweise statt zeilenweise erfolgt und zweitens der Text auch durch bestimmte Klicks und Bewegungen mit der Maus gestartet oder angehalten werden kann.

Ein Dreierpaket von Programmen erfreut sich in letzter Zeit immer größer werdender Beliebtheit. Der »PowerPacker« von **Fish 253** komprimiert nicht nur Datendateien, sondern auch Programmdateien so, daß sie trotzdem normal gestartet werden können und sich dann einige Sekunden lang automatisch selber entpacken. Außer einer kleinen Wartezeit merkt man also nichts – neben der Tatsache natürlich, daß wesentlich mehr Programme auf die Diskette oder Festplatte passen. Für Programmierer sind die Kompressions- und Dekompressionsfunktionen als Sourcecode beigelegt, so daß auch eigene Programme von komprimierten Daten Gebrauch machen können. Genau dafür sind nämlich »PPMore« und »PPShow«, beide von **Fish 334**, gute Beispiele. Ersteres zeigt komprimierte Textdateien genauso an, als wären sie nicht komprimiert, letzteres tut dasselbe mit IFF-Grafiken. Dabei ist vor allem »PPMore« nicht nur ein Ersatz für das »More« der Workbench, sondern bietet darüber hinaus noch zahlreiche andere Funktionen, wie Scrollen mit der Maus, schnelle Suchalgorithmen und so weiter.

»MachII« (**Fish 254**) vereint zahlreiche andere Hilfsprogramme in sich und ist die derzeit beste Hilfe, die man sich bei der generellen Arbeit mit dem Amiga vorstellen kann. Nicht nur die Anzeige von Zeit, Datum und freiem Speicher in der Titelzeile, sondern auch Funktionen wie das Aufrufen eines CLI bei Tastendruck, die freie Erstellung von Tastatur- und Mausmakros, der Maus-Beschleuniger, die vereinfachte Wahl des vordersten Bildschirms, der tastengestützten Vor- und Zurückbewegung beliebiger Fenster oder der automatischen Aktivierung des Fensters, über dem sich gerade die Maus befindet – all das und noch mehr ist hier in einem einzigen Programm vereint, das zu Recht als eines der besten Hilfsprogramme in aller Munde ist.

**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.

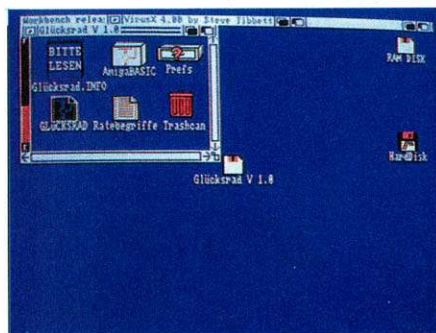
Auch in diesem Fall hilft Ihnen die Databox von Amiga DOS aus der Patsche: Neben den Quelltexten im ASCII-Format finden Sie jeweils auch das fertige, lauffähige Programm. Sie brauchen es nur von der Databox-Diskette aus zu starten.

DATA

**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".



»Glücksrad« – ein Denk- und Ratespiel

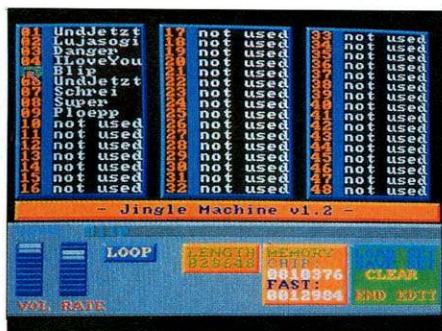


Würfeln und geschicktes Setzen stehen bei »Risiko« im Vordergrund

Alle Listings und Programme auf Diskette –
Computer einschalten – Diskette einlegen –
los geht's

2 Disketten
voller
Programme

BOX



Sound- und Spezial-
effekte, mit der »Jingle-
Machine« kein Problem

24,- DM

Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:

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

Zahlungsweise:

Am einfachsten per Vorkasse (Verrechnungsscheck) oder als Nachnahme zuzüglich der Nachnahmegebühr. (Bei Lieferungen in das Ausland ist Nachnahme nicht möglich.)

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege



Basic

C

Modula-2

Assembler

Lauffähiges Programm

Inhalt der Databox AMIGA DOS 9/90

X	X	»EasyCalc« ist eine Tabellenkalkulation, mit der sich für den eigenen Bedarf alle anfallenden Rechnungen in einem Haushaltsplan eintragen lassen.
X	X	Mit »AUTOFFS« ist ein problemloser Diskettenwechsel möglich.
X	X	Geheime Nachrichten, mit »Disknote« kein Problem
X	X	Programme zu den Grundlagen der Library-Programmierung finden Sie in der Requester-Werkstatt.
X	X	Um Schleifen und Verschachtelungen dreht sich der Modula-Kurs.
X	X	Mit der »Jingle-Machine« lassen sich "irre" Soundeffekte erzielen.
X	X	Und zusätzlich auf der Diskette:
X	X	»Glücksrad« – ein Ratespiel der gleichnamigen Sendung auf SAT 1 nachempfunden
X	X	»Risiko« ist ein ansprechend aufbereitetes Würfelspiel

Auf **Fish 285** findet sich »Dyna-Show«, ein Programm, das auch im Lieferumfang des kommerziellen Produktes »Digi-View Gold 4.0« enthalten ist. »Dyna-Show« ist das derzeit einzige Programm, das Bilder in den neuen Grafik-Modi »Dynamic-HAM« und »Dynamic-HiRes« darstellen kann.

Die allzu langsamen Diskettenoperationen unter AmigaDOS können Sie mittels »FastDisk« von **Fish 286** beschleunigen. Dieses Programm liest den Inhalt einer Diskette ein und beschreibt die Daten- und Verwaltungsblöcke genauso neu, daß das Einladen von Dateien und das Anzeigen von Verzeichnissen schneller vonstatten geht. Das kommt daher, daß das Programm die Blöcke so anordnet, daß sie möglichst nacheinander liegen und der Computer nicht dauernd den Schreib-/Lesekopf des Diskettenlaufwerks neu positionieren muß, was ja auch immer Zeit kostet.

Wie schnell ein Laufwerk, eine Festplatte oder ein beliebiges anderes Speichermedium ist, kann man wohl am genauesten mit »DiskSpeed« feststellen. Dieses Programm von **Fish 329** ist dabei eines der objektivsten, die derzeit auf dem Markt gefunden werden können. Es mißt die Geschwindigkeit von der Datei-erstellung bis hin zur -löschung, dem Auffinden bestimmter Einträge und natürlich der reinen Schreib- oder Lesegeschwindigkeit. Zusätzlich können die Ergebnisse verschiedener Laufwerke archiviert werden.

Die Übersicht vor allem auf Festplatten zu behalten, ist Ziel des Hilfsprogramms »Atree«, das vor allem die Neugier ehemaliger MS-DOS-Benutzer auf sich ziehen dürfte. Das kommt daher, daß es ähnlich aufgebaut ist wie Programme à la »Xtree« oder Teile des bekannten »PCTools«. Es stellt eine Festplatte als Baumstruktur mit Verzeichnissen, Unterverzeichnissen und Dateien dar, die tastatur- und mausgesteuert kopiert, umbenannt, verschoben und gelöscht werden können. Möchten Sie nicht nur das Benutzergefühl von MS-DOS haben, sondern auch Disketten dieses veralteten Betriebssystems lesen und beschreiben können, ist ein fantastisches Programm namens »Msh« das richtige für Sie. Es ermöglicht nämlich genau dies – das Le-

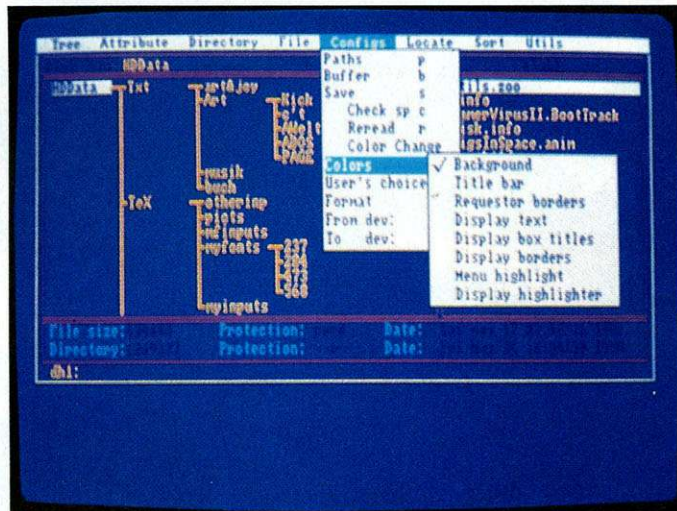


Bild 7. Für die MS-DOS-Begeisterten kann ein Hilfsmittel wie »Atree« den Umstieg erleichtern (Fish 289)

sen und Schreiben von MS-DOS-Disketten und -Festplatten (!) im 3,5-Zoll- oder 5,25-Zoll-Format. Dabei erledigt es seine Aufgabe genauso gut oder noch besser als beispielsweise kommerzielle Programme wie »Dos-2-Dos« oder »Crossdos«.

»BBCChampion« (**Fish 298**) archiviert für Sie die manchmal kostbaren Bootblöcke oder Boottracks kommerzieller Programmdisketten, die durch Virenbefall leicht zerstört werden könnten – haben Sie sie vorher archiviert, können sie in einem entsprechenden Fall einfach zurückkopiert werden, und die Programme bleiben so, trotz kurzzeitigen Virenbefalls, brauchbar. Manchmal liegt es jedoch auch an einem falsch eingestellten Laufwerk oder an nicht ganz sauber beschriebenen Disketten, daß sich ein Programm nicht mehr laden läßt. Hier hilft unter Umständen »TrackSalve«, das nicht nur alle bekannten Fehler des »trackdisk.device« patcht, das heißt beseitigt, sondern auch noch andere Funktionen implementiert – so zum Beispiel ein automatisches »Verify« beim Schreiben beliebiger Daten, so daß ein Schreib-Lesefehler oder eine defekte Diskette wesentlich schneller erkannt und andererseits besser verhindert werden kann. »TrackSalve« ist auf **Fish 312** zu finden.

Eines der interessantesten Programme für Leute, die immer alles genau wissen möchten, ist »Xoper« von **Fish 318**. Dieses Programm ist einer der derzeit umfangreichsten Systemmonitore. Es zeigt nicht nur laufende Programme inklusive Pointer, Prioritäten

und Namen, sondern auch eingeladene Bibliotheken, Gerätetreiber, geöffnete Fenster und Bildschirme, geladene Zeichensätze und vieles mehr auf Wunsch an. »Xoper« kann per Knopfdruck im Hintergrund verschwinden, seine Informationen automatisch immer auf dem neuesten Stand halten und Programmen fiktive Signale senden, was vor allem für Programmierer interessant ist. Auch dies einer der Diamanten der PD-Szene, ebenfalls von einem deutschen Autor.

Wer viel mit Grafiken arbeitet, hat sicherlich auch schon nach einem zufriedenstellenden Tool für die Anzeige von IFF-Bildern gesucht. Einige können viel, aber nicht alle alles. Dies hat sich mit »Mostra« (**Fish 330**) anscheinend geändert. Keine Funktion, die aus einem anderen Anzeigeprogramm bekannt ist, ist nicht auch hier zu finden. Automatische Erkennung von HAM/EHB, Overscan, Zentrierung von NTSC-Bildern auf PAL-Bildschirmen, Script-Datei und vieles mehr machen Mostra zum Prinzen der Grafik-Anzeigeprogramme.

In den tiefen Winkeln der **Fish 333** (Unterverzeichnis Multiplot) versteckt ist ein weiteres Programm, das seinesgleichen sucht – auch im kommerziellen Markt. Mit »plt« steht Ihnen nach erfolgter Installation ein Handler zu Verfügung, der die Ausgabe von Grafiken in der Plottersprache HPGL (Hewlett Packard Graphics Language) auf jedem Preferences-Matrixdrucker erlaubt – auf Wunsch sogar in Farbe. Dabei müssen die Grafikdaten nicht einmal unbe-

PD-Utility	PD-Disk
ILBMLib	Fish 348
RKMCompanion	Fish 344
SKsh	Fish 342
SID	Fish 338
CManual	Fish 337
FileWindow	Fish 336
PPShow	Fish 334
PPMore	Fish 334
plt	Fish 333
LVR	Fish 331
Mostra	Fish 330
DiskSpeed	Fish 329
Msh	Fish 328
MRBackup	Fish 327
WhereIs	Fish 321
Mathtrans	Fish 319
Xoper	Fish 318
PKAZip	Fish 318
Lhwarf	Fish 318
TrackSalve	Fish 312
LHArc	Fish 312
BBCChampion	Fish 298
FmsDisk	Fish 294
Atree	Fish 289
VirusX 4.0	Fish 287
FastDisk	Fish 286
Dyna-Show	Fish 285
MachII	Fish 254
PowerPacker	Fish 253
MuchMore	Fish 253
DiskSalv	Fish 251
ASDG-rrd	Fish 241
Runback	Fish 240
KwikBackup	Fish 234
MyMenu	Fish 225
SetCPU	Fish 223
VirusX 3.2	Fish 216
KeyMapEd	Fish 193
IFF-Disk	Fish 185
SetFont	Fish 182
Browser	Fish 180
Conman	Fish 165
Zoo	Fish 164
NewZap	Fish 164
ScreenX	Fish 158
Surveyor	Fish 151
TurboBackup	Fish 139
AmiCron	Fish 139
DosKwik	Fish 129
FillDisk	Fish 61
MoreRows	Fish 54
ARP	TBAG 31
Performance	Kick 250

Tabelle 2. Super-Utilities auf einen Blick

dingt in eine eigene Datei geschrieben und dann ausgedruckt werden, sondern können direkt an das neue AmigaDOS-Gerät »PLT:« gesendet werden. Dieses rechnet die Plotterbefehle in eine Matrix um und gibt sie auf dem angeschlossenen Drucker aus. Sowohl Farbe wie auch fast alle anderen Befehle von HPGL werden unterstützt.

Ein komfortables und mit zahlreichen Funktionen ausgerüstetes, mausgesteuertes Dateikopierprogramm, das nebenbei noch zahlreiche andere Programme aufrufen, Da-

PD-Applikation	PD-Disk
PCQ	Fish 339
AnalytiCalc	Fish 328
Drawmap	Fish 315
A68k	Fish 314
UUCP	Fish 313
Elements	Fish 297
MandelMountains	Fish 295
UEdit	Fish 286
Dme	Fish 284
LPE	Fish 277
HP11	Fish 274
Image-Ed	Fish 242
MandelVroom	Fish 215
MicroEMACS	Fish 195
Rim	Fish 143
SBProlog	Fish 140/141
MicroSPELL	Fish 101
A-Render	Fish 99
Draco	Fish 76/77
mCAD	Fish 74
Vdraw	Fish 52
Blink	Fish 40
MC68010	Fish 18
JRCComm	Mailbox

Tabelle 3. Wo ist welche Applikation zu finden?

teiarhive entpacken und vieles mehr kann, findet sich mit »SID« auf **Fish 338**. Und wem der CLI mit dem oben erwähnten »Conman« noch nicht komfortabel genug ist, sollte Conman mit »SKsh« von **Fish 342** verbinden. So stehen Ihnen dann auch Programm-Piping (Weiterleitung der Ausgabe eines Programms als Eingabe zu einem weiteren), Befehls-Abkürzungen (Alias), residente Kommandos und vieles mehr zur Verfügung, was Sie in der Shell der Workbench 1.3 nicht finden.

Wie die erwähnte »IFF-Disk« auch enthalten die **Fish-Disks 337/344** ebenfalls größtenteils Texte statt Programme. Auf der erstgenannten Disk findet sich ein kompletter Kurs über die Programmierung des Amiga in der Sprache C mit über 200 Seiten, neun Kapiteln und unzähligen Beispielprogrammen, geeignet für Einsteiger und Fortgeschrittene. Diskette 344 enthält sämtliche Beispielprogramme und Sources der neuesten Version der offiziellen Handbücher zum Amiga (ROM-Kernal-Manuals, Band I & II, Addison Wesley Verlag).

Die zwei letzten Highlights aus dem Utilities-Bereich befinden sich nicht auf Fish-Disks, sondern auf Disketten der Reihe **TBAG** und **Kickstart**. »ARP«, das AmigaDOS Resource Project, ersetzt nicht nur alle CLI-Befehle durch neue, kürzere und leistungsfä-

higere, aber weiterhin kompatible Äquivalente. Das Programm stellt auch eine Funktionsbibliothek namens »arp.library« zur Verfügung, die im Laufe der letzten Zeit von mehr und mehr PD-Programmen als Voraussetzung angesehen wird, da sie dem Programmierer bei der Software-Erstellung viel Arbeit abnimmt. »ARP« ist inklusive der umfassenden Dokumentation auf **BAG 31** zu finden, Erweiterungen zur Grundausrüstung finden sich auf vielen Fish-Disks.

Um die durchschnittliche Leistung und Belastung eines Systems zu testen, ist derzeit am besten »Performance« von **Kickstart 250** geeignet. Es mißt nicht nur die Rechenleistung eines Systems als eine Zahl, sondern gibt mehrere verschiedene Werte (Betriebssystem-Performance, Mathematik-Performance und so weiter) als Ziffern und grafische Balken aus. Das erhöht die Übersichtlichkeit und steigert den Nutzen eines solchen Tools.

PD-Applikationen

Applikationen sind komplexere Programme und Anleitungen für direkte Anwendungen. Das unterscheidet sie von kleineren Hilfsprogrammen wie Utilities oder Tools. Naturgemäß finden sich bei weitem nicht so viele Applikationen auf Basis der PD, wie Hilfsprogramme zu finden sind. Doch kann mit dieser Übersicht hoffentlich dem Vorurteil entgegengetreten werden, daß es entweder so etwas in der PD nicht gäbe oder, wenn doch, daß es qua-

litativ weitaus minderwertiger sei als vergleichbare kommerzielle Programme.

Wie man seinen Amiga kostengünstig etwas mehr Rechenleistung verschafft, zeigt **Fish 18** mit einer Anleitung zum Einbau eines 68010-Prozessors. Hier wird detailliert der Einbau, die Installation und die Anwendung eines mitgelieferten Patch-Programms beschrieben, das dafür sorgt, Unterschiede zwischen dem 68000- und 68010-Prozessor auszugleichen. Als Vorteil erhalten Sie für knapp 40 DM eine um etwa 7 bis 10 Prozent höhere Rechenleistung.

Programmierer ist »Blink« von **Fish 40** inzwischen zur Genüge bekannt, wird er doch auch als offizieller Linker dem Lattice-C-Compiler beigelegt. Er ist um vieles schneller und leistungsfähiger als der anfangs weitverbreitete »Alink« und wird mit umfassender Dokumentation ausgeliefert sowie ständig überarbeitet und weiter verbessert. Eine komplette Programmiersprache findet sich auf **Fish 76/77**. Mit »Draco« geht der Programmierer vollkommen neue, für den Amiga und dessen Programmierung jedoch geradezu prädestinierte Wege. Eine umfangreiche Dokumentation führt in diese neue Sprache ein.

Für den Lernwilligen finden sich übrigens auf zahlreichen nachfolgenden Fish-Disks Programme, die vollkommen in »Draco« entwickelt und geschrieben wurden – Sie sehen also, daß Sie es hier keineswegs mit einer Spielzeug-Programmiersprache zu tun haben.

»VDraw« von **Fish 52** ist ein vollständiges Malprogramm mit allen von anderen Programmen her bekannten Grundfunktionen. Frei definierbare Bildschirmausschnitte, zahlreiche Werkzeuge, Musterfüllfunktionen und vieles mehr machen das Programm zu einer preiswerten Konkurrenz anderer Malprogramme. Ebenso verhält es sich mit »mCAD«, einem schnellen CAD-Programm, das eine – auch im kommerziellen Bereich – nicht gekannte Ausgabequalität auf 9-Nadel-Druckern besitzt. Es kann durch spezielle Bibliotheken auch zur Erstellung schematischer Schaltpläne eingesetzt werden, arbeitet jedoch erst ab 1 MByte Arbeitsspeicher vernünftig. Finden können Sie es auf **Fish 74**.

»A-Render« (**Fish 99**) ist ein komplettes Raytracing-System, dem es jedoch an einer Intuition-Oberfläche sowie einem Objekt-Editor mangelt. Trotzdem ermöglicht es die Erstellung von Grafiken, die teilweise denen kommerzieller Programme wie »Reflections«, »Sculpt« oder »TurboSilver« nicht nachstehen. Für einen Einstieg in die faszinierende Welt des Raytracing ist vielleicht gerade dieses Programm genau das richtige für Sie.

Die neueste Version des Editors »MicroEMACS« von **Fish 195** ist nicht nur auf dem Amiga verbreitet. Auf **Fish 23** finden sich Source-Codes für Systeme wie Atari ST, MS-DOS, Unix oder CP/M. Der Vorteil liegt darin, daß Sie nur einmal lernen müssen, das Programm zu bedienen, um auf all diesen Rechnern sofort mit dem Editor zurechtzukommen. Interessant in diesem Zusammenhang ist auch die Tatsache, daß »MicroEMACS« in immer mehr und mehr Mailboxen als komfortabler Full-Screen-Editor Anwendung findet, so daß man nicht mehr an die Zeileneditoren betagter Mailboxen gebunden ist. Der Editor selbst bietet auf dem Amiga komfortable Menüs, Tastaturabkürzungen für alle Funktionen, einen großen Textspeicher, schnelle Suchen/Ersetzen-Funktionen und eine Vielzahl anderer Features. In neueren Paketen des Betriebssystems Version 1.3 ist er auch auf der Extras-Diskette enthalten, dort jedoch nur mit spärlicher Anleitung beschrieben. Zu »MicroEMACS« passend gibt es auf **Fish 101**

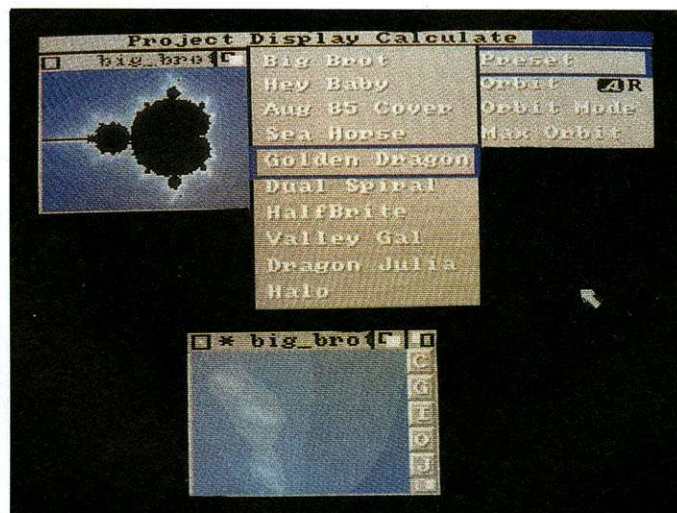


Bild 8. Exkursion in die Computergrafik. Mit »MandelVroom« (Fish 215) ist das kein Problem

das Programm »MicroSPELL«, das den Editor um eine englische Rechtschreibprüfung ergänzt. Mit ein wenig Arbeit läßt sich diese jedoch auch auf die deutsche Sprache »umbauen«.

Andere gute PD-Editoren sind »DME« von **Fish 284** und »UEdit« von **Fish 286**. Beide bieten AREXX-Anwendungen (-Ports) und sind in großem Maße programmierbar. »UEdit« erlaubt sogar die vollkommen freie Definition seiner Menüeinträge, »DME« wird auf vielen folgenden Fish-Disks mit zahlreichen Hilfsprogrammen und AREXX-Macros unterstützt.

Eine weitere Programmiersprache finden Sie auf **Fish 140/141**. Mit »SBProlog« steht ein leistungsfähiger Prolog-Compiler zur Verfügung, der im Bereich der PD auch auf anderen Rechnern seinesgleichen sucht. Auch **Fish 143** bietet mit »RIM« etwas Interessantes. »RIM« ist eine komplette relationale Datenbank mit fast unbegrenzten Datensatzzahlen. Zwar ist sie nicht mausgesteuert und besitzt keine Intuition-Oberfläche, doch zeichnet sie gerade dieser Punkt aus, da eine Portierung auf andere Systeme so wesentlich einfacher ist und nicht nur die Daten, sondern auf Wunsch auch das Programm relativ schnell zwischen verschiedenen Systemen ausgetauscht werden kann.

Eines der umfangreichsten Programme zur Erforschung der mathematischen Grafiken der Mandelbrot- oder Apfelmännchen-Menge, ist mit »MandelVroom« auf **Fish 215** erschienen. Vollkommen mausgesteuert führt es komfortabel durch die Mandelbrot-Menge, strengt Zooms an, verwendet schnellere Prozessoren, erlaubt die freie Wahl der Farben, der Farbübergänge, der Berechnungstiefen und so weiter. Ein Muß für alle Mandelbrot-Interessierten. Möchten Sie jedoch nicht nur die normalen Abbildungen der Mandelbrot-Menge sehen, befassen Sie sich mal mit »MandelMountains« von **Fish 295**. Es erstellt fantastisch aussehende dreidimensionale Darstellungen der üblichen Mandelbrot-Menge. Faszinierende Bilder wie aus einem Science-fiction-Film, einer Mondlandschaft oder anderen futuristischen Bereichen sind mit diesem Programm nur noch eine Frage der Rechenzeit und des Forscherdrangs.

Icons für die Workbench zu erstellen, ist ein recht kniffliges Unterfangen, sind Sie dabei auf den mitgelieferten »IconEd« angewiesen. Kostengünstige Alternative: »ImageEd« von **Fish 242**. Er ermöglicht nicht nur die Erstellung beliebig großer Icons, sondern bietet dafür auch noch zahlreiche Werkzeuge vom Kreis bis hin zur Füllfunktion an. Bis zu 32 Farben können Sie verwenden, falsche Befehle mittels »Undo« rückgängig machen und vieles mehr.

Besitzer des wissenschaftlichen Taschenrechners »HP11« von Hewlett Packard wird es freuen, daß auf **Fish 274** unter dem gleichen Namen eine fast hundertprozentige Emulation dieses Rechners für den Amiga zu finden ist. Natürlich wurden nicht nur alle wissenschaftlichen Operatoren, sondern auch der Programm-Modus dieses Rechners implementiert, was die Entwicklung von Programmen auf dem Amiga ermöglicht.

Arbeiten Sie mit dem Satzprogramm »TeX«? Vielleicht sogar mit »LaTeX«? In diesem Falle ist »LPE« von **Fish 277** sicherlich ein Leckerbissen für Sie. Hier können per Maus strukturierte Grafiken komfortabel und schnell gezeichnet werden, bevor man sie mittels einer Programmfunktion in »LaTeX«-Sourcecode übersetzen lassen kann. Vorbei sind somit die Zeiten der zahlreichen Probelaufe mit dem »TeX«-Compiler, bevor eine Grafik so aussah, wie sie es sollte. Ein Programm, das nicht nur Zeit spart, sondern ganz besonders auch die Nerven schont.

PD-Spiel	PD-Disk
Car	Fish 336
CRobots	Fish 331
Empire	Fish 329
Tetris	Fish 324
AmiOmega	Fish 320
AmigaTrek	Fish 320/278
ChinaChallenge	Fish 312
Go	Fish 289
EFJ	Fish 259
TrekTrivia	Fish 252
Monopoly	Fish 251
BallyIII	Fish 232
Moria	Fish 194
Hack	Fish 62

Tabelle 4. Die besten Spiele im Überblick

Vor allem für Schüler interessant dürfte »Elements« von **Fish 297** sein. Es beinhaltet das komplette Periodensystem der chemischen Elemente, stellt dieses auf dem Bildschirm dar und kann per Mausclick mit vielen zusätzlichen Informationen für jedes Element aufwarten. Darunter sind nicht nur übliche, wie Siedetemperatur oder das Gewicht, sondern auch etwas ausgefalleneres, wie die Atomkonfiguration oder das Verhalten in magnetischen Feldern. Einziger Nachteil ist die Tatsache, daß alle Informationen in Englisch sind und die meisten ohne Wörterbuch in diesem Bereich wohl nicht auskommen werden.

Auch für Schüler interessant ist »DrawMap« von **Fish 315**. Es baut aufgrund vermessungstechnischer Daten des CIA eine Weltkarte in verschiedenen Projektionen (Mercator-Projektion, als Kugel und so weiter) aus jedem

beliebigen Blickwinkel auf. Zooms sind möglich und offenbaren die Genauigkeit, mit der Kontinente und Küsten dargestellt werden.

Wer von Ihnen mit »UseNet«, dem amerikanischen Datenbank-System, zu tun hat, wird sich entweder über »UUCP« freuen oder es schon kennen. Speziell für dieses System zugeschnitten, ermöglicht es die zeitsparende Übertragung aller neuen Meldungen und Programme, die dann offline, das heißt ohne kostenintensive Telefonanbindung, durchgelesen und beantwortet werden können. Diese Antworten können Sie dann ebenfalls zeitsparend als geschlossene Datei mit »UUCP« wieder an »UseNet« übermitteln. »UUCP« ist in der neuesten Version auf **Fish 313** zu finden.

»A68K« von **Fish 314** ist der wohl erfolgreichste Assembler aus dem Bereich der PD. Er kann sowohl von Geschwindigkeit als auch von der Leistungsfähigkeit her in einigen Punkten kommerziellen Produkten durchaus die Hand reichen oder gar überlegen sein. Auf alle Fälle ermöglicht er jedoch den günstigen Einstieg in die Welt der Assembler-Programmierung. Ebenso ist »PCQ« zu sehen, ein kompletter Pascal-Compiler mit nicht zu unterschätzenden Funktionen. Er erlaubt die direkte Verwendung der Funktionsbibliotheken des Amiga und ist auf **Fish 339** abgelegt; angesichts der »Vielzahl« von Pascal-Compilern auf dem Amiga-Software-Markt für viele von Ihnen sicherlich einen Blick wert.

Mit »AnalytiCalc« schließlich findet sich auf **Fish 328** eine erstaunlich komfortable, ausgereifte und brauchbare Tabellenkalkulation. Geschwindigkeit und Zellenanzahl sowie die möglichen verknüpfenden Operatoren sind nicht nur für den einfachen Hausgebrauch, sondern auch für geschäftliche Ansprüche vollkommen ausreichend. Wer noch nie mit einer Tabellenkalkulation gearbeitet hat, jedoch meint ein Problem zu haben, das sich mit einem solchen Programm lösen läßt, der sollte an »AnalytiCalc« keineswegs vorbeigehen. Den letzten Vertreter unserer Applikationen zu erhalten ist recht schwierig, darf er doch laut Vertriebsbedingungen nicht auf PD-Reihen verbreitet werden. Sicherster Weg, das



Bild 9. Das bekannte Brettspiel Monopoly, hier einmal als Computerversion (Fish 251)

fantastische Terminalprogramm »JRCComm« zu bekommen, ist daher die heimische Mailbox oder ein DFÜ-begeisterter Bekannter, der sicherlich von dem Programm schon gehört hat. In vielen Bereichen ist es jetzt schon allen bislang erschienen, kommerziellen DFÜ-Programmen überlegen – allen voran in der Übertragung von Dateien mit dem Z-Modem-Protokoll. Hohe Geschwindigkeiten verträgt es besser als manch anderes Programm und bietet eine hundertprozentige Emulation des VT100- und des ANSI-Terminals. Da bislang jedoch lediglich die Version 0.99 im Umlauf ist, darf man gespannt auf die fertige Version 1.00 warten, die noch im Laufe dieses Jahres erscheinen soll, bei Drucklegung dieses Artikels vielleicht schon erschienen ist.

PD-Spiele

Auch im Bereich der Spiele finden sich in der PD-Szene interessante Programme. Wie üblich kostengünstig und trotzdem gut, sind auch sie nicht so oft zu finden wie Tools oder Utilities, decken jedoch nicht nur den Bereich der taktischen Spiele, sondern auch den der Action-Unterhaltung ab.

»Hack« (Fish 62) und »Moria« (Fish 194) sind Portierungen bekannter Spiele der Unix-Welt. Beide sind äußerst komplex und mehr im taktischen Bereich angesiedelt. In tiefen Labyrinthen, sogenannten Dungeons, muß der Spieler gegen Monster kämpfen, Schätze und Zauberkraft suchen und schließlich einen Ausgang finden. Beide Programme erfreuen sich großer Beliebtheit und sind auch in einigen Mailboxen als Online-Spiele zu finden. Dort spielen dann mehrere Teilnehmer in einem Dungeon, was den Reiz noch erhöht.

»Bally III« von Fish 232 ist ein Action-Spiel, das auf der Idee des Spielhallenautomaten »Click« beruht. Mit einem Pixel, das einen Strich hinterläßt, müssen Sie den Bildschirm so abfahren, daß umrandete Bereiche eingefärbt werden können. Das Ganze wird durch herumfliegende Störenfriede erschwert, die Ihnen ein Leben nehmen, wenn Sie getroffen werden.

Eine grafisch sehr ansprechende und volle Implementation des bekannten »Mono-

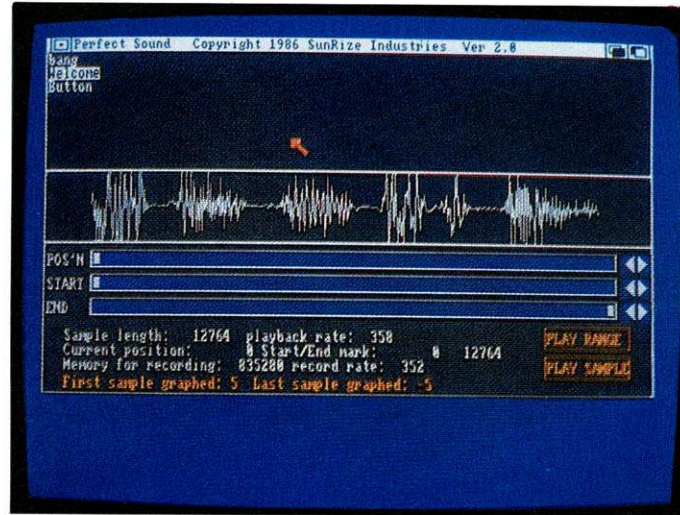


Bild 10. Digitizer-Software voll funktionsfähig (»Perfect Sound«, Fish 50)

poly« können Sie mit Fish-Disk 251 erhalten. Die verwendeten Bezeichnungen und Karten beziehen sich zwar alle auf das amerikanische Original, doch ist der Spielreiz der gleiche und mit mehreren Spielern macht die Computer-Variante sehr viel Spaß.

Wer das Spiel »Trivial Pursuit« kennt und zudem noch Star-Trek-Fan ist, dem wird »TrekTrivia« gefallen. Mit zahlreichen digitalisierten Sounds, über 100 Fragen aus dem Bereich der Star-Trek-Serie und der Möglichkeit, beim Autor zusätzliche Disketten mit Fragen zu erhalten, wird dieses Spiel nicht so schnell langweilig.

»EFJ«, oder ausgeschrieben »Escape From Jovi«, ist ein weiteres Action-Spiel des deutschen Autoren von »Bally III«. Mit einem Raumschiff müssen Sie einem Labyrinth aus Felsen und feindlich gesinnten Objekten entkommen. Weiches Scrolling und guter Sound zeichnen dieses Spiel aus.

Kein Spiel, aber trotzdem sehr unterhaltend sind die auf Fish 278/320 veröffentlichten Texte namens »AmigaTrek«. Im Stil der Star-Trek-Handlungen wird hier in insgesamt sechs Teilen die gesamte Computerszene – insbesondere die des Atari ST – auf die Schippe genommen. Ein ganz besonderer Leckerbissen, der schon vielen die Tränen in die Augen getrieben hat – den Amiga-Usern vor Lachen selbstverständlich.

Das Spiel asiatischen Ursprungs namens »Go« wird in einer PD-Version auf Fish 289

vertrieben. Dabei können Sie natürlich nicht nur mit einem computergesteuerten Spielbrett gegen einen Freund spielen, sondern auch gegen den Computer gewinnen beziehungsweise verlieren. Die Grafik ist gut und der Spielreiz ebenso groß, wie beim Original. Wie »Go«, ist auch »Mah Jongg« ein verbreitetes Spiel aus dem asiatischen Raum. Mit »China Challenge« steht es jedem Interessierten auf Fish 312 zur Verfügung. Nach bestimmten Kriterien muß eine Anhäufung von Spielsteinen entweder allein (im Stile einer gepflegten Patience) oder mit einem Partner zusammen abgebaut werden. Auf dem kommerziellen Markt ist ein ähnliches Programm unter dem Namen »Shanghai« zu finden.

Auch »AmiOmega« ist ein komplexes Spiel nach der Art von »Hack«. Es bietet jedoch noch weitaus mehr Orte, Monster, verschiedene Landschaften und viele, viele Zaubersprüche. Bedingt durch die Ausmaße benötigt es einen Amiga mit mindestens 1 MByte Arbeitsspeicher.

Kennen Sie »Tetris«? Wenn ja, wird es Sie freuen, daß eine grafisch gute und vor allem schnelle Umsetzung auf Fish 324 zu finden ist. Wenn nicht, sollten Sie sich dieses Action-Denk-Spiel einmal angucken. Von oben in ein rechteckiges Spielfeld hereinfallende, verschieden geformte geometrische Körper müssen durch Drehung und Positionierung so angeordnet werden, daß sich keine Lücken am Boden des Spielfeldes befinden. Ein

Spiel, das schon viele Benutzer süchtig gemacht hat, ist die Idee doch sehr einfach, die Faszination aber unbeschreiblich groß.

»Empire« (Fish 329) ist in der anfangs erwähnten Sprache »Draco« geschrieben und stellt ein komplexes Taktik-Spiel um Gewinn, Wirtschaft, Krieg und Entdeckertum dar. Es kann mit mehreren Spielern auf einem Rechner oder im Sinne eines Online-Spiels per Mailbox auch von mehreren Computern aus gespielt werden. Doch ist es nichts für das schnelle Zwischendruck-Spielen, kann es doch Monate dauern, bis ein Sieger feststeht.

Ein Spiel für Programmierer? Doch, so etwas gibt es. Bei »CRobots« von Fish 331 müssen Sie Ihre Spielfigur durch eine einfache, nur wenige Befehle umfassende Sprache mit einem Programm ausstatten, dessen Auswirkungen Sie dann in der zweiten Spielphase sehen können. Die Spielidee ist recht neu und der Amiga immer noch einer der ersten Computer, auf dem dieses Konzept realisiert wurde.

Der letzte hier vorgestellte Vertreter des PD-Spielebereiches soll »Car« sein. Ein realistisches, zweidimensionales Autorennen mit zehn verschiedenen Kursen, sehr guter Grafik und Ausnutzung des PAL-Modus. Auch dieses Spiel steht vergleichbaren kommerziellen Produkten weder in Ausführung noch im Spielgedanken nach.

Mit dieser Übersicht über die besten PD-Programme kann wohl allen negativen Vorurteilen über diese Software-Gattung wirkungsvoll entgegengetreten werden. Die PD-Szene des Amiga ist eine der besten, die es auf irgendeinem System gibt – sowohl von der Quantität, als auch von der Qualität her.

Wir hoffen, Ihnen mit diesem Leitfaden einen guten Start in die Welt der PD-Software gegeben zu haben, und würden uns über Meinungen, Tips und vor allem Tricks zu zahlreichen PD-Programmen freuen. Vielleicht ist ja auch für Sie ein Problem leicht lösbar, während es für andere Benutzer ein unüberwindbares Hindernis darstellt. Bis zu einer Veröffentlichung in unserem Tips&Tricks-Bereich ist es dann bei guten Vorschlägen nicht mehr weit...

(vb)

Oliver Wagner

Optimaler Code mit Lattice C

Mit dem Compiler auf du und du

Der Lattice-C-Compiler ist einer der leistungsfähigsten Compiler für den Amiga. Um die enorme Leistungsfähigkeit des Pakets zu nutzen, bedarf es allerdings einiger Einarbeitungszeit.

Der Compiler bietet die Möglichkeit, die Parameterübergabe zwischen Funktionen nicht über den Stapel abzuwickeln, wie es jeder andere Compiler tut, sondern dafür soweit als möglich die freien Daten und Adreßregister des 680x0 auszunutzen. Damit ergibt sich schon ein wesentlich kürzerer Code, da der Compiler effektivere Befehle zum Laden der Parameter erstellen kann, und ferner eine wesentlich höhere Geschwindigkeit, da der aufgerufenen Funktion die Parameter "mundgerecht" zur Verfügung stehen und sie diese nicht noch extra vom Stapel in die Datenregister laden muß. Aktiviert wird diese Funktion mit dem Schalter »-rr«, leider ergibt sich hier ein Pferdefuß. Die Registerübernahme ist so geregelt, daß die ersten beiden Zeigerparameter ins Adreßregister und die ersten Zahlenwerte ins Datenregister geladen werden. Nun kann der Compiler beim Aufruf ei-

ner Funktion prinzipiell nicht wissen, ob ein Parameter nun in ein Datenregister oder in ein Adreßregister geladen werden soll. Dazu muß zuerst ein Prototyp der Funktion erzeugt werden. Ein Prototyp ist eine Funktionsdeklaration ohne Funktionskörper. Beispiel: Für die Funktion

```
void print(s)
char *s;
```

```
{
Write(Output(),s,strlen(s))
}
```

sieht der Prototyp so aus:

```
void print(char*);
```

Positiver Nebeneffekt der Prototypen ist, daß der Compiler noch wesentlich sicherer als bisher Typenkonflikte (Fehlerquelle Nr. 1) anmahnen kann. Wenn man die Deklaration von Prototypen von Anfang an konsequent durchhält, handelt es sich auch nicht um einen nennenswerten Mehraufwand.

Ähnlich läuft der Aufruf von Betriebssystemfunktionen ab. Während normale Compiler Betriebssystemaufrufe wie normale Funktionsaufrufe behandeln und die aufgerufene Funktion erst den Sprung ins Kickstart durchführt, erzeugt der Lattice-C-Compiler direkte Sprünge in die Bibliotheken. Definiert werden solche Bibliotheksfunktionen mit einer »#pragma«-Anweisung, einer im ANSI-Standard verankerten Möglichkeit, systemspezifische Funktionen auszuführen. Keine Angst, mit der genauen Syntax der Anweisung braucht man sich gar nicht zu beschäftigen, für sämtliche Standard-Bibliotheken sind diese in den Include-Dateien zum Compiler definiert. Einfach im Listing

```
#include <proto/all.h>
```

eingeben, und schon ist man diese Sorge auch los. Auch das Linken mit der gigantischen »amiga.lib« kann man sich so ersparen.

In der Kürze liegt die Würze

Apropos linken: Die Bibliothek für kürzesten Code ist die »lcr.lib«. Hierzu muß, wie oben erwähnt, mit »-rr« kompiliert werden.

Die Prototypen für die Bibliotheksfunktionen sind zum Glück in den (normalerweise verschmähten) Standard-Include-Dateien

```
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
```

definiert, so daß man sich auch hier um nichts weiter kümmern muß.

Die Include-Datei »string.h« weist gleich auf eine weitere Eigenart des Lattice-C hin: Die sogenannten »BUILTIN«-Funktionen. Das sind Funktionen, die der Compiler bereits während der Compilierung durch lokalen Code ersetzt. Als wesentlichen Vorteil ergibt sich, daß der erzeugte Code genau auf die jeweilige Anwendung optimiert ist. So wird zum Beispiel bei einem »strlen()«-Aufruf mit einer Stringkonstante erst gar kein Code erzeugt, sondern die Länge bereits bei der Compilierung berechnet und eingetragen. Hiermit sind elegante Konstruktionen wie die oben bereits dargestellte Routine »print()« möglich.

Auch der absolute Code-Bläher, der Befehl »printf()«, existiert als »BUILTIN«. Der Compiler analysiert den Formatstring und ersetzt den Aufruf »printf()« jeweils durch »puts()« beziehungsweise »tinyprintf()«, falls dies möglich ist.

Die Funktionen »BUILTIN« erkennt der Compiler an einem dem Namen vorangestellten »_BUILTIN_«, aus »memcpy« wird also »_BUILTIN_memcpy«.

Aber auch hier braucht man sich keine Gedanken zu machen, das Einbinden der Standard-Include-Dateien (siehe oben) genügt.

Glanzpunkt des Pakets ist natürlich der sogenannte »Global Optimizer«. Dieser analysiert den Programmcode und führt jede Menge kleinere Optimierungen durch, indem er Funktionen als Ganzes analysiert und den Ablauf auswertet. Zusätzlich kann man anweisen, ob auf Codelänge »-ms« oder Geschwindigkeit »-mt« hin optimiert werden soll.

Der »Global Optimizer« ist bei gut geschriebenem Code, das heißt bei einem gut strukturierten Programm, nicht sehr effektiv. Er bringt aber viel, wenn man ohne Nachdenken drauflos programmiert hat beziehungsweise an einem Projekt schon länger feilt. Da dieser Compiler-Teil aber recht

Optionen

FROM files

TO file

WITH file

VER file

OVLYMGR file

LIBRARY files

XREF file

BUFSIZE n

NODEBUG o. ND

SMALLCODE o. SC

SMALLDATA o. SD

ONEDATA o. OD

FAST

CHIP

LIBFD file.fd

LIBPREFIX pfx

LIBID "Text"

QUIET

PRELINK

ADDSYM

NOALVS

MAXHUNK n

DEFINE s=s!v

VERBOSE

FASTER

BATCH

MAP datei FLAGS

PLAIN

FANCY

HWIDTH n

HEIGHT n

WIDTH n

FWIDTH n

PWIDTH n

SWIDTH n

Erläuterung

Objekt(„o.“)-Dateien, Synonym: ROOT

Zielfile

Batch-Datei abarbeiten

Ausgabe-File angeben

Overlay-Manager ersetzen (vergeßt es!)

Link-Libraries angeben. Synonym: LIB

Cross-Reference-Liste erzeugen

Interner Buffer

keine Debug-Information übernehmen
alle möglichen Code-Hunks zusammenziehen

alle möglichen Data und BSS-Hunks zusammenziehen

wie SD (?)

komplettes File nur ins Fast-Mem laden

komplettes File nur ins Chip-Mem laden

.fd-File für zu erzeugende Library angeben

Prefix für Library-Einsprünge angeben

ID-String für Library angeben

keine Ausgabe

keine Fehlermeldung für unaufgelöste Referenzen

Symbol-Hunks mit übernehmen

keine ALVS erzeugen

max. Hunk-Größe festlegen (für SC/SD)

Symbol festlegen; entweder Symbol oder Festwert

Status-Meldungen ausgeben

macht rein gar nix (ALINK-Kompatibilität)

Kein Nachfragen nach undefinierten Symbolen

Map erzeugen; FLAGS: H X S F L O

Keine Steuerodes im Map-File

Gegenteil von PLAIN [default]

Größe der Map

Größe der Map

Größe der Map

Größe der Map

Größe der Map

Größe der Map

Tabelle 1. Die Optionen des Linkers BLINK



InterComputing Deutschland Inc.

Schönebecker Str. 55-57
Telefon 0202/89155

5600 Wuppertal-2
Telefax 0202/89304

USA: InterComputing, Inc.

2112 Sandy Lane
Dallas, TX 75220

Telefon: 214-556-9666
Telefax: 214-556-2336

Frankreich: InterComputing France

34, Avenue des Champs Elysees
75008 Paris

Phone: (1) 42821603
FAX: (1) 42806649

MEMORY

8-UP Karte-2MB	699.-DM
8-UP Karte-4MB	999.-DM
8-UP Karte-6MB	1399.-DM
8-UP Karte-8MB	1799.-DM
Starboard II für A1000	
mit 512K	499.-DM
mit 1MB	649.-DM
M501 Erweiterung mit 512K/ batteriegepufferter Uhr	179.-DM

VIDEO

Digi View 4.0	329.-DM
Flicker Fixer (PAL)	1199.-DM
Vid Tech	
Scanlock	1999.-DM

LAUFWERKE/ FESTPLATTEN

Internal 3.5 laufwerke für A2000	199.-DM
Cutting Edge MAC laufwerke für AMAX	449.-DM
Quantum Hardcard 40MB	1299.-DM
Quantum Hardcard 80MB	1999.-DM
HardFrame SCSI Kontroller	499.-DM
GVP SCSI Kontroller mit raum für 8MB Speicher	499.-DM
Supra Wordsync Karte	399.-DM

GVP 68030 KARTEN

GVP 68030 - 28
68882 / 4MB
4799.-DM

GVP 68030 - 33
68882 / 4MB
6399.-DM

GVP 68030 - 50
68882 / 4MB
8999.-DM

SOFTWARE

3D Professional	745.-DM
Turbo Silver	249.-DM
Disney's Animation Studio	299.-DM
Saxon Publisher	649.-DM
Scene Generator	79.-DM
Pro Write 3.0	299.-DM
Deluxe Paint III	249.-DM
Lattice 5.0	499.-DM



AMIGA 500

Supra 40MB Quantum	1599.-DM
Supra 80MB Quantum	2199.-DM

AMIGA 1000

Supra 40MB Quantum	1699.-DM
Supra 80MB Quantum	2299.-DM

ICD ADRAM 540 mit 4MB

Ersetzt die reguläre 512K Erweiterung in Ihrem AMIGA 500!
999.-DM

Händler- und Auslandsanfragen erwünscht!

Bestellungen bitte schriftlich oder telefonisch an eines unserer Büros. Lieferung erfolgt gegen Vorkasse zzgl. DM 5.- Versandkosten oder per Nachnahme zzgl. DM 10.- Versandkosten. Kompletter Katalog gegen DM 3.50 in Briefmarken.

Sie können diese Produkte auch
bei den folgenden Händlern beziehen:

BERLIN:

Goldvision • 030 / 88 33 505
Prolinea • 030 / 26 18 387

WUPPERTAL: Hifi-Studio 9 GmbH • 0202 / 55 50 53

LC	LC1	LC2	Funktion	LC	LC1	LC2	Funktion
-ab		-cb	Chip BSS	-Lm			BLINK: math library anlinken
-ac		-cc	Chip Code	-Ln			BLINK: ND
-ad		-cd	Chip Data	-Lo			BLINK: overlay map erstellen
-b	-b		Base-Relative Data (A4)	-Ls			BLINK: symbol map erstellen
-b0	-b0		Non-Base-Relative Data	-Lt			BLINK: tiny modules (SC SD ND und so weiter)
-b1	-b1		wie -b	-Lv			BLINK: verbose
-c +	-c +		keine Strukturprüfung (für C + +)	-Lx			BLINK: cross reference map erstellen
-ca	-ca		Ansi-Kompatibilität	-l	-l		Objekte immer Langwortgrenzen ausrichten (BCPL-kompatibel)
-cc	-cc		Verschachtelte Kommentare möglich	-M			nur modifizierte Sources neu compilieren
-cd	-cd		Ermöglicht \$ in Identifiern	-m		-m	Code für 68000
-ce	-ce		Keine Fehlerausgabe	-m0		-m0	Code für 68000
-cf	-cf		Benötige Funktions-Prototypen	-m1		-m1	Code optimiert für 68010 (läuft auch auf 68000!)
-ci	-ci		Unterdrücke mehrfaches Include desselben Files	-m2		-m2	Code für 68020 (nur 68020 und höher)
-ck	-ck		Neue Keywords ermöglichen	-m3		-m3	Code für 68030 (auch 68020 und höher)
-cm	-cm		Ermöglicht Mehrzeichen-Konstanten (z.B. 'Olli')	-ma		-ma	Code für alle 680x0, optimiert für 68020
-co	-co		Alter Preprozessor	-mc		-mc	Stack-Aufräumreduzierung ausschalten
-cq	-cq		Keine Warnung bei Übergabe identischer Strukturen	-mr		-mr	Automatische Register-Variablen abschalten
-cr	-cr		Register-Parameter einschalten	-ms		-ms	Code auf Kürze hin optimieren
-cs	-cs		Identische Strings zusammenziehen; Strings ins Code-Segment	-mt		-mt	Code auf Geschwindigkeit hin optimieren
-ct	-ct		Warnung bei Struktur-Zeiger ohne Definition	-n	-n		Symbole nur 8 Zeichen unterscheiden
-cu	-cu		char ist unsigned	-O			Global Optimizer GO aufrufen
-cw	-cw		Keine Warnung für return ohne Rückgabewert	-ox		-ox	Object-Datei in x abspeichern
-cx	-cx		Alle globalen Deklarationen extern	-p		-p	Stack-Check Instruktionen erzeugen
-C			Bei Fehler weitermachen	-p	-p		Nur Preproces (LC1B !)
-d	-d		Debugging einschalten	-pe	-pe		Prototypes nur für externe Funktionen erzeugen
-d0	-d0		Debugging ausschalten	-ph	-ph		Precompilierte Header-Files erzeugen
-d1	-d1		Debugging: Linetable	-pp	-pp		Prototypen mit __PARMS erzeugen (Aztec-kompatibel)
-d2	-d2		Debugging: Symbol-Information	-pr	-pr		Prototypes erzeugen
-d3	-d3		Debugging: Symbol-Information, mit Zeilengrenze	-ps	-ps		Prototypes nur für static Funktionen
-d4	-d4		Debugging: volle Symbol-Information	-qx	-qx		Quad "q"-File in x abspeichern
-d5	-d5		Debugging: volle Symbol-Information, mit Zeilengrenze	-q <n> e	-q <n> e		Nach <n> Fehlern abbrechen
-dx = y			Preprozessor-Symbol x definieren	-q <n> w	-q <n> w		Nach <n> Warnungen abbrechen
-f	-f		Motorola FFP benutzen	-q			wie -q1e1w
-ff	-ff		wie -f	-r	-r		Alle Funktionsaufrufe NEAR (PC-Relativ)
-f8	-f8		Code für MC68881 erzeugen	-r0	-r0		Alle Funktionsaufrufe FAR (Absolute Adressierung)
-fi	-fi		Code für "math....libraries" erzeugen	-r1	-r1		wie -r
-fl	-fl		Normale Lattice-Libraries benutzen	-rr	-rr		Alle Aufrufe/Einsprünge defaultmäßig mit Register-Parametern
-gd	-gd		Cross Reference für Symbole	-rs	-rs		Alle Aufrufe/Einsprünge defaultmäßig mit Stack-Parametern (Standard)
-gc	-gc		Cross Reference der vom Compiler erzeugten Files	-rb	-rb		Einsprünge sowohl Register als auch Stack, Aufrufe Register
-ge	-ge		Ausgeklammerte Zeilen auflisten	-Rx			compilierte ".o"-Files in Library x abspeichern
-gh	-gh		Header-Files auflisten	-s		-s	Default-Segmentnamen benutzen (text, _____MERGED)
-gi	-gi		Macro-Expansionen auflisten	-sb = x		-sb = x	Name für BSS-Segment
-gn	-gn		Schmale Zeilen ausgeben	-sc = x		-sc = x	Name für Code-Segment
-gs	-gs		Source mit auflisten	-sd = x		-sd = x	Name für Data-Segment
-gx	-gx		Cross-Reference-Listing erstellen	-u	-u		Alle Preprozessor-Symbole abschalten
-Hx	-hx		Header(Include)-File x einlesen	-ux	-ux		Preprozessor-Symbol x abschalten
-hb		-hb	Fast BSS	-v		-v	Kein Stack-Checking-Code erzeugen
-hc		-hc	Fast Code	-w	-w		Defaultmäßig int = 16 Bit
-hd		-hd	Fast Data	-x	-x		Alle globalen Deklarationen wie extern behandeln
-ix	-ix		Directory für Include-Files angeben	-y		-y	A4 mit LinkerDB bei jedem Funktionseinsprung neu laden
-j <n>	-j <n>		Nachricht <n> ausschalten				
-j <n> e	-j <n> e		Nachricht <n> Fehler statt Warnung				
-j <n> i	-j <n> i		Nachricht <n> ausschalten				
-j <n> w	-j <n> w		Nachricht <n> Warnung statt Fehler				
-L +			Zusätzliche Linker-Dateien angeben				
-La			BLINK: ADDSYM				
-Lc			BLINK: SC				
-Ld			BLINK: SD				
-Lf			BLINK: ffp math library anlinken				
-Lh			BLINK: hunk map erstellen				
-Ll			BLINK: library map erstellen				

Tabelle 2. Eine Übersicht über die Kommandos des Lattice-C-Compilers

lange für seine Arbeit benötigt, sollte man sich Optimierungsvorgänge für die endgültigen Versionen eines Projekts aufheben.

Kommen wir nochmals auf das Linken zurück. Entscheidend über die Codelänge gerade kürzerer Programme ist der Startup-Code, der normalerweise jede Menge Dinge enthält, die im Programm gar nicht benötigt werden (Module-Anwender wissen ein Lied davon zu singen). Normalerweise beginnt ein Programm mit der Funktion

```
void main(int argc, char
**argv)
```

Der Startup-Code hat an diese Funktion sowohl die Parameter schön unterteilt als auch die Standard-IO-Routinen fertig initialisiert zu übergeben. Was aber, wenn man die IO-Routinen und die Parameter-trennung gar nicht benötigt? Dazu muß man wissen, daß der Startup-Code »c?.o« eigentlich eine Funktion

```
int _main(char *line)
```

anspringt, die in der jeweiligen Bibliothek »lc?.lib« definiert ist.

Name	Erläuterung
QUAD:	Platz für ".q"-Files
INCLUDE:	Default-Include-Directory
LIB:	Platz, wo ".lib"s abgelegt sind
LC:	Suchpfad für LC (ruft LC1, LC1B, LC2, GO auf)

Tabelle 3. Die logischen Geräte (ASSIGNS) auf einen Blick

niert ist. Erst diese Bibliothek öffnet die Ein-/Ausgabe (und das störende Fenster bei Start von der Workbench) und "parst" die Parameter. Hat man in seinem Programm aber eine eigene Funktion »_main«, springt »c?.o« direkt ins Hauptprogramm, und die recht umfangreiche Routine »_main« der Bibliothek wird gar nicht erst mitangelinkt. Das Argument »line« enthält übrigens einen Zeiger auf die Kommandozeile mit dem Programmnamen vorangestellt!

So weit, so gut. Was aber tun, wenn man zwar ohne die Standard-IO auskommt, auf das Parameter-Trennen nicht verzichten will? Nicht verzagen, »BLINK« fragen: In der »lc?.lib« existiert noch eine

zweite Funktion »_main«, genannt »_tinymain«, die genau dies tut. Mit einem

```
DEFINE @_main=@_tinymain
```

in der Linker-Kommandozeile erreicht man, daß »BLINK« statt der langen Routine die Kurzversion benutzt. Arbeitet man übrigens ohne Registerparameter, muß die Zeile

```
DEFINE _main=_tinymain
```

lauten. Noch ein Wort an Entwickler größerer Projekte: Der Compiler bietet eine einfache Möglichkeit, den erzeugten Code auf 68020-Prozessoren (oder höheren) zu optimieren. Warum nicht die Chance nutzen und eine zweite Programmversion für diese Prozessoren herausbringen? Die größer werdende Anzahl von Prozessorkarten und Ami-

ga-3000-Besitzern wird es zu schätzen wissen!

(vb)



Public Domain Studio Nürnberg GmbH

Werderstr.4, 8500 Nürnberg 20

Tel.: 0911 / 45 77 54 Fax: 0911 / 446 72 79

Ca. 10000 Public Domain + Shareware Disketten für Amiga, Atari ST und PC.

Gratis Katalog und Info für Ihren Computer anfordern !!!

DISKETTEN

3.5" 2DD

	10 St.	50 St.
Blau	DM 10.90	50.00
Grau	DM 10.90	50.00
Gelb	DM 17.90	85.90
Rot	DM 17.90	85.00
Grün	DM 17.90	85.90
Schwarz	DM 17.90	85.90

5.25"2D

	10 St.	100 St.
Schwarz	DM 5.90	54.90
Weiss	DM 9.90	94.90
Rot	DM 9.90	94.90
Gelb	DM 9.90	94.90
Grün	DM 9.90	94.90
Blau	DM 9.90	94.90

3.5 HD

Schwarz	DM 29.90	140.00
---------	----------	--------

5.25" HD

Schwarz	DM 12.80	124.00
---------	----------	--------

Diskettenboxen

DD80 für 80 Disketten 3.5" DM 14.90

DD100 für 100 Disketten 5.25" DM 14.90

Speichererweiterung für Amiga 500
512 KB mit Uhr DM 179.-

Versandkosten:

Nachnahme: DM 6.00 Vorkasse: DM 3.00

Warum nicht gleich bei uns?

TOP - Aktualität bei der PD!

Alle gängigen Serien und das Neueste von ACS über Fish bis Taifun. Weit über 7000 Disketten, mehr als 200 Reihen! Gute unbekannte Serien wie Chemie, Docs by Dr. Knox, Georgie's Music Show, TeamSoft, TuCs Digis. Vollgepackte deutsche Katalogdisk incl. Versand: DM 5.-

Alle Programme zum Buch "Public Domain Schatztruhe" von "technicSupport" vollständig auf 10 Disks nur DM 25,- !!

Die 10 besten PD-Disks eines Monats PD auf Qualitätsdisks!
aus allen Serien 2.80 DM / 3.5" 2DD
von uns zusammengestellt 1.80 DM / 5.25" 2DD
zum Abo-Preis von nur DM 25,- !!

TOP - Qualität bei der Hardware!

512 KB RAM für A500, Uhr + Abschalter	DM 129,-
3.5" LW extern, Chinon oder NEC, mit Bus	DM 179,-
5.25" LW extern, TEAC, mit Bus, 40/80 Tr.	DM 229,-
5.25" LW extern mit Schreibschutz + Bootsel.	DM 259,-
2 MB für A500, voll autokonfigurierend	DM 549,-
2 MB/8MB für A2000, voll autokonfigurierend	DM 649,-
HD 20 MB für A500, autoboot ab Kick 1.2	DM 949,-
HD 47 MB für A2000, autoboot ab Kick 1.2	DM 1099,-
Quantum HD für A2000, 42 MB, SCSI, 19 ms	DM 1199,-
Quantum HD für A500, 42 MB, SCSI, 19 ms	DM 1349,-
Quantum HD für A2000, 105 MB, SCSI, 19 ms	DM 1999,-
SyQuest Wechselplatte A2000, 44 MB, 25 ms	DM 1899,-

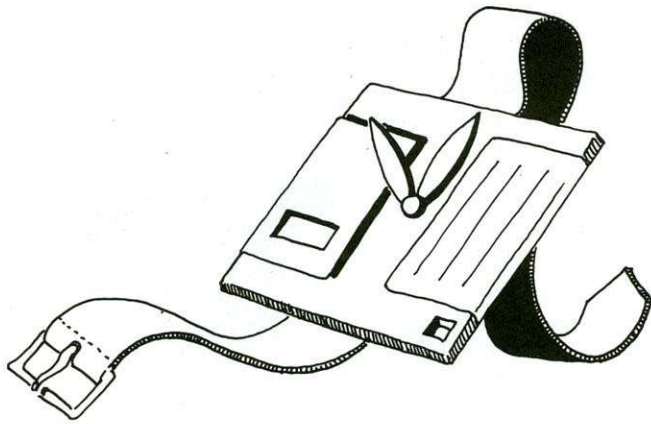
Versandkosten bei Public Domain und Hardware pauschal DM 6,- (Vorkasse) und DM 10,- (Nachnahme). Fordern Sie unsere aktuelle Preisliste an!

Gabriele von Thienen

Groß- und Einzelhandel für Computerzubehör

Postfach 100 648, 1000 Berlin 10

Tel: 030 - 322 63 68 bis 22.00 Uhr Fax: 030 - 321 31 99



Für Aztec-C-Benutzer ist das Programm »mclk« ein gewohnter Anblick. In einem Fenster auf der Menüleiste wird ständig der freie Speicher und die Uhrzeit angezeigt. Leider haben dieses Programm und seine Verwandten einige unangenehme Eigenschaften. Sie müssen mit »Run« vom CLI gestartet werden, wenn man das CLI weiterhin benutzen möchte. Dies verursacht nicht nur Tipparbeit, sondern verbraucht auch Zeit zum Laden von »Run«. Auch läßt sich das CLI nur schließen, wenn vorher »EndCLI« eingegeben wurde. Dies bewirkt aber den Abbruch der neuen Tasks. Viele dieser Utilities sind außerdem viel zu groß und fressen wertvollen Speicher und Rechenzeit.

In dem abgedruckten Listing finden Sie eine saubere, schnelle und effiziente Lö-

sung dieser Probleme. »MemClk« benutzt das »timer.device«, um sich alle 35 Mikrosekunden aus einem Tiefschlaf wecken zu lassen.

Kurz und effektiv dank Timer-Device

In der Zwischenzeit befindet sich der Task im Wait-Status. Während der aktiven Zeit wird die Uhrzeit und der freie Speicher ermittelt und formatiert in einem Window ausgegeben.

Starten läßt sich »MemClk« vom CLI oder von der Workbench. Beim Start vom CLI kann dieser weiterbenutzt oder geschlossen werden. Geben Sie das Listing mit Ihrem bevorzugten Editor (zum Beispiel ED von der Workbench) ein, und speichern Sie es unter dem Namen »memclk.c« ab. Nun können

Wolf-Jürgen Faust

Wem die Stunde schlägt!

»MemClk« - Speicher und Uhrzeit immer im Blick

Besitzern des Aztec-C-Compilers ist das Programm »mclk« mit Sicherheit bekannt. In einem kleinen Window wird ständig der verfügbare Speicher sowie die aktuelle Uhrzeit angezeigt. Leider haben »mclk« und vergleichbare Programme einige Mängel, die unser Listing professionell umgeht.

Sie das Programm sehr einfach mit »Execute memclk.c« compilieren. Sie sollten jedoch darauf achten, daß sich die Programme »Execute« und »Quit« im C-Verzeichnis Ihrer Startdiskette befinden. Als Compiler sollte der Lattice-C Version 5.04 verwendet werden. Es kann jedoch auch eine ältere Version vorliegen, doch durch die veränderten Prototypes der Version 5.04 werden beim Compilieren viele unnötige Warnungen ausgegeben.

Komfortables Compilieren

Das Compilieren mit einem Aztec-C-Compiler ist leider durch die verwendeten Prototypes und komplizierten Funktionsaufrufe (zum Beispiel »RawDoFmt«) nicht möglich.

(tb)

AMIGA DOS Info

Etwas Außergewöhnliches, gerade weil so einfach, hat sich der Programmierer von »MemClk« für den Compiler- und Linker-Aufruf einfallen lassen. Anstatt mühsam bei jedem Compilervorgang den Compiler und Linker aufzurufen, wurden diese Eingaben ganz einfach in den Kopf des Quellcodes integriert. Das erste Zeichen ist ein Semikolon. Dies teilt dem CLI mit, daß nun ein Kommentar folgt, nämlich das »/*«. Anschließend werden der Compiler, der Linker und das Programm »Quit« aufgerufen, welches wiederum das Ende der Batch-Datei anzeigt. Der Compiler wiederum überliest die CLI-Kommandos, da sie zwischen »/*« und »*/« stehen, was einen Kommentar im Quelltext darstellt.

Listings

```
1: ;/*
2: lc -j73 -v -cus -M -O memclk
3: blink FROM lib:cback.o+memclk.o TO memclk LIB lib:lc.1
4: quit
5: */
6:
7: /*-----*/
8: /*
9: /*      Urheber:      Wolf-Juergen Faust
10: /*
11: /*      West Germany
12: /*
13: /*      Zum compilieren dieses File unter dem Namen
14: /*      memclk.c speichern
15: /*      und mit "execute memclk.c" starten
16: /*
17: /*-----*/
18:
19: #include <exec/types.h>
20: #include <exec/memory.h>
21: #include <exec/tasks.h>
22: #include <devices/timer.h>
23: #include <libraries/dos.h>
24: #include <intuition/intuition.h>
25: #include <proto/all.h>
26: #include <string.h>
27:
28: #define INTUITION_REV 33
29:
Listing: MemClk
```

```
30: struct NewWindow New_Window = { 221,0,369,10,-1,-1,CLO
31: SEWINDOW,
32: WINDOWCLOSE | WINDOWDEPTH | WINDOWDRAG | SMART_REFRESH
33: | NOCAREREFRESH,
34: 0,0,"",0,0,0,0,0,0,WBENCHSCREEN };
35: struct Date Buffer[80];
36: struct IntuiText Date_Text = { 1,0,JAM2,0,0,0,Date_B
37: uffer,0 };
38: struct IntuitionBase *IntuitionBase;
39: struct Window *Window;
40: struct MsgPort *Timer_Port, *CreatePort();
41: struct IntuiMessage *Msg;
42: extern long _Backstdout; /* Filestream fuer Fehlerau
43: sgabe */
44: long BackGroundIO = 1; /* CLI Fenster fuer Fehle
45: rausgabe */
46: long stack = 2000; /* Stack fuer memclk */
47: char *procname = "memclk"; /* Prozessname */
48: long priority = 20; /* Task Prioritaet */
49: short data[5];
50:
51: void _regargs endall(char *);
52: void RawDoFmt(char *, short *, void (*)(), char *);
53: #pragma syscall RawDoFmt 20a ba9804
54: void _regargs prbuf(char c);
55: #define R_A3 (8+3)
56:
57: void _main()
58: {
59: struct DateStamp now;
60: if ((IntuitionBase = (struct IntuitionBase *)
```

Listing: MemClk


```

59: OpenLibrary("intuition.library", INTUITION_REV)
60: ) == NULL)
61: endall("Intuition library\n");
62: if ((Timer_Port = CreatePort("Timer Port",0)) == NU
63: LL)
64: endall("timer port\n");
65: if (!Time Req = (struct timerequest *) CreateExtIO(
66: Timer_Port, sizeof(struct timerequest)))
67: endall("timer request\n");
68: if (OpenDevice(TIMENAME, UNIT_VBLANK, (struct IOReq
69: uest *) Time Req, 0) != NULL)
70: endall("timer device\n");
71: if ((Window = (struct Window *) OpenWindow(&New_Win
72: dow)) == NULL)
73: endall("window");
74: Time Req->tr_node.io Command = TR_ADDREQUEST;
75: Close (_Backstdout);
76: _Backstdout = 0;
77: for (;;)
78: {
79: DateStamp(&now);
80: data[0] = (short)(AvailMem(MEMF_CHIP) >> 10);
81: data[1] = (short)(AvailMem(MEMF_FAST) >> 10);
82: data[2] = (short)(now.ds_Minute / 60);
83: data[3] = (short)(now.ds_Minute % 60);
84: data[4] = (short)(now.ds_Tick / TICKS_PER_SECO
85: ND);
86: RawDoFmt(" Chip:%3d Fast:%4d Time:%2d:%02d:%02
87: d ",data,prbuf,Date Buffer);
88: PrintIText(Window -> RPort, &Date_Text, 28, 1);
89: Time Req->tr_time.tv_secs = 0;
90: Time Req->tr_time.tv_micro = 350000;
91: SendIO((struct IORequest *) Time Req);
92: Wait(1 << Window -> UserPort -> mp_SigBit | 1 <<
93: Timer_Port -> mp_SigBit);
94: while (Msg = (struct IntuiMessage *) GetMsg(Wind
95: ow -> UserPort))
96: {
97: if (Msg->Class == CLOSEWINDOW)
98: {
99: ReplyMsg((struct Message *)Msg);
100: endall("");
101: }
102: ReplyMsg((struct Message *)Msg);

```

Listing: MemCik

```

98: }
99: GetMsg(Timer_Port);
100: }
101: }
102: }
103: /*-----*/
104: /* endall(why) - alles aufräumen vor dem Verlassen */
105: /* char *why = Zeiger auf Fehlermeldung */
106: /*-----*/
107: void _regargs endall(why)
108: UBYTE *why;
109: {
110: if (_Backstdout)
111: {
112: Write(_Backstdout, "\nCan't open ", 12);
113: Write(_Backstdout, why, strlen(why));
114: Close (_Backstdout);
115: }
116: if (Window) CloseWindow(Window);
117: if ((Time Req)&&(Time Req->tr_node.io_Device))
118: {
119: AbortIO((struct IORequest *)Time Req);
120: WaitIO((struct IORequest *)Time Req);
121: CloseDevice((struct IORequest *)Time Req);
122: if (Time Req) DeleteExtIO ((struct IORequest *)Time
123: Req);
124: if (Timer_Port) DeletePort(Timer_Port);
125: if (IntuitionBase) CloseLibrary((struct Library *)
126: IntuitionBase);
127: _exit();
128: }
129: }
130: /*-----*/
131: /* Dieser Code wird von der Kickstart-Routine
132: /* RawDoFmt fuer jedes
133: /* Zeichen im String aufgerufen
134: /* mit folgenden Parametern
135: /* D0 - naechstes zu formatierendes Zeichen
136: /* A3 - Zeiger auf Datenpuffer
137: /*-----*/
138: void _regargs prbuf(char c)
139: {
140: char *p = (char *)_builtin_getreg(R_A3);
141: *p++ = c;
142: _builtin_putreg(R_A3, p);
143: }

```

Listing: MemCik



Software-Autoren 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 einige Punkte beachten.

Wenn Sie nachstehendes befolgen, wird Ihre Post zügig und ohne große Rückfragen und Verzögerungen bearbeitet.

Senden Sie uns Ihr Programm mit

- allen benötigten Files auf der mit dem Programmnamen bezeichneten Diskette,
- den kompletten Ausdrucken/Listings aller Files der Diskette,
- einer Beschreibung Ihres Programms und
- 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 · Redaktion AMIGA DOS
Postfach 250 · 3440 Eschwege

Tabellenkalkulationen sollen dem Anwender die Arbeit abnehmen, komplexe Berechnungen wie Kosten/Leistung oder Abrechnungen zu erstellen. Solche Programme erlauben die Definierung von Zellen, die zur Berechnung herangezogen werden. Zellen sind nichts weiter als Rechenfelder, in die ein beliebiger Wert eingetragen wird oder die mit Hilfe einer Formel einen Wert errechnen. Solchermaßen errechnete Werte können für andere Rechnungen weiterverwendet werden. Als Beispiel stellen Sie sich bitte vor, eine Firma kauft und verkauft Waren und möchte nun die Vorsteuer von der Mehrwertsteuer abziehen. Dazu wird zuerst ein Feld für den Einkaufspreis definiert. Vom Einkaufspreis werden 14 Prozent in das Feld Vorsteuer gebracht. Vom Verkaufspreis werden ebenfalls 14 Prozent errechnet und in die Zelle Mehrwertsteuer gebracht. Nun muß nur noch der Differenzbetrag ermittelt werden und in das Differenzfeld übertragen werden. Die ganze Rechnung benötigt also 5 Zellen, von denen nur zwei bekannte Werte beinhalten (Einkaufspreis und Verkaufspreis), sowie 3 Rechenformeln (Vorsteuer aus Einkaufspreis, Mehrwertsteuer aus Verkaufspreis und Differenz aus Mehrwertsteuer und Vorsteuer). Bei einigen hundert Ein- und Verkäufen im Monat ist dies ein beträchtlicher Rechenaufwand, der ohne Kalkulationsprogramm recht lange dauern dürfte. Selbstverständlich stehen in der Realität sehr viel komplexere Rechnungen an.

Komplexe Rechnungen einfach aufbereitet

»EasyCalc« ist ein sehr einfach zu bedienendes, vollmaus- und menügesteuertes Tabellenkalkulationsprogramm in AmigaBASIC. Für einfache Anwendungen, wie eine individuelle, flexible Haushaltsabrechnung, einfache Buchführung, Statistiken, Rechnungen und anderes, ist

Guido Wimmel

EasyCalc

Eine Tabellenkalkulation für jeden

Wenn Sie sich öfters fragen, warum am Ende des Geldes soviel Monat übrig ist, oder wieviel der hauseigene Stubentiger im Jahr an Futter verschlingt, dann ist unser Programm »EasyCalc« genau das richtige für Sie. Ohne komplizierte Bedienungsanleitung wird es auch Ihnen gelingen, endlich Ordnung in das Chaos der Finanzen zu bringen.

»EasyCalc AMIGA« ein wirkungsvolles Hilfsmittel, das in vielen Fällen ein professionelles Tabellenkalkulationsprogramm ersetzen kann.

Leistungsmerkmale

»EasyCalc AMIGA« wurde bewußt in AmigaBASIC geschrieben, da jedem Amiga-Besitzer diese Sprache zur Verfügung steht und er sich so nicht unbedingt einen teuren C-Compiler oder ähnliches kaufen muß, um das Programm zu benutzen. Ferner verstehen die meisten Computerbenutzer BASIC, und das Programm kann leicht geändert werden. Wer einen Compiler für AmigaBASIC besitzt, dürfte keine Probleme haben, das Programm zu compilieren und so einen zusätzlichen Geschwindigkeitszuwachs zu erreichen.

Viel Leistung - wenig Programm

»EasyCalc AMIGA« besitzt folgende Leistungsmerkmale:

- vollständige Menü- und Maussteuerung
- Tabelle als ASCII-Datei speicherbar (Textprogramme)
- schnelle, optimierte Formelberechnung
- beliebige Schachtelung der Formeln möglich
- sehr kurzes und kompaktes Programm (nur zirka 400 Zeilen)

Um das Programm kurz zu halten, habe ich vorläufig die

Tabellengröße auf den Bildschirm beschränkt. Eine Erweiterung ist jederzeit möglich.

Bedienungsanleitung

»EasyCalc« wird wie jedes BASIC-Programm durch einen Doppelklick auf das entsprechende Icon gestartet (AmigaBASIC muß sich natürlich auf der Diskette befinden). Nach kurzer Ladezeit wird die Tabelle mit 25 Zeilen (1-25) und 7 Spalten (A-G) angezeigt. Der Cursor befindet sich auf Zelle A1 und kann durch die Pfeiltasten bewegt werden. Ein Mausklick auf die gewünschte Zelle bewegt den Cursor ebenfalls dorthin. Unter der Tabelle werden dabei Koordinaten und Typ der

jeweiligen Zelle angezeigt. Durch Drücken von [Backspace] oder [Enter] wird die Zelle, auf der sich der Cursor befindet, gelöscht. Bei einem Druck auf eine Zahlen- oder Buchstabetaste verschwindet der große Cursor, und die Zelle kann beschrieben werden. Dabei löscht [Backspace] das Zeichen links vom Cursor. Ein Druck auf [Esc] stellt den alten Zustand der Zelle wieder her und ein Druck auf [Enter] übernimmt den neuen Zelleninhalt. »EasyCalc« erkennt automatisch, ob eine Zahl oder ein Text eingegeben wurde und zeigt die Zelle dementsprechend an. Zur Unterscheidung wird das erste Zeichen der Zelle benutzt. Texte, die mit einer Zahl beginnen, sollten daher ein führendes Leerzeichen erhalten. Schreibt man bei der Eingabe einer Zelle über den rechten Rand hinaus, so wird die Zelle automatisch übernommen und der Cursor nach rechts bewegt. Dies vereinfacht die Eingabe von fortlaufenden Texten.

Kompatibel zu Textverarbeitungen

Um eine Formel einzugeben, fährt man mit dem Cursor auf die entsprechende Zelle und drückt [F1]. Danach kann man in der unteren Bildschirmzei-

Konstanten und Variablen

Konstanten:

mx,my:	Größe der Tabelle (7*25)
sx,sy:	Offset für Bildschirmanzeige (0,0)
form\$	Format der Zahl- und Formelzellen (# # # # # . # #) = > änderbar!
wmax,wmin:	Größter und kleinster darstellbarer Wert, sonst "?????????" = > änderbar
maxfunc:	Maximale Anzahl von Funktionen (100)
typ\$(0..3)	Typbezeichnungen (leer, Zahl, Formel, Text)

Variablen: (Hauptprogramm)

w(mx,my):	Zahleninhalte der Zellen 1E + 18 für Leer- und Textzelle sowie für ungültige Formelzelle 1E + 17 für noch nicht berechnete Formel (FuncCalc)
txt\$(mx,my):	Textinhalte der Zellen bei Formelzelle: Formel im Klartext bei Leer- und Zahlzelle: ""
typ(mx,my):	Typ der Zellen: 0 = leer, 1 = Zahl, 2 = Formel, 3 = Text
f(maxfunc,1):	Positionen der Formeln (0 = x; 1 = y)
f\$(maxfunc):	vorberechnete Formel (codiert)
fz:	momentane Anzahl der Formeln
x,y:	momentane Cursorposition
pgend:	Programmende-Flag
akfl:	AutoCalc-Flag
mve:	letzte Cursorbewegung
autmve:	AutoCurs-Flag (= Bewegungsrichtung)

Beispielformeln für »EasyCalc«

A1:A20	Summe von A1..A20
A1:B3	A1 + A2 + A3 + B1 + B2 + B3
A2:OB6	Durchschnitt von A2,B2,A3,B3..A6,B6
B6:<F6	Minimum von B6,C6,D6,E6,F6
B6:>F6	Maximum von B6,C6,D6,E6,F6

le eine Formel eingeben. Bei [Enter] ohne die Eingabe einer Formel kehrt das Programm ohne Veränderung zurück.

Format der Formeln

Formeln bestehen aus drei verschiedenen Elementen, die beliebig mit den vier Grundrechenarten kombiniert werden können.

Dies sind:

- Zahlen wie 3.14 oder 10023
- Zellen, die Zahlen oder Formeln beinhalten, zum Beispiel B12
- Bereiche:

»EasyCalc« kann Summen, Maxima/Minima und Durchschnitte von beliebigen rechteckigen Bereichen der Tabelle berechnen. Dabei werden die zwei Eckzellen des Bereichs durch einen Doppelpunkt getrennt eingegeben. Bei der Berechnung des Durchschnitts wird hinter den Doppelpunkt noch eine "0" gesetzt, bei Minimum und Maximum ein "<" beziehungsweise ">". Beispiele hierzu finden Sie in der Beispielfox.

Vorsicht! Geänderte Rechenhierarchie!

Bei Kombination dieser Elemente mit den vier Grundrechenarten ergeben sich zum Beispiel folgende gültige Formeln:

```
3+A1*B6: > F6
1.14*A1:A20/C7
```

Achtung! Das Programm beachtet nicht die Regel "Punkt vor Strich"! $1+3*A1$ wird wie $4*A1$ ausgerechnet! Man sollte also besser $3*A1+1$ schreiben. Außerdem dürfen keine Klammern gesetzt werden. Eine Formel, deren Wert nicht darstellbar ist, weil

- die Formel auf Text oder Leerzellen zugreift,
 - Formeln gegenseitig aufeinander zugreifen
 - der Wert außerhalb des zulässigen Bereichs liegt,
- wird mit "?????????" angezeigt.

Beschreibung der Menüpunkte

Im Menü »Projekt« befinden sich folgende Menüpunkte:

- »Löschen«:
Tabelle wird nach Sicherheitsabfrage gelöscht.
- »Laden«:
In der unteren Bildschirmzeile muß der Name der zu ladenden Datei eingegeben werden. [Enter] ohne Eingabe bricht die Operation ab, bei einem fehlerhaften

oder nicht existenten Dateinamen wird ein "Ein-/Ausgabefehler" angezeigt.

- »Speichern«:
Speichern der Tabelle (Handhabung wie Laden)
- »Sp.ASCII«:
Speichern der Tabelle im ASCII-Format zur Übernahme in Textverarbeitungsprogramme oder Editoren oder zum späteren Drucken (Handhabung wie Laden).
- »Drucken«:
Ausdrucken der Tabelle nach Sicherheitsabfrage.
- »Ende«:
Beenden des Programms nach Sicherheitsabfrage. Im Menü »Modus« befinden sich folgende Menüpunkte:
- »AutoCalc«:
Mit diesem Menüpunkt wird die automatische Formelberechnung ein- beziehungsweise ausgeschaltet. Der jeweilige Zustand läßt sich im unteren Teil des Bildschirms ablesen (»AutoCalc« ein/aus). Bei eingeschaltetem »AutoCalc« werden nach jeder Veränderung einer Zelle alle Formeln neu ausgerechnet, bei ausgeschaltetem »AutoCalc« geschieht dies nur vor dem Speichern und Drucken oder durch den Menübefehl »Berechnen«. Bei ausgeschaltetem »AutoCalc« kann man die Tabelle wesentlich schneller editieren.
- »AutoCurs«:
Mit diesem Menüpunkt kann man die automatische Cursorbewegung ein- beziehungsweise ausschalten. Der Zustand wird ebenfalls im unteren Teil des Bildschirms angezeigt. Um die automatische Cursorbewegung zu aktivieren, bewegt man den Cursor in eine bestimmte Richtung (mit den Pfeiltasten) und wählt dann »AutoCurs« an. Nach jeder Eingabe einer Zelle wird der Cursor dann automatisch in die vorher gewählte Richtung bewegt. Dies erleichtert die Eingabe von ganzen Spalten oder Reihen. Durch nochmalige Anwahl von »AutoCurs« wird diese Funktion wieder ausgeschaltet.
- »Berechnen«:
Bei Anwahl dieses Menüpunkts berechnet »EasyCalc« die Ergebnisse aller Formeln und zeigt sie in den vorgesehenen Zellen an. Dieser Menüpunkt soll-

te bei ausgeschaltetem »AutoCalc« angewählt werden, um die richtigen Ergebnisse zu erhalten.

Beliebige Änderungen möglich

Programmaufbau

Die wichtigsten Konstanten und Variablen des Programms und deren Funktionen erschauen Sie bitte aus der Konstantenbox.

Programmablauf

Nach Variableninitialisierung und Bildschirmaufbau schaltet das Programm den Cursor ein (»CurOn«) und wartet auf ein Ereignis (»SLEEP«). Während der Bearbeitung eines Ereignisses wird dafür gesorgt, daß der Cursor ausgeschaltet ist (»CurOff«). Der Programmteil »PrintCell« ist für die Darstellung einer Zelle im entsprechenden Format (Text beziehungsweise Zahl) auf dem Bildschirm zuständig. »CurOn« und »CurOff« wählen vorher die entsprechenden Farben (invers beziehungsweise normal) an, »CurOn« zeigt gleichzeitig Typ, Koordinaten und gegebenenfalls die beinhaltete Formel in der Statuszeile an. Es können drei verschiedene Ereignisse auftreten:

- Tastendruck = >
»KeyHandle«
- Mausklick = >
»MouseHandle«
- Menüanwahl = >
»MenuHandle«

»KeyHandle«:

Auswertung des Tastendrucks und Verteilung an die entsprechenden Programmteile (Formeleingabe [F1] => »FuncInp«, Cursortasten => »CurKeys«). Bei einer anderen Taste wird die Zelle editiert. Danach wird automatisch deren Typ anhand des ersten Zeichens festgestellt und der Zelleninhalt in das Array geschrieben. Bei eingeschaltetem »AutoCurs« oder einem Schreiben über den Zellrand wird noch eine Cursorbewegung ausgeführt, und bei eingeschaltetem »AutoCalc« wird die Funktionsberechnung (»FuncCalc«) aufgerufen.

»MouseHandle«:

Umrechnen der Mauskoordinaten in Zellkoordinaten. Bei gültigen Werten wird die Cursorposition neu festgelegt (x,y).

»MenuHandle«:

Festlegen des angewählten Menüpunkts (»title«, »poi«), Bearbeitung in Projekt, Modus. Bei Projekt und Modus wird anhand von »poi« weiter unterteilt in: Loeschen, Speichern, SpASCII, Laden, Drucken, Out, Ende. »SpASCII« und »Drucken« öffnen die entsprechende Datei, bei »Drucken« »LPT1:«, und erledigen die Ausgabe gemeinsam über »Out« auf Kanal #1. »Ende« setzt nur das Flag »pgend«, damit die While-Wend-Hauptschleife beendet wird. »Loeschen«, »Speichern« und »Laden« erklären sich von selbst.

»ErrHandle«:

In diesen Programmteil wird bei einem aufgetretenen Fehler verzweigt (ON ERROR). Bei Ein-/Ausgabefehlern (Codes 53-74) und Division durch null wird eine entsprechende Warnung erteilt und das Programm in der Hauptschleife fortgesetzt. Bei einem unerwarteten Fehler (zum Beispiel Tippfehler) wird ebenfalls eine Warnung erteilt und Gelegenheit gegeben, die Tabelle noch zu speichern.

»FuncInp«:

Eingabe einer Formel und Einreihen in die Formeltabelle.

»FuncClear«:

Löschen der Formel an x,y aus der Tabelle.

»PreCalc«:

Umwandeln der Formel t\$ in optimierten, schnell berechenbaren Code (f\$). Dadurch entfällt bei der Formelberechnung die komplizierte Suche nach Operanden, die Auswertung und Fehlerbehandlung etc.

»FuncCalc«:

Das eigentliche Berechnen aller Formeln. Das Programm berechnet solange alle Formeln, bis jede einen gültigen Wert hat. Bei der Berechnung der einzelnen Formeln wird die vorberechnete Formel aus f\$() benutzt. Auf deren Bytes wird aus Geschwindigkeitsgründen mit »fadr« = SADD (f\$()-1) und »byte« = PEEK (fadr + Position) zugegriffen. Dies ist eigentlich nur ein Ersatz für ASC(MID\$(f\$(), Position, 1)), was deutlich langsamer berechnet wird.

(tb)

Listings



Sprache: AmigaBASIC

```

1: REM EasyCalc AMIGA V1.0 (C) 1990 by G.Wimmel Soft Ltd.
2: REM Initialisierung und Bildaufbau
3: CLEAR ,40000&
4: DEFINT a-z:DEFSNG e,w
5: SCREEN 1,640,256,2,2
6: WINDOW 1,"EasyCalc AMIGA V1.0 by G.Wimmel",,0,1
7: maxx=7:maxy=25:maxfunc=100
8: form$="#####.##":wmax=999999.9:wmin=-99999.99
9: DIM txt$(maxx,maxy),w(maxx,maxy),typ(maxx,maxy)
10: DIM typ$(3),f(maxfunc,1),fs(maxfunc)
11: FOR i=0 TO 3:READ typ$(i):NEXT
12: DATA "leer","Zahl","Formel","Text"
13: FOR y=1 TO maxy:FOR x=1 TO maxx:
  w(x,y)=1E+18:NEXT:NEXT
14: FOR y=1 TO 25:LOCATE y+1,3:PRINT USING "##.":y:NEXT
15: FOR x=1 TO 7:LOCATE 1,(x-1)*10+7:
  PRINT CHR$(64+x);";":NEXT
16: LINE (0,211)-(624,211),1:LINE -(624,8)
17: x=1:y=1:pgend=0:sx=0:sy=0:fz=0:
  akfl=-1:mve=0:autmve=0
18: MENU 1,0,1,"Projekt":MENU 2,0,1,"":
  MENU 3,0,0,"":MENU 4,0,0,""
19: MENU 1,1,1,"Löschen":MENU 1,2,1,"Laden":
  MENU 1,3,1,"Speichern"
20: MENU 1,4,1,"Sp.ASCII":MENU 1,5,1,"Drucken":
  MENU 1,6,1,"Ende"
21: MENU 2,0,1,"Modus":MENU 2,1,1,"AutoCalc":
  MENU 2,2,1,"AutoCurs"
22: MENU 2,3,1,"Berechnen"
23: LOCATE 28,50:PRINT "AutoCurs aus";
  TAB(65);"AutoCalc ein"
24: ON ERROR GOTO ErrHandle
25: MLoop: REM Hauptschleife, Aktionsabfrage
26: GOSUB CurOn
27: WHILE NOT pgend:SLEEP
28: i$=INKEY$:IF i$<>" THEN GOSUB CurOff:
  GOSUB KeyHandle:GOSUB CurOn
29: m=MOUSE(0):IF m THEN GOSUB CurOff:
  GOSUB MouseHandle:GOSUB CurOn
30: title=MENU(0):IF title THEN GOSUB CurOff:
  GOSUB MenuHandle:GOSUB CurOn
31: WEND
32: MENU RESET
33: SYSTEM
34: END
35:
36: MouseHandle:
37: xx=MOUSE(1):yy=MOUSE(2)
38: xx=xx\8+1:yy=yy\8+1
39: xx=(xx-7)\10+1:yy=yy-1
40: IF yy>0 AND yy<26 AND xx>0 AND xx<8 THEN
  x=xx+sx:y=yy+sy
41: RETURN
42: ClrStat:
43: LINE (0,224)-(639,231),0,bf:LOCATE 29,1:RETURN
44: CurOn:
45: COLOR 1,0:LOCATE 28,1:PRINT CHR$(x+64);y;
  TAB(6);"- Typ: ";typ$(typ(x,y))
46: GOSUB ClrStat
47: IF typ(x,y)=2 THEN PRINT txt$(x,y);
48: COLOR 3,2:GOSUB PrintCell
49: RETURN
50:
51: CurOff:
52: COLOR 1,0:GOSUB PrintCell:RETURN
53:
54: PrintCell:
55: IF y<=sy OR y>sy+25 OR x<=sx OR x>sx+7 THEN RETURN
56: LOCATE (y-sy)+1,(x-sx-1)*10+7
57: typ=typ(x,y)+1
58: ON typ GOSUB leer,zahl,zahl,text
59: RETURN
60:
61: leer:
62: PRINT SPACES(10);:RETURN
63: zahl:
64: IF w(x,y)>wmax OR w(x,y)<wmin THEN
65: PRINT "?????????";
66: ELSE
67: PRINT USING form$;w(x,y);
68: END IF:RETURN
69: text:
70: PRINT txt$(x,y);:RETURN
71:
72: KeyHandle:
73: i=ASC(i$):IF i>=28 AND i<=31 THEN CurKeys
74: IF i=129 THEN FuncInp
75: IF i=8 THEN typ(x,y)=0:txt$(x,y)=":RETURN
76: t$="":yy=(y-sy)+1:xx=(x-sx-1)*10+6:rdy=0:first=-1
77: LOCATE yy,xx+1:COLOR 1,0:PRINT SPACES(10);
  ix=1
78: WHILE ix<11 AND rdy=0
79: LOCATE yy,xx+ix:COLOR 3,2:PRINT " ";
80: IF NOT first THEN
81: i$="":WHILE i$="":i$=INKEY$:WEND
82: ELSE
83: first=0
84: END IF
85: LOCATE yy,xx+ix:COLOR 1,0:PRINT " ";
86: i=ASC(i$)
87: IF i>=32 THEN
88: LOCATE yy,xx+ix:PRINT i$;
89: t$=t$+i$:ix=ix+1
90:

```

Listing: EASYCALC

```

91: ELSEIF i=8 THEN
92: ix=ix-1:IF ix=0 THEN ix=1
93: t$=LEFT$(t$,ix-1)
94: ELSEIF i=13 THEN
95: rdy=1
96: ELSEIF i=27 THEN
97: rdy=2
98: END IF
99: WEND
100: IF rdy=2 THEN RETURN
101: IF typ(x,y)=2 THEN GOSUB FormClear
102: t$=t$+SPACES(10-LEN(t$))
103: i=INSTR("0123456789.-",LEFT$(t$,1))
104: IF t$=SPACES(10) THEN
105: typ(x,y)=0:txt$(x,y)="":w(x,y)=1E+18
106: ELSEIF i>0 THEN
107: typ(x,y)=1:txt$(x,y)="":w(x,y)=VAL(t$)
108: ELSE
109: typ(x,y)=3:txt$(x,y)=t$:w(x,y)=1E+18
110: END IF
111: IF akfl THEN GOSUB FuncCalc
112: IF ix=11 THEN
113: GOSUB PrintCell:GOSUB CurRgt
114: ELSE
115: IF autmve THEN GOSUB PrintCell:
  ON autmve GOSUB CurUp,CurDn,CurRgt,CurLft
116: END IF
117: RETURN
118:
119: CurKeys:
120: mve=ASC(i$)-27:ON mve GOSUB
  CurUp,CurDn,CurRgt,CurLft:RETURN
121:
122: CurUp:y=y-1:IF y<=sy THEN y=25'DownWrap
123: RETURN
124: CurDn:y=y+1:IF y>sy+25 THEN y=1'UpWrap
125: RETURN
126: CurLft:x=x-1:IF x<=sx THEN x=7'RightWrap-
127: RETURN
128: CurRgt:x=x+1:IF x>sx+7 THEN x=1'LeftWrap
129: RETURN
130:
131: MenuHandle:
132: poi=MENU(1)
133: ON title GOTO Projekt,Modus
134:
135: Projekt:
136: ON poi GOTO
  Loeschen,Laden,Speichern,SpASCII,Drucken,Ende
137:
138: Loeschen:
139: GOSUB ClrStat:INPUT"Wirklich löschen (J/N) ?",m$
140: IF UCASE$(m$)<>"J" THEN RETURN
141: FOR x=1 TO maxx:FOR y=1 TO maxy
142: typ(x,y)=0:txt$(x,y)="":w(x,y)=1E+18:GOSUB PrintCell
143: NEXT y,x
144: fz=0:x=1:y=1
145: RETURN
146:
147: Speichern:
148: IF NOT akfl THEN GOSUB FuncCalc
149: GOSUB ClrStat:INPUT"Speichern - Dateiname: ",nam$
150: IF nam$="" THEN RETURN
151: OPEN nam$ FOR OUTPUT AS #1
152: PRINT #1,maxx:PRINT #1,maxy
153: FOR xx=1 TO maxx:FOR yy=1 TO maxy
154: PRINT #1,MKIS(typ(xx,yy));
  MKIS(w(xx,yy));txt$(xx,yy)
155: NEXT yy,xx
156: PRINT #1,fz
157: FOR i=0 TO fz-1
158: PRINT #1,MKIS(f(i,0));MKIS(f(i,1));
159: NEXT
160: CLOSE #1
161: RETURN
162:
163: SpASCII:
164: GOSUB ClrStat:
  INPUT"Speichern ASCII - Dateiname: ",nam$
165: IF nam$="" THEN RETURN
166: OPEN nam$ FOR OUTPUT AS #1
167: GOSUB Out:CLOSE#1:RETURN
168:
169: Laden:
170: GOSUB ClrStat:INPUT"Laden - Dateiname: ",nam$
171: IF nam$="" THEN RETURN
172: OPEN nam$ FOR INPUT AS #1
173: INPUT #1,maxx:INPUT #1,maxy
174: FOR x=1 TO maxx:FOR y=1 TO maxy
175: typ(x,y)=CVI(INPUT$(2,#1)):
  w(x,y)=CVS(INPUT$(4,#1))
176: LINE INPUT#1,txt$(x,y)
177: GOSUB PrintCell
178: NEXT y,x
179: INPUT #1,fz
180: FOR i=0 TO fz-1
181: f(i,0)=CVI(INPUT$(2,#1)):f(i,1)=CVI(INPUT$(2,#1))
182: t$=txt$(f(i,0),f(i,1)):GOSUB PreCalc:f$(i)=f$
183: NEXT
184: CLOSE #1
185: x=1:y=1
186: RETURN
187:
188: Drucken:
189: GOSUB ClrStat:INPUT"Wirklich drucken (J/N) ?",m$
190: IF UCASE$(m$)<>"J" THEN RETURN
191: OPEN "LPT1:" FOR OUTPUT AS #1

```

Listing: EASYCALC


```

192: GOSUB Out:CLOSE#1:RETURN
193:
194: Out:
195: IF NOT akfl THEN GOSUB FuncCalc
196: FOR yy=sx+1 TO sx+25
197:   FOR xx=sx+1 TO sx+7
198:     typ=typ(xx,yy)+1:ON typ GOSUB
       oleer,ozahl,ozahl,otext
199:   NEXT
200:   PRINT #1,""
201: NEXT
202: RETURN
203: oleer:
204: PRINT #1,SPACES(10);:RETURN
205: ozahl:
206: IF w(xx,yy)>wmax OR w(xx,yy)<wmin THEN
207:   PRINT #1,"????????? ";
208: ELSE
209:   PRINT #1,USING form$;w(xx,yy);
210: END IF:RETURN
211: otext:
212: PRINT #1,txt$(xx,yy);:RETURN
213:
214: Ende:
215: GOSUB ClrStat:INPUT"Wirklich verlassen (J/N) ?",m$
216: IF UCASE$(m$)="J" THEN pgend=-1
217: RETURN
218:
219: Modus: ON poi GOTO akalk,acurs,Berechnen
220:
221: Berechnen: GOSUB FuncCalc:RETURN
222: akalk:
223: akfl=NOT akfl
224: LOCATE 28,74:IF akfl THEN PRINT "ein"
       ELSE PRINT "aus"
225: RETURN
226:
227: acurs:
228: IF autmve THEN autmve=0 ELSE autmve=mve
229: LOCATE 28,59:IF autmve THEN PRINT "ein"
       ELSE PRINT "aus"
230: RETURN
231:
232: ErrHandle:
233: GOSUB ClrStat
234: IF ERR>52 AND ERR<75 THEN
235:   PRINT "Ein/Ausgabefehler aufgetreten !
       Tastendruck !"
236:   CLOSE:WHILE INKEY$="":WEND:RESUME MLoop
237: END IF
238: IF ERR=11 THEN
239:   PRINT "Division durch Null ! Tastendruck !"
240:   WHILE INKEY$="":WEND:erflag=-1:
       wert=1E+18:RESUME NEXT
241: END IF
242: PALETTE 1,1,0,0:PALETTE 0,0,0,0
243: PRINT "Programmfehler";ERR;
       "aufgetreten - Programm überprüfen ! ";
244: PRINT "Taste zum Speichern !":WHILE INKEY$="":WEND
245: GOSUB Speichern:STOP
246:
247: FuncInp:
248: GOSUB ClrStat
249: LINE INPUT"Formel: ",t$:IF t$="" THEN RETURN
250: t$=UCASE$(t$):txt$=t$:GOSUB ClrStat:GOSUB PreCalc
251: IF erflag THEN
252:   GOSUB ClrStat:
       PRINT "Fehlerhafte Formel - Tastendruck !"
253:   WHILE INKEY$="":WEND:GOTO FuncInp
254: END IF
255: IF typ(x,y)=2 THEN GOSUB FormClear
256: txt$(x,y)=txt$:typ(x,y)=2:w(x,y)=1E+18
257: f(fz,0)=x:f(fz,1)=y:f$(fz)=f$:fz=fz+1
258: IF akfl THEN GOSUB FuncCalc
259: RETURN
260:
261: FormClear:
262: FOR i=0 TO fz-1
263:   IF f(i,0)=x AND f(i,1)=y THEN
264:     FOR j=1 TO fz-2:f(j,0)=f(j+1,0):
       f(j,1)=f(j+1,1):f$(j)=f$(j+1):NEXT
       fz=fz-1
265:   END IF
266: NEXT
267: RETURN
268:
269:
270: PreCalc:
271: erflag=0:f$="":f1=1
272: WHILE f1 AND NOT erflag
273:   nop=0:f1=0
274:   FOR po=2 TO LEN(t$)
275:     nop=INSTR("+-*/",MID$(t$,po,1))
276:     IF nop THEN f1=po:po=1000
277:   NEXT
278:   IF f1 THEN ex$=LEFT$(t$,f1-1):
       t$=MID$(t$,f1+1) ELSE ex$=t$:t$=""
279:   GOSUB PrecExp:f$=f$+CHR$(nop)
280: WEND
281: RETURN
282:
283: PrecExp:
284: IF ex$="" THEN erflag=-1:RETURN
285: fc$=LEFT$(ex$,1)
286: IF INSTR("0123456789.+-",fc$) THEN
287:   f$=f$+CHR$(1)+MK$(VAL(ex$))
288: ELSEIF fc$>="A" AND fc$<="Z" THEN

```

```

289:   ipo=INSTR(ex$,".")
290:   IF ipo THEN GOSUB PrecCellRan ELSE GOSUB PrecCell
291: ELSE
292:   erflag=-1
293: END IF
294: RETURN
295:
296: PrecCell:
297: GOSUB TkCellCoo:f$=f$+CHR$(2)+CHR$(cx)+CHR$(cy):
       RETURN
298:
299: PrecCellRan:
300: lt$=LEFT$(ex$,ipo-1):rt$=MID$(ex$,ipo+1)
301: rtyp=INSTR("0<",<,LEFT$(rt$,1))
302: IF rtyp THEN rt$=MID$(rt$,2)
303: IF lt$="" OR rt$="" THEN erflag=-1:RETURN
304: ex$=lt$:GOSUB TkCellCoo:c1x=cx:c1y=c1y
305: ex$=rt$:GOSUB TkCellCoo:c2x=cx:c2y=c2y
306: IF c1x>c2x THEN SWAP c1x,c2x
307: IF c1y>c2y THEN SWAP c1y,c2y
308: f$=f$+CHR$(3+rtyp)+CHR$(c1x)+CHR$(c1y)
       +CHR$(c2x)+CHR$(c2y)
309: RETURN
310:
311: TkCellCoo:
312: cx=ASC(ex$)-64:cy=VAL(MID$(ex$,2))
313: IF cx<1 OR cx>maxx OR cy<1 OR cy>maxy THEN
       erflag=-1:RETURN
314: RETURN
315:
316: FuncCalc:
317: FOR fu=0 TO fz-1:w(f(fu,0),f(fu,1))=1E+17:NEXT
318: rdy=0:cnt=1:ox=x:oy=y
319: WHILE NOT rdy AND cnt<=fz
320:   LOCATE 29,1:PRINT cnt;rdy=-1
321:   FOR fu=0 TO fz-1
322:     x=f(fu,0):y=f(fu,1)
323:     IF w(x,y)=1E+17 THEN GOSUB CellCalc
324:   NEXT fu
325:   cnt=cnt+1
326: WEND
327: FOR fu=0 TO fz-1
328:   x=f(fu,0):y=f(fu,1):
       IF w(x,y)=1E+17 THEN w(x,y)=1E+18
329:   GOSUB PrintCell
330: NEXT
331: x=ox:y=oy
332: RETURN
333:
334: CellCalc:
335: fadd$=SADD(f$(fu))-1:ptr=1:erflag=0
336: GOSUB CalcExp:erg=wert
337: WHILE PEEK(fadd$+ptr)<>0 AND NOT erflag
338:   op=PEEK(fadd$+ptr):ptr=ptr+1
339:   GOSUB CalcExp
340:   ON op GOSUB plu,mi,mul,div
341: WEND
342: IF erflag THEN rdy=0:w(x,y)=wert ELSE w(x,y)=erg
343: RETURN
344: plu: erg=erg+wert:RETURN
345: mi: erg=erg-wert:RETURN
346: mul: erg=erg*wert:RETURN
347: div: erg=erg/wert:RETURN
348:
349: CalcExp:
350: typ=PEEK(fadd$+ptr)
351: IF typ=2 THEN
352:   cx=PEEK(fadd$+ptr+1):cy=PEEK(fadd$+ptr+2)
353:   GOSUB CellWt:ptr=ptr+3
354: ELSEIF typ>2 THEN
355:   c1x=PEEK(fadd$+ptr+1):c1y=PEEK(fadd$+ptr+2)
356:   c2x=PEEK(fadd$+ptr+3):c2y=PEEK(fadd$+ptr+4)
357:   ptr=ptr+5:rtyp=typ-2
358:   ON rtyp GOSUB Sum,Avg,Max,Min
359: ELSEIF typ=1 THEN
360:   wert=CVS(MID$(f$(fu),ptr+1,4)):ptr=ptr+5
361: END IF
362: RETURN
363:
364: Sum:
365: canz=0:we=0:FOR cx=c1x TO c2x:FOR cy=c1y TO c2y
366:   canz=canz+1:GOSUB CellWt:we=we+wert
367:   IF erflag THEN cx=1000:cy=1000
368: NEXT:NEXT:IF NOT erflag THEN wert=we
369: RETURN
370: Avg:
371: GOSUB Sum:IF NOT erflag THEN wert=wert/canz
372: RETURN
373: Max:
374: we=-1E+18:FOR cx=c1x TO c2x:FOR cy=c1y TO c2y
375:   GOSUB CellWt:IF wert<we THEN we=wert
376:   IF erflag THEN cx=1000:cy=1000
377: NEXT:NEXT:IF NOT erflag THEN wert=we
378: RETURN
379: Min:
380: we=1E+18:FOR cx=c1x TO c2x:FOR cy=c1y TO c2y
381:   GOSUB CellWt:IF wert>we THEN we=wert
382:   IF erflag THEN cx=1000:cy=1000
383: NEXT:NEXT:IF NOT erflag THEN wert=we
384: RETURN
385:
386: CellWt:
387: wert=w(cx,cy):
       IF wert=1E+18 OR wert=1E+17 THEN erflag=-1
388: RETURN
389:

```



Gewußt wie!

Die Seiten für Einsteiger und Fortgeschrittene

Kennen Sie die beste Medizin gegen Anfälle von Wie-ging-das-denn-noch-Fragen? Na klar, unsere Gewußt-wie-Ecke. Hier ist sie wieder, die erste Hilfe gegen Computerfrust.

Zentrierter Text

Manchmal steht man vor dem Problem, einen Text zentriert ausgeben zu müssen. In GFA-BASIC läßt sich dieses relativ leicht erreichen:

```
1 DEFFN x=(80-LEN(TRIM$(text$)))/2
2 text$=TRIM$("....das ist die Mitte")
3 PRINT AT (FNx,10);text$
```

Je nach Fenster-Typ kann auch ein anderer Wert als 80 in Frage kommen.

(Pana Plytas/jb)

"Blitz"-Tip

Bevor wir zu den Tips kommen, erst einmal ein Hinweis zum Tip von Jürgen Bielig aus AMIGA DOS 4/90 ("1 > 0"). Er

verwendet in der vorletzten Zeile seines Batch-Files den Befehl »Path Reset«. Dies ist jedoch ein grober Fehler, da dieser Befehl den Pfad auf das aktuelle Verzeichnis und das logische Verzeichnis »C:« zurücksetzt, auch dann, wenn man in der Startup-Sequence den Pfad gesetzt hat. Es gibt im AmigaDOS bis jetzt noch keine Möglichkeit, einen einzelnen Pfad wegzunehmen, also sollte jeder Benutzer des Batch-Files diesen Befehl entfernen.

Bildschirmblitz im AmigaDOS

Wenn Sie den berühmten Bildschirmblitz für Warnungen im AmigaDOS verwenden wollen, machen Sie einfach folgendes:

Schreiben Sie in der Shell den Befehl »COPY * TO Blitz«. Nach dem abschließenden Druck auf [Return] kommt jetzt kein Prompt, sondern jede Zeile, die jetzt eingegeben wird und mit [Return] endet, wird in die Datei »Blitz« kopiert. Schreiben Sie deshalb jetzt »ECHO « und drücken Sie danach die Tastenkombination [Ctrl] und [G]. Wenn Sie jetzt ein invertiertes "G" auf dem Bildschirm sehen, setzen Sie ein Anführungszeichen, und drücken Sie [Return]. Beenden Sie nun das Erstellen der Copy-Datei durch die Tastenfolge [Ctrl]+[\\]+[Return]. Durch Eingabe von »Execute Blitz« kommt man jetzt in den Genuß des "Blitzers". Durch Benutzen des Befehls »Alias« der Shell 1.3 kann man sich auch komplette Kommandofolgen mit Blitz (ohne Donner) unterlegen.

Multitasking total

Ideen kommen ja grundsätzlich dann, wenn man gerade mit was anderem beschäftigt ist. Wohin also mit den Gedankensprüngen? Ein Schreibblock ist nicht zu finden, und mit

Die Supertips

Primzahlen in BASIC

Folgendes kleine BASIC-Programm errechnet die Primzahlen von 1-15000 und gibt sie auf den Bildschirm aus:

```
1 DIM P(3000)
2 WIDTH 75
3 FOR z=1 TO 15000
4 FOR b=2 TO h
5 IF z/P(b)=INT(z/P(b)) THEN N
6 IF z < P(b)↑2 THEN b=h
7 NEXT b
8 h=h+1:P(h)=z
9 PRINT z,
10 N:NEXT z
```

Damit dürfte das Primzahlenproblem gelöst sein, bleibt nur noch das Problem, wie man den Amiga ins Klassenzimmer schmuggelt.

(Thomas Renner/jb)

Kleine Ursache, großer Screen

In der Ausgabe 4/90 wurde auf Seite 14 ein Programm zum Ändern der Bildschirmauflösung vorgestellt. Mit folgendem BASIC-Programm läßt sich der gleiche Effekt zum Nulltarif erzielen:

```
1 OPEN "I", #1, "devs:system-configuration"
2 syscon$=INPUT$(232, #1)
3 CLOSE #1
4 MID$(syscon$,119,1)=CHR$(220) ! Bildschirmbreite
5 MID$(syscon$,120,1)=CHR$(240)
6 MID$(syscon$,217,1)=CHR$(24) ! Bildschirmhöhe
7 MID$(syscon$,218,1)=CHR$(90)
8 OPEN "O", #2, "devs:system-configuration"
9 PRINT #2, syscon$;
10 CLOSE #2
```

Dabei beträgt die horizontale Auflösung 730 Bildpunkte und die vertikale 280. Die Werte für die Bildschirmbreite

und -höhe lassen sich noch etwas verändern. Allerdings ist hier Vorsicht geboten. Zu extreme Werte führen schnell ins "Aus". Änderungen empfehlen sich sowieso nur auf einer Kopie der Workbench-Diskette. Nach dieser Änderung muß das Monitorbild noch auf die neuen Bildschirmmaße eingestellt werden. Bei mir läuft diese Auflösung seit vielen Monaten ohne nennenswerte Probleme. Bei einigen wenigen Programmen kommt es aber zu merkwürdigen "Zerpfückungen" des Mauszeigers.

(Axel Winkler/jb)

Entdeckung des "RAD:es"

Hier habe ich einen Tip zur resetfesten RAM-Disk »RAD:« (Workbench 1.3). Es ist möglich, die resetfeste RAM-Disk mittels des »FastFileSystems« zu beschleunigen, und dies sogar recht einfach. Ergänzen Sie Ihre »MountList« im Verzeichnis »Devs« der Workbench auf folgende Art:

```
RAD: FileSystem = L:FastFileSystem
Device = ramdrive.device
Unit = 0
Flags = 0
Surfaces = 0
BlockPerTrack = 11
Reserved = 2
Interleave = 0
LowCyl = 0
HighCyl = 45
Buffers = 5
BufMemType = 1
GlobVec = -1
DosType = 0x444F5301
Mount = 1
```

Binden Sie die neue RAD: mit dem Befehl »Mount RAD:« in das System ein. Neben einer schnelleren RAD: haben Sie nun auch 512 statt 488 Byte pro Block zur Verfügung.

(Norbert Cohen/jb)

Dieser Schönheitsfehler schreckt sicherlich viele Anwender ab, die Vorteile des Fast-Filing-Systems zu nutzen. Die Idee, ein Programm zu schreiben, welches einen Diskettenwechsel auch unter dem Fast-Filing-System er-

Problemloser Diskettenwechsel im Fast-Filing-System

Nach dem Compilieren mit dem Aztek C 3.6 müssen Sie das Programm in die »Startup-Sequence« einbinden. Die »Startup-Sequence« muß folgende Befehle enthalten:

```
»Autoffs Dfx:=name x:
```

(cd)

Listing: Autooffs.c

Listing: Autooffs.c




```

IOExtTD)))
113: alarm("CreateExtIO Fehler");
114:
115: /* ----- Device pro Gerät öffnen */
116: if(OpenDevice(TD_NAME,unit[i],dreq[i],0))
117: alarm("OpenDevice Fehler");
118:
119: /* ----- Signal - Bit pro Gerät reservieren */
120: if((signum[i] = AllocSignal(-1)) < 0)
121: alarm("AllocSignal Fehler");
122:
123: /* ----- Signal - Bit - Maske erzeugen */
124: signals |= (1 << signum[i]);
125:
126: /* ----- MsgPort pro Gerät ermitteln */
127: if(!pport[i] = (struct MsgPort*)DeviceProc(unam[i]))
128: alarm("DeviceProc Fehler");
129:
130: /* ----- Interrupt - Struktur pro Gerät einrichten */
131: ih[i].is_Data = (APTR)&signum[i];
132: ih[i].is_Code = handlechange;
133: ih[i].is_Node.In_Pri = 0;
134:
135: /* ----- Interrupt pro Gerät aktivieren */
136: dreq[i]->iotd Req.io_Data = (APTR)&ih[i];
137: dreq[i]->iotd Req.io_Command = TD_ADDCHANGEINT;
138: SendIO(dreq[i]);
139:
140:
141: /* - Endlos - Schleife Programm hat keinen Ausgang */
142: for(;;)
143: {
144: /* ----- Warten auf Signale der Interrupt - Routine */
145: sigbit = Wait(signals);
146: for(i=0; i<numio; i++)
147: {
148: /* ----- Signal - Bit pro Gerät prüfen */
149: if(sigbit & (1 << signum[i]))
150: {
151: /* ----- DiskChange durchführen */
152: sendpacket(pport[i],ACTION_INHIBIT,1);
153: sendpacket(pport[i],ACTION_INHIBIT,0);
154: }
155: }
156: }
157: }
158:
159: /*-----*/

```

Listing: Autooffs.c

```

160: * DOS - Packet senden *
161: *-----*/
162: sendpacket(port,type,arg1)
163: struct MsgPort *port;
164: long type,arg1;
165: {
166: pack->sp_Msg.mn.Node.In Name = (char *)&pack->sp_Pkt;
167: pack->sp_Pkt.dp_Link = &pack->sp_Msg;
168: pack->sp_Pkt.dp_Port = dport;
169: pack->sp_Pkt.dp_Type = type;
170: pack->sp_Pkt.dp_Arg1 = arg1;
171:
172: PutMsg(port,&pack->sp_Msg);
173: WaitPort(dport); GetMsg(dport);
174: }
175:
176: /*-----*/
177: * Fehlermeldung ausgeben *
178: *-----*/
179: alarm(text)
180: char *text;
181: {
182: register short s,l;
183:
184: bl.text[TLEN-1] = 0;
185: l = strlen(text);
186: s = (TLEN - 1) / 2;
187: strncpy(&bl.text[s],text,l);
188: DisplayAlert(0,&al,56);
189: exit(1);
190: }
191:
192: #asm
193:
194: public _LVOSignal
195:
196: _handlechange movem.l a0-a1/a6,-(sp) ;Register retten
197: move.l 4,a6 ;ExecBase laden
198: move.w (a1),d1 ;Signal-Bit-Nummer laden
199: moveq #1,d0
200: lsl.l d1,d0 ;Signal-Bit berechnen
201: move.l task,a1 ;zu signalisierender Task
202: jsr _LVOSignal(a6) ;Signal senden
203: movem.l (sp)+,a0-a1/a6 ;Register restaurieren
204: moveq #0,d0 ;weitere Handler sind möglich
205: rts
206:
207: #endasm

```

Listing: Autooffs.c



Wie allgemein bekannt, ist eine Amiga-Diskette in 1760 Blöcke (80 Spuren zu je 22 Sektoren) zu je 512 Bytes organisiert. Nun schreibt das Betriebssystem natürlich nicht die 512 Bytes direkt auf die Diskette, sondern codiert diese entsprechend und schreibt noch einige Bytes Verwaltungsinformationen mit dazu. Das genaue Verfahren ist bereits hundertmal durchgekauft worden und soll hier nicht nochmals breitgetreten werden, wichtig für uns ist nur, daß zu jedem 512-Byte-Sektor noch ein sogenanntes »Sector Label« gehört, das 16 Bytes lang ist. Ursprünglich waren diese »Sector Labels« dazu gedacht, Informationen für die Dateistruktur aufzunehmen (Block-Verbindungszeiger). Da diese ab Kickstart 1.1 aber vollständig in den 512 Datenbytes untergebracht wurden, werden die »Sector Labels« nicht mehr benutzt. Sie stehen also für unsere Zwecke zur Verfügung. 1760 Labels zu 16 Bytes ergeben demzufolge die oben bereits erwähnten 28160 Bytes. Im folgenden werden Grundkenntnisse des Diskettenzugriffs mittels »trackdisk.device« (Gerätetreiber für Diskettenlaufwerke) vorausge-

Oliver Wagner

Sector Label Freier Speicher gut genutzt

Wußten Sie schon, daß auf jeder AmigaDOS-Diskette gut 28 kByte Speicher unbenutzt herumliegen? Warum diesen Platz brachliegen lassen?

setzt. Zusätzlich zu den »normalen« IO-Befehlen wie »CMD_READ«, »CMD_WRITE« und anderen, kennt das »trackdisk.device« noch einen zweiten Satz Kommandos, die sogenannten »ETD-Com-mands«. Diese werden zur Unterscheidung mit »ETD_...« eingeleitet. Der Unterschied zu den normalen Kommandos besteht einerseits darin, daß ein Zähler über die Diskettenwechsel überprüft wird (damit nicht zwischen zwei Schreibzugriffen die Diskette gewechselt wird). Zum anderen werden auf Wunsch die »Sector Labels« automatisch mitgelesen beziehungsweise mitgeschrieben. Die ETD-Befehle benötigen dazu eine erweiterte IO-Struktur:

Zusätzlich zum »normalen« IO-Request enthält der Request »ExtTD« noch einen Zähler für den Diskwechsel

sowie einen Zeiger auf das Sector-Label-Feld. Der Diskwechsel-Zähler muß vor jedem Schreib- oder Lesezugriff mit dem aktuellen Wert initialisiert werden. Diesen Wert erhält man mit dem Kommando »TD_CHANGEUM« in »iotd.Reg.io_Actual«. Ist das Feld »iotd_SecLabel« nicht null, so muß es auf einen genügend großen Speicherplatz (16 Bytes mal Anzahl Sektoren) im »ChipMem« zeigen, wo die »Sector Labels« abgelegt beziehungsweise gelesen werden. Damit stellt der Zugriff auf den zusätzlichen Diskspeicherplatz kein Problem mehr dar.

Eine sinnvolle Anwendung der »Sector Labels« besteht beispielsweise darin, auf Disketten eine versteckte Notiz anzubringen, die ohne Hilfsmittel nicht zu entdecken ist. Die »Sector Labels« werden

nämlich, solange man dies nicht ausdrücklich befiehlt, von keinem Trackdisk-Kommando angetastet (außer »TD_FORMAT«). Auch kopieren die gängigen Kopierprogramme die Labels einfach mit »Pcopy«, »Turbobackup« oder »XCopy«.

Das kleine C-Programm »Disknote« legt einen bis zu 16 Zeichen langen Text im »Sector Label« des Blocks 880 ab beziehungsweise zeigt einen eventuellen Text an. Man kann so zum Beispiel PD-Disketten, die man in Umlauf bringt, markieren und später feststellen, wo sie herkommen. Das Programm soll natürlich nur ein Beispiel sein. Spinnt man den Faden weiter, könnte man ganze Textdateien oder kurze Programme auf der Diskette verstecken, solange diese nicht länger als 28 kByte sind.

Die Benutzung von »Disknote« ist simpel. Die Syntax lautet »disknote DFx:«. »DFx:« steht dabei für DF0: / DF1: / DF2: / DF3: und kann den Block 880 vom entsprechenden Laufwerk lesen und zeigt dementsprechend das »Sector Label« an. »disknote DFx: blabla« schreibt zusätzlich noch den Text »blabla« in das »Sector Label«. Bei beiden Varianten werden Fehler beim Schreiben ange-

zeigt. Dies soll aber weiter nicht stören. »Disknote« ist mit dem Lattice-C-Compiler in der Version 5.04 geschrieben und

deshalb mit weniger als 2 kByte relativ kurz. Es läßt sich natürlich resident machen (unter ARP mit »ARES diskno-

te NOCHECK«). Das Nachvollziehen des Programms sollte dank der ausführlichen Dokumentation kein Problem dar-

stellen, bei Fragen wenden Sie sich bitte über die Redaktion an den Autor (Rückporto nicht vergessen). (vb)

Listings

```
1: /******  
2: /* DISKNOTE.C  
3: /* schreibt einen max. 16 Zeichen langen Text in  
4: /* das Sector-Label des Blocks 880 bzw. liest  
5: /* diesen aus  
6: /* by Oliver Wagner  
7: /* (c) 1990 AMIGA DOS  
8: /* Sprache: Lattice C V5.04  
9: /******  
10:  
11: /* Aufrufe:  
12:  
13: lc -rr -v -ms -rr -csmq -O disknote  
14: blink lib:c.o,disknote.o to disknote ND SC SD CHIP LIB  
15: lib:lcr.lib DEFINE @_main=@_tinymain  
16:  
17: Das Programm ist resident-fähig!  
18:  
19: Benutzung:  
20: Label-lesen : disknote DFx:  
21: Label-schreiben : disknote DFx: text(max.16 Zeichen)  
22:  
23: */  
24:  
25: /* Notwendige Includes (Strukturen, Protos usw.) */  
26:  
27: #include <proto/exec.h>  
28: #include <proto/dos.h>  
29: #include <devices/trackdisk.h>  
30: #include <stdlib.h>  
31: #include <string.h>  
32:  
33:  
34: /* Trick für kürzeren Code */  
35:  
36: void MemCleanup() { }  
37:  
38:  
39: /* Globale Variablen */  
40:  
41: char blockbuff[512];  
42: /* Puffer für den Block (muß im CHIP liegen) */  
43: char labelbuff[18];  
44: /* Puffer für das Label (mit Sicherheitsabstand  
45: für strcpy() (siehe unten) */  
46: struct MsgPort *port; /* IO-Port */  
47: BPTR output; /* Output-Filehandle (für Textausgabe) */  
48: struct IOExtTD ior; /* IO-Request */  
49: int drive; /* Gerätenummer */  
50: int open; /* schon geöffnet? */  
51:  
52: /* kurze Ausgaberoutine */  
53:  
54: #define print(s) Write(output,s,strlen(s))  
55:  
56: /* lokale Funktions-Prototypen */  
57:  
58: void main(int,char**);  
59: void cleanup(char*);  
60: void tdopen(void);  
61: void readblock(void);  
62: void writeblock(void);  
63:  
64: /* Funktionen */  
65:  
66: /* Aufräumen; evt. Fehlermeldung ausgeben */  
67:  
68:  
69: void cleanup(s)  
70: char *s;  
71: {  
72: if(s) {  
73: print("\33[33m");  
74: print(s);  
75: print("\33[0m\n");  
76: }  
77: if(open) {  
78: ior.iotd Req.io Command=TD_MOTOR;  
79: ior.iotd Req.io Length=0;  
80: DoIO(&ior);  
81: }  
82: if(port) DeletePort(port);  
83: XEXIT((s)?20:0);  
84: }  
85:  
86: /* Trackdisk-Device für Gerät "drive" öffnen */  
87:  
88: void tdopen()  
89: {  
90: open=0;  
91: if(!port>CreatePort(0,0)) cleanup("kann IO-MsgPort  
nicht erzeugen");  
92: ior.iotd Req.io Message.mn ReplyPort=port;  
93: if(OpenDevice("Trackdisk.device",drive,&ior,0))  
94: cleanup("kann 'trackdisk.device' für das angegebene  
Laufwerk nicht öffnen");
```



```
95: }  
96:  
97: /* Block 880 lesen */  
98:  
99: void readblock()  
100: {  
101: /* Flag für cleanup() */  
102: open=1;  
103:  
104: /* Changenum holen */  
105: ior.iotd Req.io Command=TD_CHANGENUM;  
106: DoIO(&ior);  
107: ior.iotd Count=ior.iotd Req.io Actual;  
108:  
109: /* Block lesen */  
110: ior.iotd Req.io Command=ETD_READ;  
111: ior.iotd Req.io Length=512;  
112: ior.iotd Req.io Offset=880*512;  
113: ior.iotd Req.io Data=(APTR)blockbuff;  
114: ior.iotd SecLabel=(ULONG)labelbuff;  
115: DoIO(&ior);  
116:  
117: /* Fehler ? */  
118: if(ior.iotd Req.io Error) cleanup("Block 880  
Lesefehler!");  
119: }  
120:  
121: /* Block 880 schreiben */  
122:  
123: void writeblock()  
124: {  
125: /* Block schreiben */  
126: ior.iotd Req.io Command=ETD_WRITE;  
127: DoIO(&ior);  
128:  
129: /* Fehler? */  
130: if(ior.iotd Req.io Error) cleanup("Block 880  
Schreibfehler!");  
131:  
132: /* Update wichtig, sonst wird der Block nicht auf  
133: Disk geschrieben */  
134: ior.iotd Req.io Command=ETD_UPDATE;  
135: DoIO(&ior);  
136:  
137: /* Nochmals: Fehler? */  
138: if(ior.iotd Req.io Error) cleanup("Block 880  
Schreibfehler!");  
139: }  
140:  
141:  
142: /* Hauptfunktion: Argumente auswerten usw. */  
143:  
144: void main(argc,argv)  
145: int argc;  
146: char **argv;  
147: {  
148: output=Output();  
149: print("\33[1;4mDisknote\33[0m von Oliver Wagner,  
- 1990 AmigaDOS\n");  
150:  
151: /* Benutzung */  
152: if(argc<2||*argv[1]!='?')  
153: cleanup("Benutzung ist: DISKNOTE DFx: [labeltext]");  
154:  
155: /* Welches Laufwerk? */  
156: if(!strcmp(argv[1],"df0:")) drive=0;  
157: else if(!strcmp(argv[1],"df1:")) drive=1;  
158: else if(!strcmp(argv[1],"df2:")) drive=2;  
159: else if(!strcmp(argv[1],"df3:")) drive=3;  
160: else cleanup("falsche Laufwerksangabe");  
161:  
162: /* "trackdisk.device" öffnen */  
163: tdopen();  
164:  
165: /* Block 880 lesen */  
166: readblock();  
167:  
168: /* Ausgeben des alten Labels */  
169: print("Das aktuelle Label lautet: ");  
170: Write(output,labelbuff,16);  
171: print("\n");  
172:  
173: /* Label aendern? */  
174: if(argc==3) {  
175: if(strlen(argv[2])>16)  
176: cleanup("Das Label darf hoechstens 16 Zeichen  
lang sein");  
177: memset(labelbuff,0,16);  
178: strcpy(labelbuff,argv[2]);  
179: writeblock();  
180: }  
181:  
182: /* Fertig */  
183: print("Ok.\n");  
184: cleanup(0);  
185: }  
186:  
187: /****** Ende von Disknote.c *****/
```

Listing: Disknote.c

Listing: Disknote.c

Eine Speichererweiterung im Eigenbau macht es möglich, auf der PC-Karte auch "speicherplatzfressende" MS-DOS-Programme starten zu können. Es handelt sich hier um eine Speichererweiterung, die die PC-Karte im A2000 oder das Side-Car des A1000 preiswert um 128 kByte auf 640 kByte Ram aufrüstet.

Die Platine hat folgende Vorteile:

- Der Zusatzspeicher wird automatisch vom BIOS erkannt.
- Kein Einstellen von Dip-Switches notwendig.
- Kurze PC-Karte (paßt in den freien halben Slot einer File-Card).
- Geringe Stromaufnahme durch Verwendung von statischen CMOS-RAMs.
- Einfacher Nachbau bei Verwendung der angebotenen fertigen Platine.
- geringe Kosten (kompletter Bausatz: 135,- DM).

Die Amiga-PCs (sowohl PC-Karte als auch Side-Car) verfügen von Haus aus nur über

Peter Engels

Preiswerte Aufrüstung der XT-Karte

Selbstbau einer 128-kByte-Speichererweiterung

"Herr, laß RAM regnen..."; dieses mit einem Stoßseufzer versetzte Bittgebet ist nicht nur den reinen Amiga-Besitzern bekannt, auch die Benutzer der PC-Karte im Amiga 2000 könnten so manch zusätzliches Plätzchen im Speicher gut gebrauchen. Wir von der AMIGA DOS stellen da mal eine Frage: Warum nicht selbst Abhilfe schaffen?

maximal 512 kByte RAM. Um aber neuere Software wie MS Word5, dBASE IV oder den Ventura-Publisher fahren zu können, benötigt man den maximalen DOS-Speicher von 640 kByte.

Bei der Aufrüstung entstehen aber vielerorts Probleme bezüglich der Kompatibilität. Tatsächlich ist es so, daß die

meisten käuflich zu erwerben RAM-Erweiterungen für die PC-Seite nicht im Amiga-PC laufen, da diese als höchste einzustellende Startadresse \$70000(hex) vorgeben. Der Amiga-PC verlangt jedoch die Startadresse \$80000(hex) für RAM-Erweiterungen.

Auf Grund dieser Tatsache wurde eine Schaltung ent-

wickelt, die folgende Voraussetzungen erfüllen sollte:

1. leichte Nachbaumöglichkeit,
2. geringe Kosten,
3. einfache Installation,
4. 100%ige Kompatibilität.

Alle Forderungen wurden durch die nachfolgend beschriebene RAM-Erweiterung erfüllt.

Die Schaltung der RAM-Erweiterung

Die komplette Schaltung der Erweiterung (Bild 1) konnte auf Grund der verwendeten RAMs (statisch 32K * 8 Bit) einfach gehalten werden. IC6 und IC7 übernehmen die Pufferung der Adreßsignale A0 bis A14 und der MEMR-Leitung des PC-Bus, um diesen zu entlasten. IC8, ein bidirektionaler Datenpuffer, übernimmt die Pufferung der Datenleitungen D0 bis D7. Da die Daten in zwei Richtungen laufen (Lesen und Schreiben), wird die Richtung des Datenpuffers über die gepufferte MEMR-Leitung vorgenommen. Will die CPU Daten ins

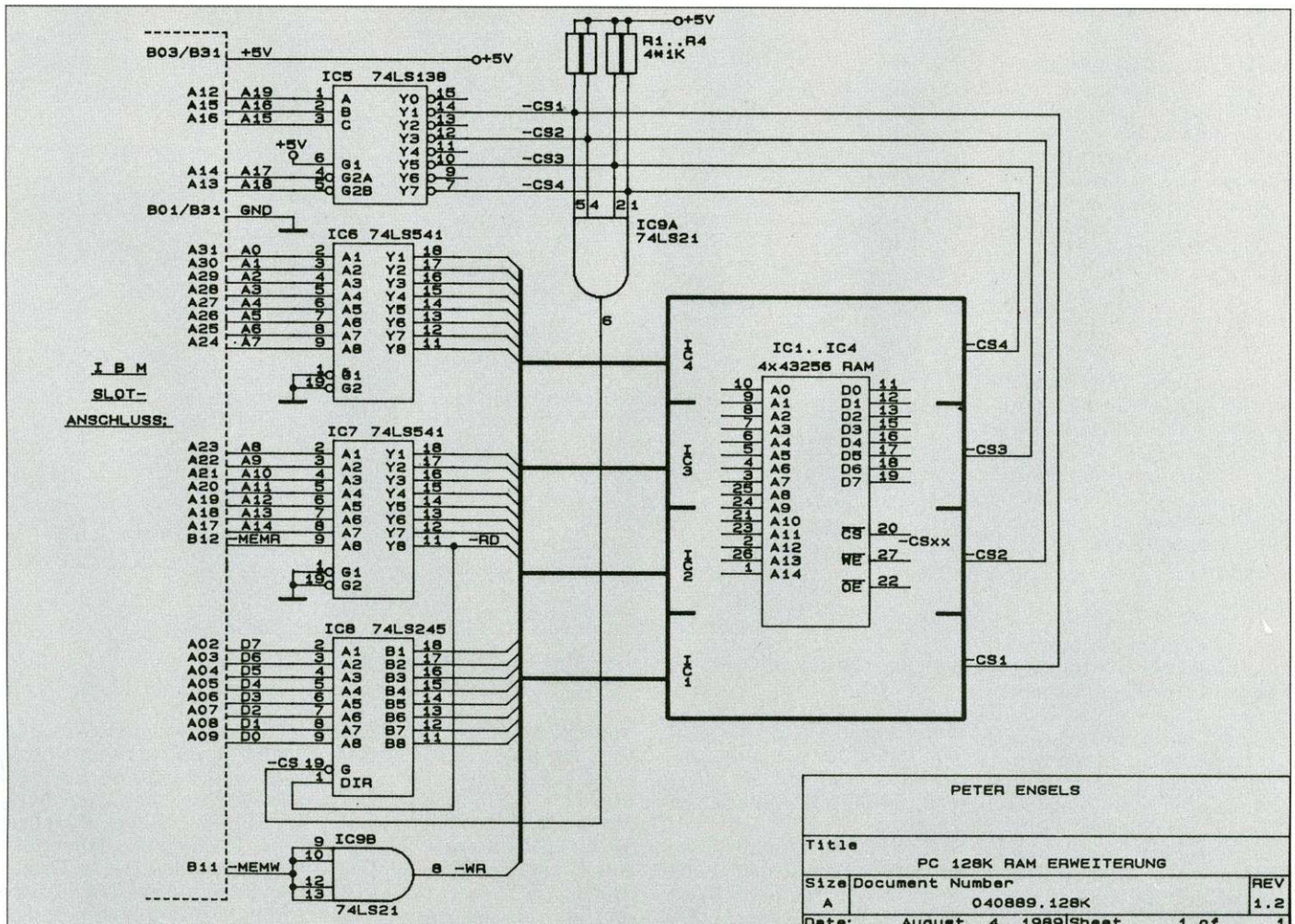


Abb. 1. Der Schaltplan der Erweiterungskarte

RAM schreiben, wird die Richtung des Datenpuffers vom Slot zu den RAMs geschaltet. Sollen Daten aus dem RAM gelesen werden, dann wird die Richtung des IC8 über den Pegel an Pin 1 in die andere Richtung umgeschaltet. Das heißt, die Daten-signale können nun von den RAMs zum Slot fließen.

Da alle Pins der vier RAM-Bausteine bis auf die Chip-Select Leitung (Pin 20) parallel liegen, muß nun dafür gesorgt werden, daß immer nur der entsprechende Baustein auf den Prozessorbus geschaltet wird. Dafür sorgt IC5, das einen 3-Bit-Binärcodecorder enthält: Je nach Zustand der Eingangspins A, B, C wird einer der Ausgänge Y0 bis Y7 auf LO gelegt.

Durch die Beschaltung von IC5 werden die Leitungen -CS1 bis -CS4 in folgenden Adreßbereichen LO:

-CS1 zwischen \$80000 und \$87FFF

-CS2 zwischen \$88000 und \$8FFFF

-CS3 zwischen \$90000 und \$97FFF

-CS4 zwischen \$98000 und \$9FFFF

Bei allen anderen Speicherzugriffen, die nicht in den oben aufgeführten Bereichen liegen, sind -CS1 bis -CS4 auf HI-Pegel und sperren so den entsprechenden RAM-Baustein. Somit kann es nie zu Konflikten auf dem Daten- oder Adreßbus kommen.

Ein Beispiel zum Betrieb der Karte

Zusätzlich zur RAM-Freigabe wird über IC9A der Datenpuffer freigegeben. Wird eine der Select-Leitungen -CS1 bis -CS4 LO, dann wird auch der Ausgang des UND-Gatters IC9A LOW und gibt die Datenleitungen der RAMs über IC8

Wichtig!

Aus Platzgründen mußten wir die Platinenlayouts verkleinert darstellen, sie sind also nicht maßstabsgerecht. Wer die Erweiterung komplett selbst bauen will, sollte einen an sich selbst adressierten und ausreichend frankierten DIN-A4-Umschlag an die Redaktion der AMIGA DOS schicken, in dem wir 2:1-Vorlagen zurücksenden.

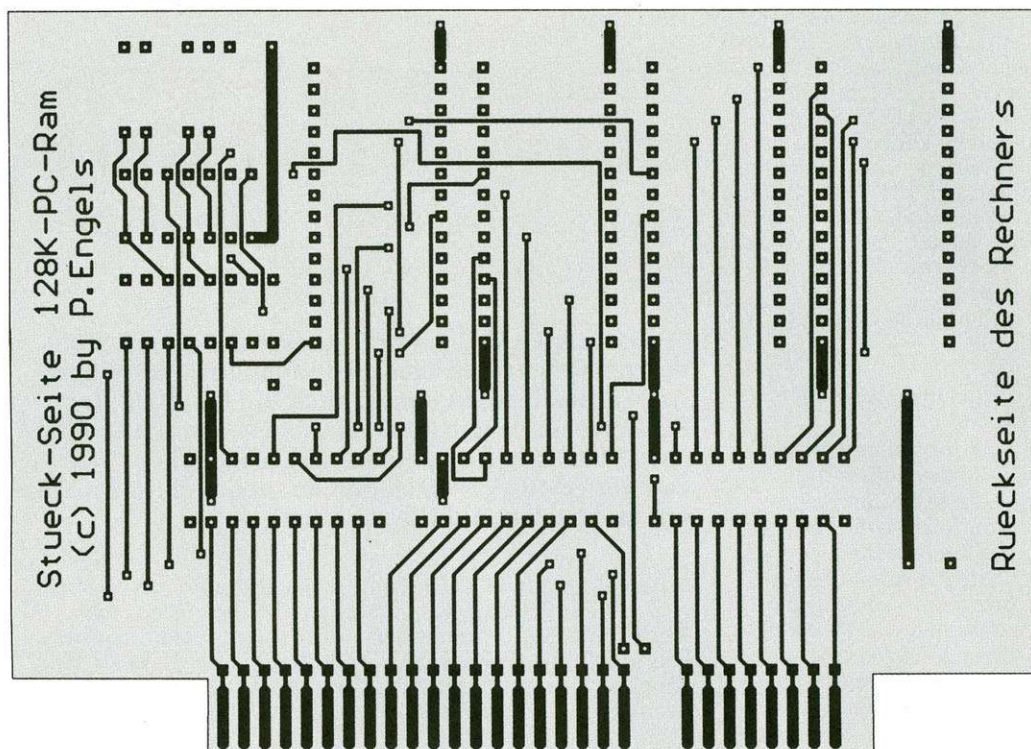


Abb. 2. Die Platine der RAM-Erweiterung. Hier sehen Sie die Bestückungsseite...

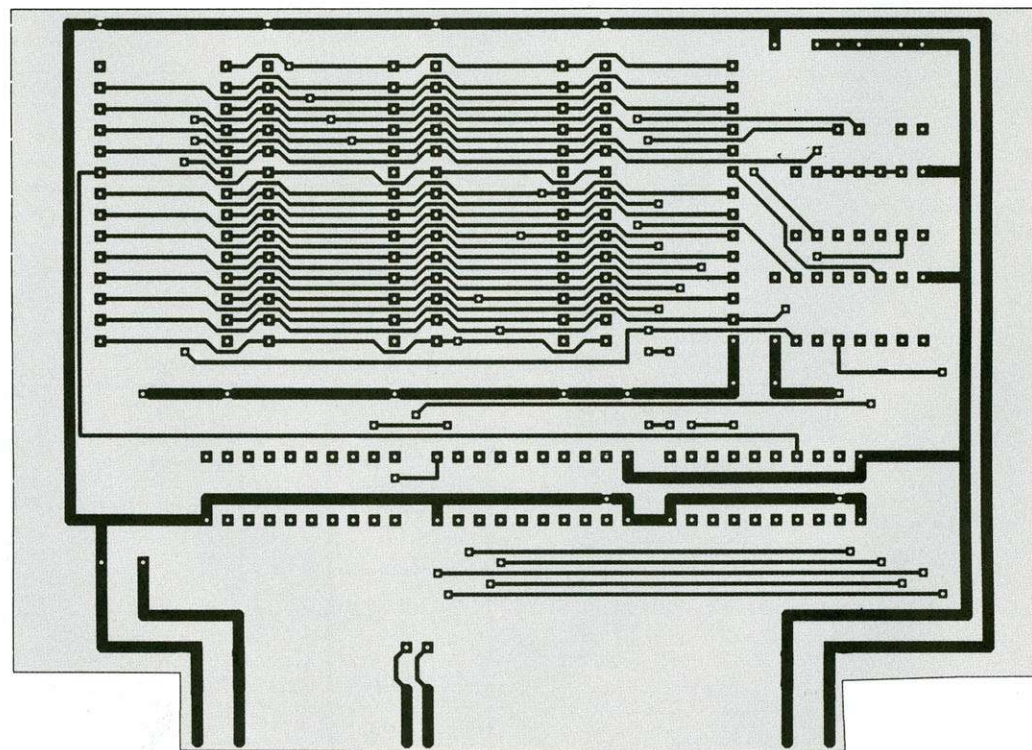


Abb. 3. ...und hier die Lötseite, beide Seiten sind nicht maßstabsgerecht

frei. Somit wird zusätzlich der Datenbus nur beim Ansprechen einer Adresse im Bereich \$80000 bis \$9FFFF freigegeben.

Stellen wir uns einmal vor, der Prozessor will augenblicklich Daten aus der RAM-Adresse \$8C700 lesen. Der Prozessor legt die Adresse auf seinen Bus und aktiviert die

Leitung -MEMR (Lesen), IC8 wird durch den Pegel der -RD Leitung in die Richtung "RAM -> Prozessor" geschaltet. Die Adreßleitungen A0 bis A14 liegen über IC6 und IC7 an den RAMs an. Der Decoder IC5 legt abhängig von den Adreßleitungen A15 bis A19 den Ausgang -CS2 auf LOW-Pegel. Dieses Signal se-

lektiert nun den RAM-Baustein IC2, welcher sein Datenbyte, das durch A0 bis A14 adressiert ist, auf seine Ausgänge D0 bis D7 legt. Der über IC9A freigegebene Datenpuffer IC8 schaltet nun die Daten vom RAM IC2 auf den Prozessorbus. Der Prozessor liest danach das anstehende Datenbyte ein. Der Lesevorgang ist

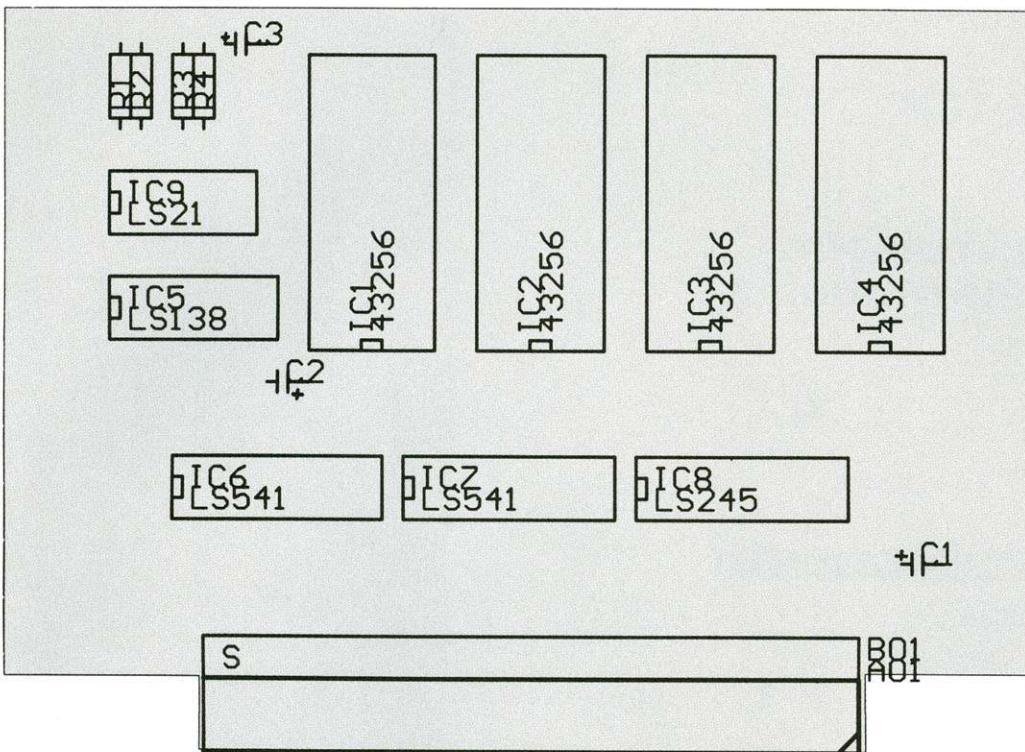


Abb. 4. Die Bestückung mit den Bauteilen sehen Sie hier

abgeschlossen. Der Aufbau der Platine gestaltet sich sehr einfach: Bild 2 und Bild 3 geben das Layout der doppelseitigen Platine wieder, die von versierten Bastlern selbst geätzt werden kann. Es ist dabei daran zu denken, daß die Durchkontaktierungen selbst gemacht werden müssen. Wer sich das Ätzen ersparen will, der kann die fertige Leerplatine in Industriequalität (durchkontaktiert, mit Lötstopmaske) direkt beim Autor zum Preis von 32,50 DM per Nachnahme bestellen, ebenfalls ist ein kompletter Bausatz lieferbar (Preis: zirka 135,- DM). Die Bestellanschrift finden Sie in der AMIGA-DOS-Toolbox im Artikel.

Hinweise für den Eigenbau

Bei der Bestückung sollte man darauf achten, daß man mit einem LötKolben nicht über 30 Watt arbeitet, ebenfalls sollte man eine feine Lötspitze verwenden, da die Löttaugen ziemlich eng aneinander liegen.

Zuerst bestückt man die passiven Bauelemente (R1 und R4 und die Tantalelko), dann übernimmt man die Bestückung mit den ICs, wobei man zumindest für die RAM-Bausteine Sockel verwenden

sollte. Bitte achten Sie beim Bestücken auf die Lage der IC-Kerben!

Ist der Aufbau beendet, überprüft man die Platine nochmals auf etwa vorhandene Kurzschlüsse beziehungsweise nicht verlötete Pins.

Noch ein paar Anmerkungen zur Inbetriebnahme

Zum ersten Funktionstest steckt man die Karte zuerst

ohne RAM-Bausteine in einen freien PC-Slot.

Achtung! Beachten Sie bitte, daß der Text "Rückseite des Rechners" auf der Platine auch zur Rückseite des Rechners hinzeigt!

Nach dem Einschalten des Rechners muß dieser nun einwandfrei booten, allerdings noch mit 512 kByte RAM. Ist dies nicht der Fall, dann überprüfen Sie nochmals Ihre Bestückung. War dieser erste Funktionstest positiv, das heißt, läßt sich der PC einwandfrei hochfahren, dann

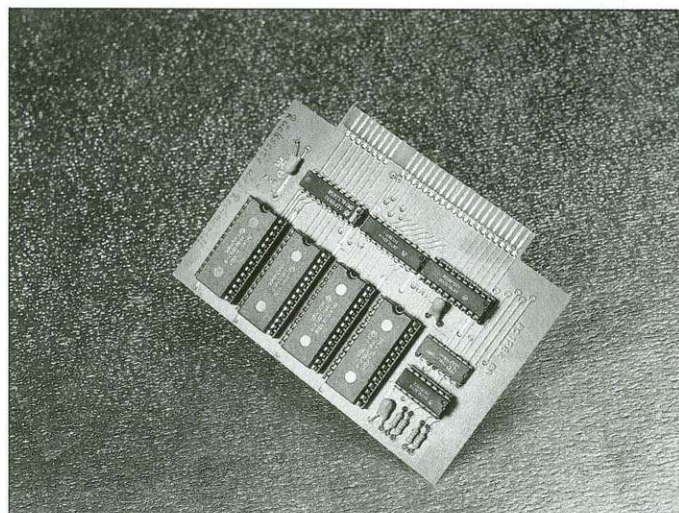


Bild 1. Fertig aufgebaut sieht die Schaltung dann so aus

bestücken Sie die Platine nun mit den vier RAM-Bausteinen (Lage der Kerben beachten), und booten Sie den Rechner nochmals.

Nun sollte der PC beim Speichertest bis 640 kByte hochlaufen; der Speicher wird somit erkannt.

Falls Sie das Hochzählen des PC nicht verfolgen können, ist eine andere Überprüfung des Speichers möglich: Geben Sie den DOS-Befehl »CHKDSK« ein. Die letzten beiden Zeilen des jetzt ablaufenden Protokolls geben Ihre Speichergrösse an. In der vorletzten Zeile sollte der Satz stehen "655360 Byte Gesamtspeicher".

Ist dies der Fall, dann haben Sie Ihren PC erfolgreich auf 640 kByte aufgerüstet, es sind nun alle Wege für große Applikationen geöffnet.

Besonderheiten beim Side-Car

Beim Side-Car muß man im Gegensatz zur PC-Karte noch die Dip-Switch-Stellung auf der Hauptplatine beachten: Wollen Sie auf 640 kByte Gesamtspeicher aufrüsten, dann müssen Schalter "3" und "4" von "SW1" auf OFF stehen (vergleichen Sie dazu die Seite 16 des User-Manuals). Bei der PC-Karte ist dagegen keine Einstellung nötig.

(jb)

AMIGA DOS Toolbox

Die Bauteile-Liste

IC1..IC4	= 43256/12 (Stat. DRAM 32K * 8)
IC5	= 74 LS 138 (3 zu 8 Binärdecoder)
IC6..IC7	= 74 LS 541 (8-Bit Bustreiber)
IC8	= 74 LS 245 (8-Bit Datentreiber)
IC9	= 74 LS 21 (2-fach AND mit 4 Eingängen)
R1..R4	= 1 kΩ (Widerstand 1K/0,25 Watt)
C1..C3	= 10µF (Tantalelko 10µF/10 Volt)

Leerplatine PC 128K

Die Leerplatine sowie alle Bauteile beziehungsweise der komplette Bausatz können bei folgender Adresse per Nachnahme bestellt werden:

Peter Engels
Postfach 1225
5358 Bad Münstereifel
Tel: 02253/2667

Andreas Polk

Freie Fahrt für Modula2

Eine Einführung in die Programmiersprache Modula2

Kursfahrplan

Teil 1: Der Compiler M2Amiga

Teil 2: Die ersten Befehle

Teil 3: Schleifen und Verschachteln

Teil 4: Prozeduren und Variablen

Teil 5: Weitere Datentypen

Teil 6: Zeiger und Datenstrukturen

Teil 7: Modulkonzept

Teil 8: Systemprogrammierung

Bisher liefen immer alle Programme vom ersten bis zum letzten Befehl linear ab. Es war nicht möglich, die Folge der Befehlsabarbeitung zu beeinflussen. Dies wird erst mit den folgenden Befehlen möglich, die in diesem Teil erklärt werden.

Nehmen wir ein einfaches Beispiel. Möchten Sie einen Text zehnmal hintereinander ausgeben, mußten Sie bisher ein Programm schreiben, das zehn gleiche Befehle beinhaltet. Dies ist nicht nur sehr umständlich, sondern auch uneffektiv, da dadurch das Programm sehr lang und langsamer wird.

Um dem abzuhelfen, gibt es die Schleife »FOR...TO...DO«. Wenn Sie schon in einer anderen Sprache programmiert haben, werden Sie diesen Befehl sicherlich kennen. Er ist nämlich in fast jeder Sprache implementiert. Seine Syntax sieht folgendermaßen aus:

```
FOR Laufvar:=Anfang TO Ende BY
Schrittweite DO Anweisungs-
folge;
END;
```

Schauen wir uns diesen mächtigen Befehl einmal genauer an. Nach dem Schlüsselwort »FOR« folgt eine Zuweisung an eine Laufvariable. Diese Variable heißt so, weil

sie innerhalb der Schleife verschiedene Werte annimmt, sie durchläuft praktisch alle Werte. Dieser Laufvariablen wird ein Anfangswert zugewiesen. Nach dem nächsten Schlüsselwort »TO« folgt dann der Endwert. Die Schleife wird so lange durchlaufen, bis die Laufvariable den Wert »Ende« angenommen beziehungsweise überschritten hat. Natürlich muß sich der Wert der Laufvariablen noch ändern, sonst würde Sie nie den Endwert annehmen, und wir hätten so eine Endlosschleife, also eine Schleife, die nie beendet wird. Das Abändern der Laufvariablen wird von dem Befehl selbst übernommen. So wird normalerweise mit jedem Schleifendurchgang der Wert der Laufvariablen um eins erhöht. Soll sich die Schrittweite, also der Wert, um den die Laufvariable erhöht wird, ändern, muß nach dem optionalen Schlüsselwort »BY« die Schrittweite angegeben werden. Nach dem Schlüsselwort »DO« folgt die

Anweisungsfolge, die innerhalb der Schleife ausgeführt wird. Nach dieser Anweisungsfolge muß das Schlüsselwort »END« folgen, damit der Compiler weiß, wo die Anweisungsfolge zu Ende ist. Nach soviel grauer Theorie ein paar Beispiele, dann wird die Materie leichter verständlich. Nehmen wir als erstes einfaches Beispiel die unsinnige Ausgabe des Wortes »Hallo«.

Dieses Wort soll zehnmal hintereinander ausgegeben werden. Schauen wir uns dazu das Modul »Ausgabe« an. In der ersten Zeile finden wir das Schlüsselwort »MODULE«. Dann folgen die Importe einiger Befehle, und anschließend wird die Variable »i« deklariert.

Hierbei handelt es sich um die Laufvariable für die Schleife. Wenn man den Wert der Laufvariablen nicht braucht, hat sich für deren Namen »i« eingebürgert. Das muß man natürlich nicht so machen, kann man aber. Nun kommt der

Programmcode, der wie üblich durch »BEGIN« eingeleitet wird. Die Schleife »FOR« folgt. Zuerst finden wir hier wieder das Schlüsselwort »FOR«. Dann wird der Variablen »i« der Wert eins zugewiesen. Da die optionale Anweisung »BY« Schrittweite weggelassen wurde, wird »i« bei jedem Schleifendurchlauf um eins erhöht. Zuerst hat »i« also den Wert 1. Dann wird die Anweisungsfolge zwischen dem »FOR« und dem »END« ausgeführt, in diesem Falle wird hier also das Wort »Hallo« ausgegeben und ein anschließender Rücklauf ausgeführt. Nun kommt der Compiler zu dem Schlüsselwort »END«, was signalisiert, daß hier der Anweisungsteil, der innerhalb der Schleife ausgeführt werden soll, zu Ende ist. Die Variable »i« wird um eins erhöht, erhält also den Wert 2, und das Programm kehrt zum Befehl »FOR« zurück. Dort wird überprüft, ob »i« (hat den Wert 2) kleiner als zehn ist, was zutrifft. (Der Varia-



blen »i« wird nun natürlich nicht mehr der Wert 1 zugewiesen. Diese Initialisierung findet nur beim ersten Mal statt.) Somit wird der Anweisungsteil ein weiteres Mal durchlaufen.

Überspringen wir nun einen Teil und stellen uns vor, daß »i« nach einigen Durchläufen den Wert 9 angenommen hat. Der Text wurde bisher neunmal ausgegeben. »i« wird um eins erhöht und hat den Wert 10. Die Schleife wird noch einmal durchlaufen, da »i« noch nicht größer als der Endwert 10 ist. Ein weiteres Mal wird "Hallo" ausgegeben. Anschließend wird »i« wieder um eins erhöht, hat somit den Wert 11. Da 11 größer ist als der Endwert 10, wird die Schleife nicht noch ein weiteres Mal ausgeführt. Achten Sie immer darauf, daß der Wert der Laufvariablen eins höher ist als der angegebene Endwert in der Abbruchbedingung. So hat »i« nach dem Durchlaufen der Schleife den Wert 11 und nicht 10! Das ist dann zu bedenken, wenn Sie den Wert noch weiterhin brauchen. Ich habe in diesem Beispiel einfach der Laufvariablen den Typ »Integer« zugeordnet.

Dies ist nur eine Möglichkeit von vielen. Erlaubt sind folgende Typen:

Cardinal
Integer
Boolean
Char
Bitset

Natürlich sind auch deren lange Versionen zugelassen. Den Typ »Bitset« kennen Sie übrigens noch nicht. Die Erklärungen dazu kommen im fünften Kursteil.

Nun ein weiteres (sinnvolles) Beispiel. Ich möchte Ihnen an dieser Stelle ein Programm vorstellen, das die Fakultät einer Zahl berechnet. Wie Sie wissen, ist die Fakultät einer Zahl n definiert als das Produkt $1 \cdot 2 \cdot 3 \cdot \dots \cdot (n-1) \cdot n$. Somit ist 3 Fakultät (man schreibt kurz: 3!) $1 \cdot 2 \cdot 3 = 6$. Für den Sonderfall 0 gilt: $0! = 1$.

Schauen Sie sich dazu das Modul »Fakultaet« einmal an. Interessant ist natürlich wieder die For-Schleife. Wie beim ersten Durchgang wird die Variable »Laufvar« mit dem Wert eins initialisiert. Dann wird innerhalb der Schleife der neue Wert für Ergebnis errechnet. Anschließend wird »Laufvar« um eins erhöht und dann überprüft, ob deren Wert

schon größer der eingegebenen Zahl ist. Ist dies nicht der Fall, wird wieder das bisherige Ergebnis mit dem Wert der Laufvariablen multipliziert. Diese Schleife wird so lange durchlaufen, bis die Laufvariable größer als die eingegebene Zahl ist.

Die unendliche Schleife

Anschließend wird das Ergebnis ausgegeben. Beachten Sie, daß der Fall für Zahl=0 ebenfalls zum richtigen Ergebnis führt. Da Zahl in diesem Falle gleich 0 ist und direkt bei der ersten Überprüfung am Anfang der For-Schleife die Laufvariable größer als Zahl ist, wird die Schleife gar nicht durchlaufen. Somit beträgt das Ergebnis dann eins, da Ergebnis am Anfang so initialisiert wurde.

Ersetzt man die Zeile »FOR Laufvar:=1 TO Zahl DO« durch »FOR Laufvar:=1 TO Zahl BY 1 DO«, bewirkt das nichts. Dies wäre nur eine Ergänzung, die noch einmal explizit angibt, um welchen Wert »Laufvar« jeweils erhöht werden soll. Da als Schrittweite immer eins angenommen wird, wenn die Anweisung »BY« weggelassen wird, kann hier der Zusatz »BY« auch weggelassen werden. Wie kann man aber nun eine Variable nicht rauf-, sondern runterzählen? Eine Möglichkeit, die sich anbietet, wäre eine negative Zahl als Schrittweite anzugeben, also beispielsweise »FOR Laufvar:=10 TO 1 BY -2 DO«. Dies klingt ganz logisch, ist an sich auch richtig, nur wird dann das Wort »TO« durch »DOWNTO« ersetzt. Auf diese Weise wird der Programmcode besser lesbar. Sehen Sie sich dazu einmal das Modul »Runterzaehlen« als Beispiel an. Es zählt eine Variable runter, und bei jedem Schleifendurchgang wird ihr aktueller Wert angezeigt. Beachten Sie, daß die Schleife bereits nach sieben und nicht erst nach acht Durchgängen beendet wird, da -11 kleiner als -10 ist. Anschließend wird allerdings der Endwert noch einmal ausgegeben; Sie sehen wieder, der Wert der Laufvariablen beim Verlassen der Schleife muß nicht unbedingt den Endwert annehmen. Natürlich können Sie die Laufvariable auch innerhalb der Schleife abändern. Dabei müssen Sie aber vorsichtig sein, wenn Sie nicht aufpas-

sen, kann es vorkommen, daß aus der Schleife eine Endlosschleife wird, da der Abbruchzustand nicht erreicht wird. Das Programm kann dann nicht mehr normal beendet werden. Ein Beispiel dafür finden Sie im Modul »Endlos«.

Sie können das Programm aber noch mit der Notbremse verlassen, indem Sie [Ctrl-C] drücken. Programme, bei denen Ausgaben gemacht werden oder Eingaben verlangt werden, werden dann nämlich abgebrochen, und es erscheint ein Requester, der Sie auf den Programmabbruch aufmerksam macht und nur noch weggeklickt werden muß. (Eventuell müssen Sie öfter [Ctrl-C] drücken, um den Augenblick zu erwischen, wo der String "und nochmal..." ausgegeben wird.) Sie wissen nun, wie Sie eine Endlosschleife programmieren können. Falls Sie das einmal tun müssen, so können Sie es mit der Schleife »FOR...TO...DO« machen. Dies ist allerdings nicht sehr elegant. Aus diesem Grunde wurde in Modula2 eine weitere Schleifenart implementiert, und zwar die Schleife »LOOP«. Während man bei der Endlosschleife »FOR...TO...DO« keine Möglichkeit mehr hat, das

Programm sauber zu verlassen oder gar die Schleife abbrechen und das Programm weiter ausführen zu lassen, ist dies mit der Schleife »LOOP« möglich. Hier zunächst ihre Syntax: »LOOP Anweisungen END;«. Sie sehen, der Befehl »LOOP« ist sehr einfach gehalten. Nach dem Schlüsselwort »LOOP« folgt der Anweisungsteil, der unendlich oft ausgeführt werden soll. Dieser Anweisungsteil wird wieder durch das »END« eingrahmt. Die Anweisungen innerhalb der Schleife werden nun unendlich oft durchgeführt, es sei denn, ein Zustand tritt ein, der zum Abbruch der Schleife (nicht des Programms!) führt. Das Verlassen der Schleife ist nämlich ebenfalls möglich. Dazu dient der Befehl »EXIT«. Er bewirkt, daß die Schleife augenblicklich verlassen wird.

Der Anweisungsteil wird also nicht mehr zu Ende ausgeführt. Das Programm wird aber nicht unterbrochen, sondern nach dem Befehl »END« weitergeführt. Sie sehen, im Gegensatz zur Endlosschleife »FOR...TO...DO« ist ein Überarbeiten des Programms durchaus möglich.

Ein Anwendungsbeispiel für eine solche Schleife wäre ein

Die Module Conversions, Heap, RandomNumber und Scan Conversions

Es gelten folgende Deklarationen:

String: ARRAY OF CHAR; Erg: LONGINT; Vorz, Fehler: BOOLEAN;
Basis, Breite: INTEGER; Füllz: CHAR;

Hier die Prozeduren:

StrToVal(String, Erg, Vorz, Basis, Fehler);
ValToStr(Erg, Vorz, String, Basis, Breite, Füllz, Fehler);

Heap

Es gelten folgende Deklarationen:

Adresse: ADDRESS; Größe: LONGINT; Chip: BOOLEAN; Level: INTEGER;

Hier die Prozeduren:

Allocate(Adresse, Größe); Deallocate(Adresse);
Available(Chip); Rückgabewert: LONGINT;
Largest(chip); Rückgabewert: LONGINT;
AllocMem(Adresse, Größe, Chip); Alloc(Adresse, Größe, Chip, Level);
BlockSize(Adresse); Rückgabewert: LONGINT;

RandomNumber

Es gelten folgende Deklarationen:

Nummer: INTEGER; Ursprung: LONGINT;

Hier die Prozeduren:

Random(); Rückgabewert: REAL;
RND(Nummer); Rückgabewert: INTEGER;
GetSeed(Ursprung); PutSeed(Ursprung);

Scan

Es gelten folgende Deklarationen:

LeseProzedur: PROCEDURE(VAR CHAR);
String: ARRAY OF CHAR; Länge: INTEGER; Endezeichen: CHAR;

Die einzige Prozedur lautet:

ScanString(LeseProzedur, String, Länge, Endezeichen);

Zu den Prozeduren:

Hier noch kurze Informationen zu den vier Modulen. »Conversions« beinhaltet zwei Funktionen zum Umwandeln eines Strings in eine Zahl oder umgekehrt. Das Modul »Heap« bietet Funktionen zur Speicherreservierung an. Random ist für das Erzeugen von Zufallszahlen zuständig. Der einzige Befehl in »Scan« dient zum Einlesen einer Zeichenkette. Sie werden nicht alle Variablen Deklarationen verstehen. Machen Sie sich keine Sorgen, die Erklärungen folgen in späteren Kursteilen.

Tabelle: Informationen zu den Modula2-Modulen


```
MODULE Speicher;
(*Hier kommen die IMPORTs und Deklarationen*)
BEGIN
LOOP
ZeigeSpeicheran;
Wenn Schliessgadget dann EXIT;
END;
END Speicher.
```

Modul 5. So könnte eine ständige Speicherplatzanzeige aussehen

Programm, das den freien Speicherplatz jede halbe Sekunde in einem Fenster anzeigt.

Betätigt der Benutzer allerdings das Schließ-Gadget des Fensters, muß das Programm unterbrochen werden. Wie der Programmablauf (nicht das fertige Programm!) aussehen würde, können Sie im Modul »Speicher« sehen.

Anschließend noch ein Beispiel für diesen Befehl. Starten Sie das Modul »Loop«, so wird zuerst der Text »Starte Endlosschleife« ausgegeben und anschließend die Endlosschleife gestartet. Dort wird zuerst überprüft, ob die rechte Maustaste gedrückt wurde.

Ein Bit für die Maus

Sie brauchen diese Zeile nicht zu verstehen. Wir werden den Befehl »IF...THEN...ELSE« aber noch am Ende dieses Kursteils besprechen. Wurde die rechte Maustaste gedrückt, wird die Endlosschleife verlassen, wofür der Befehl »EXIT« sorgt. Wurde die Maustaste nicht gedrückt, wird die Schleife einfach noch einmal ausgeführt. Dies geht so lange weiter, bis der Benutzer einmal die rechte Maustaste gedrückt hat und dadurch die Schleife unterbrochen wird. In diesem Falle wird auch nicht noch einmal der String »Endlos« ausgegeben, sondern es werden direkt die Befehle nach der Schleife ausgeführt. In unserem Beispiel werden zwei Leerzeilen und dann der String »Endlosschleife beendet« ausgegeben. Hier noch kurz eine Anmerkung zum Programm. In der Adresse »00DFF016H« gibt das zehnte Bit Auskunft darüber, ob die rechte Maustaste des linken Mausports gedrückt wurde. In diesem Fall wird das Bit gelöscht, sonst ist es gesetzt. Der Befehl »IF« überprüft dies. Sie werden den Typ »BITSET« im fünften Kursteil kennenlernen. Hier habe ich Ihnen nur schon einmal etwas Hintergrundwissen

in die Hand gegeben. Mit der Schleife »FOR...TO...DO« können wir einen Programmteil eine bestimmte Anzahl mal ausführen lassen; mit der Schleife »LOOP« kreieren wir elegant eine Endlosschleife.

Wer »WHILE« sagt, der muß auch »DO« sagen

Was ist aber, wenn als Abbruchbedingung für eine Schleife nicht irgendein Wert, sondern ein Ereignis genommen werden soll?

Genau für diesen Zweck gibt es die Schleife »WHILE...DO«. Sie ermöglicht es, vor dem ersten Ausführen der Schleife eine Bedingung zu überprüfen. Nur solange diese Bedingung wahr ist, wird die Schleife ausgeführt. Am Anfang eines jeden Durchlaufes wird dann überprüft, ob die Bedingung immer noch wahr ist. Ist dies der Fall, wird der Anweisungsteil ausgeführt, ansonsten wird mit dem Rest des Programmteiles fortgefahren. Hier nun die Syntax der Schleife »WHILE...DO«:

```
WHILE Boolean DO Anweisungen;
END;
```

Wie immer wird die Schleife mit dem Befehl »END« beendet. Gerade habe ich gesagt, daß vor dem Abarbeiten der Schleife überprüft wird, ob ein bestimmtes Ereignis zutrifft. Für diese Abfrage haben wir ja schon einen Datentyp kennengelernt. Es ist der Datentyp »BOOLEAN«. Der Ausdruck nach dem Befehl »WHILE« muß also als Ergebnis immer einen Wert des Typs »BOOLEAN« geben. Zur Erinnerung: Eine Variable, die vom Typ »BOOLEAN« ist, kann nur die Werte »TRUE« und »FALSE« annehmen, also wahr und falsch. Nehmen wir an, in einem Programm kommt folgende Konstruktion vor:

```
VAR Ausfuehren : BOOLEAN; ...
WHILE Ausfuehren DO ...;
END;
```

Die Variable »Ausfuehren« ist also vom Typ »BOOLEAN« und kann somit die Wahrheitswerte wahr und falsch annehmen. Die Schleife wird nun so lange ausgeführt, wie »Ausfuehren« den Wert »TRUE« hat. Damit die Schleife auch einmal beendet werden kann, muß sich irgendwann der Wert des Ausdrucks ändern. Ist dies nicht der Fall, hätten wir mal wieder eine Endlosschleife, denn wenn »Ausfuehren« nicht irgendwann »FALSE« annimmt, wird auch die Schleife nicht unterbrochen. Sie sehen, der Anwender hat viele Möglichkeiten, eine Endlosschleife zu kreieren, allerdings sollten Sie dafür immer »LOOP« verwenden.

Die Schleife »WHILE« werden wir in dieser Form allerdings nicht sehr häufig antreffen. Es ist nämlich möglich, einem Ausdruck den Wahrheitswert wahr (TRUE) oder falsch (FALSE) zuzuweisen.

Nehmen wir dazu ein Beispiel aus dem praktischen Leben. Sie werden gefragt, ob Sie heute Erdbeeren kaufen. Sie antworten: »Solange der Erdbeerpreis unter 2,50 DM liegt, werde ich auch Erdbeeren kaufen.« Wie sieht dieses Beispiel in Modula2 aus? Dazu müssen wir es erst abstrahieren. Wir können kurz schreiben: »Solange Preis < 2,50 DM kaufe ich Erdbeeren«. Damit ergibt die Überprüfung »Preis < 2,50«, ob Sie Erdbeeren kaufen oder nicht. Der Ausdruck »Preis < 2,50« nimmt entweder den Wahrheitswert »TRUE« an, wenn der Preis tatsächlich kleiner als 2,50 ist, und »FALSE«, wenn die Erdbeeren mehr als 2,49 kosten. Somit gibt der Ausdruck »Preis < 2,50« ein Ergebnis vom Typ »BOOLEAN«, also wahr oder falsch. In unserer Konstruktion »WHILE« sähe das dann so aus:

```
WHILE (Preis < 2,50) DO ErdbeerenKaufen;
END;
```

Vergleichsoperationen

Zugegeben, das Beispiel ist fürs Programmieren nicht sehr relevant, aber es zeigt Ihnen zwei Dinge: Ein Vergleich oder ein Ausdruck ergibt ein Ergebnis vom Typ »BOOLEAN«; außerdem ist dies der grundsätzliche Weg von der Idee zur Programmabstraktion. Sie können also als Be-

dingung für das Ausführen einer Schleife »WHILE«, jeden möglichen Ausdruck benutzen, der als Ergebnis einen Boolean-Typ ergibt. Dazu gehören alle Vergleichsoperatoren, wie =, <, >, >=, <=, # (ungleich). Auch der Operator »IN« bei Variablen des Datentyps »BITSET«, den wir noch kennenlernen werden, liefert ein Ergebnis vom Typ »BOOLEAN«.

Häufig findet man in der Umgangssprache Konstruktionen wie »Solange ich Geld habe und das Wetter in Spanien schön ist, bleibe ich dort noch ein bißchen«. Hier ergibt sich der Wahrheitswert der Aussage durch die Verknüpfung von zwei Aussagen, die jeweils beide einen Boolean-Typ liefern. Die erste Aussage lautet »Ich habe Geld«, die zweite »Das Wetter ist in Spanien schön«. Nur wenn beide Bedingungen eintreffen, wird das ausgeführt, was hinter der Bedingung angegeben wird: »Ich bleibe noch ein bißchen in Spanien«.

Genauso, wie wir in der Umgangssprache zur Verbindung von zwei Aussagen das Wort »und« verwenden, benutzen wir in Modula2 dazu den Operator »AND«. Soll überprüft werden, ob zwei Aussagen wahr sind, verbindet man diese Aussagen mit »AND«. Das Ergebnis ist nur dann wahr, wenn beide Einzelaussagen wahr sind. Ist hingegen schon eine falsch, ist der ganze Ausdruck falsch. Das gleiche gilt ebenfalls für den Operator »OR«. Er ist äquivalent zum umgangssprachlichen »oder«. Ein Ausdruck, der aus zwei mit »OR« verbundenen Aussagen zusammengesetzt ist, ist genau dann richtig, wenn mindestens eine der beiden Aussagen richtig ist. Nur wenn beide einzelnen Ausdrücke falsch sind, ist auch der Gesamtausdruck falsch.

Ein weiterer Operator ist »NOT«. Er kehrt den Wahrheitswert einer Aussage um. »NOT(Ausdruck)« ergibt somit als Ergebnis »TRUE«, wenn Ausdruck (Typ: BOOLEAN) »FALSE« ist. Wir erhalten als Ergebnis »FALSE«, wenn Ausdruck »TRUE« ist.

Nach diesem kleinen Ausflug möchte ich Ihnen noch eine Wahrheitstabelle in die Hand geben. Mit ihrer Hilfe können Sie immer ganz einfach nachschlagen, wann ein Ausdruck wahr oder falsch ist. Natürlich

wahr oder falsch ist. Natürlich können Sie diese Operatoren auch beliebig verknüpfen. Sie müssen nur immer auf die richtige Reihenfolge der Abarbeitung achten. Setzen Sie also im Zweifelsfalle Klammern. Kommen wir nun zurück zu unserer Schleife »WHILE«.

Die AND-Verknüpfung

Ausdruck1	Ausdruck2	Ergebnis
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE

Die OR-Verknüpfung

Ausdruck1	Ausdruck2	Ergebnis
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE

Der NOT-Operator

Ausdruck1	NOT (Ausdruck1)
TRUE	FALSE
FALSE	TRUE

Schauen Sie sich einmal das Modul »While« an. Dieses Programm zählt eine Variable hoch und gibt in jedem Schleifendurchlauf den Wert der Variablen aus. Zugegeben, für diese Problemlösung hätten wir auch einfacher die Schleife »FOR...TO...DO« nutzen können. Das Programm hat jedoch nur Lehrwert, außerdem zeigt es, daß mit vielen Schleifen andere Schleifen nachkonstruiert werden können, wenn auch oft viel umständlicher.

Wo findet die Schleife »WHILE« Anwendung? Immer dort, wo ein Ausdruck zu überprüfen ist, bei dem man nicht weiß, wann er den Wert »FALSE« annimmt. Ein Beispiel wäre der Umgang mit Dateien. Möchte man mit dem Amiga eine Datei bearbeiten, so muß sie erst geöffnet werden. Das geschieht über einen entsprechenden Befehl. Dieser Befehl gibt einen sogenannten Zeiger zurück. Hat der Zeiger aber den Wert »NIL«, ist etwas schiefgegangen, auf die Datei kann nicht zugegriffen werden. Das Ganze läßt sich mit der Schleife »WHILE« abfangen:

```
WHILE Zeiger#NIL DO Fuehre-
Dateibefehleaus;
END;
```

Die Dateibefehle würden so nur dann ausgeführt, wenn der Zeiger (was das ist, werden Sie später noch erfahren)

(* Hier steht der Anfang des Programmes *)

```
BEGIN
  Init;
  REPEAT
    Menu;
    ReadInt(Wahl);
    WriteLn;
    CASE Wahl OF
      1: DateiAuswaehlen(DateiHandle, EndPosition);
      2: DatenEingeben(DateiHandle, EndPosition);|
      3: DatenSuchen(DateiHandle, EndPosition)|
      4: DatenAusgeben(DateiHandle, EndPosition);|
      5: Verlassen|
    ELSE
      WriteLn;
      WriteLnString("Falscher Menuepunkt!");
      WriteLn;
    END;
  UNTIL (Wahl=5);
END DVW.
```

Abbildung: Der Menü-Ausschnitt aus einem Programm

ungleich »NIL« ist. Würden Sie trotzdem Dateizugriffe machen, würde Ihnen der Rechner abstürzen. Bei der Schleife »WHILE...DO« wird der Ausdruck immer vor dem ersten Schleifendurchlauf nach seiner Richtigkeit überprüft. So kann es vorkommen, daß die Schleife gar nicht erst durchlaufen wird, nämlich dann, wenn der Ausdruck direkt bei der ersten Überprüfung »FALSE« ergibt. Manchmal ist es jedoch erwünscht, daß eine Schleife auf jeden Fall mindestens einmal ausgeführt wird. Hierzu ist die Schleife »WHILE...DO« unangebracht. Eine Schleife, mit der der Anweisungsteil auf jeden Fall mindestens einmal durchlaufen wird, ist die Schleife »REPEAT...UNTIL«. Sie hat folgende Syntax:

REPEAT Anweisungsfolge; UNTIL
Ausdruck;

wobei Ausdruck wieder entweder TRUE oder FALSE ergeben muß. Wie Sie schon an der Syntax erkennen können, wird bei der Schleife »REPEAT...UNTIL« immer erst nach dem Durchlauf überprüft, ob ein Ausdruck »TRUE« oder »FALSE« ist. Somit wird die Schleife immer mindestens einmal durchlaufen. Auffällig bei dieser Schleife ist, daß Sie nicht mit »END« beendet wird, sondern mit dem Schlüsselwort »UNTIL« und dem dann folgenden Ausdruck. Ansonsten funktioniert die Schleife »REPEAT...UNTIL« genauso wie die Schleife »WHILE...WEND«. Beachten Sie aber, daß Sie immer (!) mindestens einmal ausgeführt wird. Sie ist daher nicht geeignet, wenn auch das einmalige Ausführen einer Schleife zum Fehler führen kann.

Wäre also in unserem obigen Beispiel keine einzige Datei geöffnet worden, würde der Rechner garantiert abstürzen, da beim ersten Schleifendurchlauf ein nicht erlaubter Dateizugriff stattfindet.

Ein Anwendungsbeispiel der Schleife »REPEAT...UNTIL« wäre das Auswerten der Parameter beim Start eines Programms vom CLI aus. Sie wissen, daß im CLI jedem Programm Parameter übergeben werden können. Dies können Programmflags oder Dateinamen sein. Wie die Auswertung im einzelnen erfolgt, soll hier nicht weiter interessieren. Interessant ist nur, daß der Programmname ebenfalls als erster Parameter übergeben wird. So ist sichergestellt, daß mindestens ein Parameter übergeben wurde, nämlich der Programmname, denn wie sollte sonst das Programm gestartet werden?

Die »IF...THEN«-Varianten

Ein weiteres Anwendungsgebiet ist die Programmierung eines Menüs (nicht über Intuition). Das Menü würde dann immer so oft aufgebaut, bis ein Menüpunkt zum Verlassen des Menüs ausgewählt worden wäre. Da das Menü mindestens einmal aufgebaut werden muß, bietet sich hier die Schleife »REPEAT...UNTIL« an. Dieses Prinzip wurde auch bei »DVW« aus dem ersten Kursteil benutzt. Ich werde am Ende dieses Kurses auf dieses Menü noch einmal genauer eingehen, so daß ich hier kein Beispiel für die Schleife »REPEAT...UNTIL« angebe. Wenn Sie interessiert sind, können Sie sich das Programm ja schon einmal anse-

hen und versuchen, es selbstständig zu verstehen.

Als nächstes werden Sie noch zwei weitere Befehle kennenlernen, mit denen Sie zwar keine Schleifen programmieren können, die aber trotzdem sehr wichtig sind. Hierbei handelt es sich um die Anweisung »CASE« und um den Befehl »IF...THEN...ELSE«. In der Umgangssprache macht man Unterscheidungen, die man in der Form »Falls irgendetwas eintritt, dann passiert etwas« antreffen könnte. Auch in Modula2 gibt es eine Konstruktion, mit der Sie diesen Ausdruck nachbauen können. Dies ist die Konstruktion »IF...THEN...ELSE«. Der Befehl hat folgende Syntax:

```
IF Ausdruck THEN Anweisung1;
ELSE Anweisung2;
END;
```

Sie sehen, daß die Konstruktion auch hier mit einem »END« beendet wird. Allerdings bezieht sich das »END« nicht auf den Anweisungsblock 2, sondern auf die ganze Konstruktion. So darf nach dem ersten Anweisungsblock kein »END« folgen. Die Variable »Ausdruck« enthält einen Wert, der entweder den Wahrheitsgehalt »TRUE« oder »FALSE« ergibt. Schauen wir uns dazu ein Beispiel an. Hier ist ein Programm, welches aus einer angegebenen Zahl, Typ »REAL«, die Wurzel zieht. Leider brauchen Sie wieder das Modul »RealInOut«, um dieses Programm zu compilieren. Anwender der PD-Version müssen gegebenenfalls dieses Programm bei einem Bekannten compilieren.

Schauen wir uns einmal das Modul »Wurzel« an. Am Anfang wird wie immer der Modulname angegeben. Anschließend folgen die Import-Anweisungen. Zum Ziehen der Wurzel einer Zahl gibt es in dem Modul »MathLib0« die Funktion »sqrt« (square root). Die »MathLib0« ist übrigens auch nur in der kommerziellen Version vorhanden. Sie sehen, es lohnt sich, das Original zu kaufen. Nach der Variablendeklaration wird nach der Zahl gefragt, deren Wurzel ermittelt werden soll.

Bekanntlich kann man aber nicht aus jeder Zahl die Wurzel ziehen. Die Wurzel einer negativen Zahl (beispielsweise sqrt(-3)) ist nicht definiert. Dies würde zum Abbruch des Programms führen. Also wird vorher mit einer Anweisung »IF...THEN...ELSE« überprüft, ob die Zahl größer oder

gleich null ist. Ist dies der Fall, wird die Wurzel gezogen. Ist die Zahl kleiner als null, wird der Anwender darauf aufmerksam gemacht.

Betrachten wir die Zeile »IF...THEN« etwas genauer. Nach dem Befehlswort »IF« folgt der Ausdruck »Zahl >= 0.0«. Hier wird überprüft, ob die Zahl größer als null ist (die Kommas in der Zeile »WriteRead...« müssen angegeben werden, da es sich um eine Realzahl handelt und diese begrenzt werden muß). Ist dies der Fall, wird der Anweisungsblock nach »THEN« ausgeführt. Dieser endet vor dem Befehl »ELSE«. In diesem Anweisungsblock wird die Wurzel der Zahl mit der bereits erwähnten Funktion »sqrt« ermittelt und ausgegeben. Ist die Zahl jedoch kleiner als null, ergibt der Ausdruck »FALSE«, und der Anweisungsteil nach dem Befehl »ELSE« wird ausgeführt. Hier wird nur ein Infotext ausgegeben. Egal, welcher Anweisungsteil ausgeführt wird, das Programm fährt dann nach dem Befehl »IF...THEN...ELSE« mit den Befehlszeilen nach »END;« weiter fort.

Hier noch ein kleiner Hinweis. In jedem Anweisungsblock wird ein Text ausgegeben. Deshalb sollte bei beiden noch die Anweisung »WriteLn« folgen, damit der Cursor wieder in der nächsten Zeile steht. Man könnte natürlich in jedem Anweisungsblock ein »WriteLn« einfügen, doch das ist unnötig. Da sowieso der Befehl »WriteLn« ausgeführt werden muß, unabhängig davon, welcher Anweisungsteil ausgeführt wurde, ist es sinnvoller, den Befehl »WriteLn« hinter den Befehl »IF...THEN...ELSE« zu setzen. »Warum?« werden Sie jetzt fragen. Nun, auf diese Weise wird der Befehl »WriteLn« nur einmal statt zweimal übersetzt. Das Programm wird dadurch kürzer und kann schneller abgearbeitet werden. Sie sollten von Anfang an auf solche Feinheiten achten, denn wenn Sie dies verinnerlicht haben, werden Ihre Programme später automatisch schneller und effektiver.

Für den Befehl »IF...THEN...ELSE« gibt es noch einige Sonderfälle, denn dieser Befehl muß nicht immer die gleiche Struktur haben. Manchmal kann man beispielsweise auf den Teil »ELSE« verzichten. Lassen Sie in diesem Falle einfach den

Teil »ELSE« weg. Natürlich muß dann noch der Befehl »END;« die IF-Konstruktion abschließen, denn sonst weiß der Compiler ja nicht, wo der Anweisungsteil zu Ende ist. Solch eine Abfrage sähe dann so aus:

```
IF Zahl < 0.0 THEN Write-
String("Zahl kleiner Null");
END;
```

Somit wird die Ausgabe »Zahl kleiner Null« nur dann erfolgen, wenn »Zahl < 0.0« »TRUE« ergibt. Wäre die Zahl größer oder gleich null, würde das Programm hinter der Anweisung »END;« weitermachen. Sie können diese Konstruktion auch ganz einfach mit der »IF...THEN...ELSE« nachbauen, ohne daß sich beim Ablaufen des Programms etwas ändert. Folgende Konstruktion bewirkt genau das gleiche wie die obige:

```
IF Zahl < 0.0 THEN Write-
String("Zahl kleiner Null");
ELSE END;
```

Somit würde zwar der Teil von »ELSE« ausgeführt, wenn die Zahl größer oder gleich null wäre, aber da in diesem Anweisungsblock außer einem direkt terminierenden »END;« nichts steht, folgt keine Aktion. Eine dritte Variante ist die Konstruktion »IF...THEN...ELSEIF«.

Wir "basteln" uns ein Menü

Manchmal müssen mehrere Zustände hintereinander abgefragt werden. Dies geschieht über die verschachtelte Anweisung »ELSEIF«. Diese Konstruktion sieht folgendermaßen aus:

```
IF Ausdruck1 THEN Anweisung1;
ELSEIF Ausdruck2 THEN
Anweisung2; ELSEIF Ausdruck3
THEN Anweisung3;
END;
```

Diese Verschachtelung kann beliebig häufig durchgeführt werden, es können also beliebig viele Anweisungen von »ELSEIF« folgen. Auch hier wird die gesamte Konstruktion wieder mit einem »END;« abgeschlossen. Die einzelnen Anweisungsteile dürfen nicht mit »END;« abgeschlossen werden. Sie sollten dies jedoch einmal probeweise tun, um zu sehen, wie sich der Compiler beziehungsweise hinterher das Programm verhält. Mit dieser Konstruktion können wir schon ein Menü programmieren. Das Programm muß nach der Eingabe

einer Zahl oder eines Buchstabens fragen und dann diesen Buchstaben mit internen vergleichen und entsprechend verschiedene Befehle ausführen. Nehmen wir an, der Anwender soll eine Zahl zwischen eins und drei eingeben. Diese Zahl wird in der Variablen Wahl gespeichert. Die Abfrage sieht dann so aus:

```
IF Wahl=1 THEN Anweisung1; EL-
SEIF Wahl=2 THEN Anweisung2;
ELSEIF Wahl=3 THEN Anweisung3;
ELSE WriteString("Zahl muß
zwischen eins und drei lie-
gen"); WriteLn;
END;
```

So wird bei der Eingabe einer Zahl zwischen eins und drei entsprechend verzweigt; wird eine andere Zahl ausgegeben, wird eine entsprechende Meldung ausgegeben. Sie sehen, so könnten wir schon ein Menü programmieren. Aber es geht noch einfacher. Dazu gibt es in Modula2 die Anweisung »CASE«. Um diese Anweisung am besten erklären zu können, schauen wir uns zuerst einmal die Syntax des Befehls an.

```
CASE Variable OF Wert1:
Anweisung1; Wert2:
Anweisung2; Wert3:
Anweisung3; ELSE Anweisung4;
END;
```

Nach dem Schlüsselwort »CASE« folgt eine Variable. Hier muß unbedingt eine Variable angegeben werden, es darf nicht eine Zahl direkt übergeben werden. Die Variable muß vom gleichen Typ sein wie die der Schleife »FOR...TO...DO«.

Schauen Sie in der Tabelle nach. Nun werden entsprechende Anweisungsblöcke ausgeführt, je nachdem, welchen Wert die Variable beinhaltet. Diese Anweisungsfolgen werden immer durch einen senkrechten Strich (|) abgeschlossen, nicht durch ein »END;«! Nach der Anweisung »CASE« können beliebig viele Werte überprüft werden. Die Anzahl ist nicht wie in unserem Beispiel auf drei beschränkt. Sollte »Variable« keine der angegebenen Werte haben, wird der Teil nach dem Befehl »ELSE« ausgeführt. Dies kann wie bei dem Befehl »IF...THEN« auch weggelassen werden. Dann sieht die Syntax so aus:

```
CASE Variable OF Wert1:
Ausdruck1; Wert2:
Ausdruck2; Wert3: Ausdruck3;
END;
```

Dabei sollten Sie jedoch folgendes beachten: Hat »Varia-

ble« keinen der angegebenen Werte angenommen, also weder Wert1 noch Wert2 noch Wert3, bricht das Programm mit einem Laufzeitfehler ab.

Sie sollten diese Konstruktion nur verwenden, wenn Sie absolut sicher sind, daß »Variable« keinen anderen Wert als den angegebenen annehmen kann. Sollte dies nicht möglich sein, schreiben Sie einfach als Abschluß statt »END;« »ELSE END«. So werden keine Befehle ausgeführt, wenn »Variable« keinen der angegebenen Werte angenommen hat, aber das Programm bricht auch nicht mit einem Laufzeitfehler ab. Um das in diesem Kurs gelernte Wissen auch einmal in der Anwendung zu betrachten, möchte ich das Hauptprogramm der Dateiverwaltung »DVW« hier einmal genauer erklären. Sie sehen diesen Programmteil als Abbildung. Dieses Programm ist **unvollständig**, es läßt sich daher nicht compilieren. Dabei handelt es sich ja auch nur um den Ausschnitt aus einem größeren Programm. Nach dem ersten Befehl »BEGIN« wird ein Befehl namens »Init« aufgerufen. Dabei handelt es sich um einen selbstgebastelten Befehl.

Dann folgt eine Schleife »REPEAT...UNTIL«. In ihr wird zuerst der Befehl »Menu« aufgerufen. Auch dieser Befehl ist selbst gebastelt. Hierbei handelt es sich streng genommen um eine Prozedur. Ich werde im nächsten Kursteil genauer darauf eingehen. Der Befehl »Menu« baut das Menü auf. Er gibt die entsprechenden Auswahlmöglichkeiten auf dem Bildschirm aus. Dann kann der Anwender eine Zahl eingeben, die in der Variablen »Wahl« gespeichert wird. Nun wird mittels »CASE« überprüft, ob »Wahl« den Wert eins angenommen hat. Ist dies der Fall, wird der Befehl »DateiAuswählen« aufgerufen. Ist dies nicht der Fall, wird überprüft, ob »Wahl« den Wert zwei angenommen hat. Ist dies der Fall, wird der Befehl »DatenEingeben« aufgerufen. Diese Überprüfung geht so lange weiter, bis der Vergleich Wahl=Zahl »TRUE« ergibt.

Hat »Wahl« einen Wert kleiner eins oder größer fünf, wird der Teil »ELSE« ausgeführt, womit der Anwender darauf aufmerksam gemacht wird, daß der ausgewählte Menüpunkt nicht existiert.

Anschließend wird mit der Anweisung »UNTIL« (sie gehört zur Schleife »REPEAT«) überprüft, ob »Wahl« den Wert fünf hat. Ist dies nämlich der Fall, wird die Schleife »REPEAT« verlassen, ansonsten wird sie wiederholt. Sie wird genau dann verlassen, wenn »Wahl« den Wert fünf hat, da dies der Menüpunkt zum Verlassen des Programms ist. So viel nun zu den Schleifen und

Kontrollstrukturen. Führen Sie sich das hier gesagte gut zu Gemüte, denn Sie werden diese Befehle sehr häufig brauchen. Machen Sie sich auch die Unterschiede zwischen den einzelnen Schleifen klar. Wir sehen uns dann im nächsten Kursteil wieder, in dem Sie das Prozedurprinzip kennenlernen werden. Bis dann!

(cd)

Hinweise zur Literatur:

- [1] M2-Amiga Compilerhandbuch
- [2] Krüger I.: Amiga – Programmieren mit Modula2; Markt & Technik 1988, ISBN 3-89090-554-4
- [3] Gzella H.: Amiga Modula2 – Programmieren für Fortgeschrittene; Markt & Technik 1989, ISBN 3-89090-744-x
- [4] Dal Cin, Lutz, Risse: Programmierung in Modula 2; Teubner Verlag 4. Auflage 1989, ISBN 3-519-02280-x
- [5] Bleck, Dittrich, Gelfand, Jennrich, Schemmel, Schulz: Amiga Intern, Data Becker Verlag 1990, ISBN 3-89011-398-2

Listings

```
MODULE Ausgabe;

FROM InOut IMPORT WriteString, WriteLn;

VAR i : INTEGER;

BEGIN
  FOR i:=1 TO 10 DO
    WriteString("Hallo");
    WriteLn;
  END;
END Ausgabe.
```



```
MODULE Fakulttaet;

FROM InOut IMPORT WriteString, WriteLn, WriteCard,
  ReadLongCard;

VAR Zahl, Laufvar, Ergebnis : LONGCARD;

BEGIN
  Ergebnis:=1;
  WriteString("Geben Sie eine Zahl ein: ");
  ReadLongCard(Zahl);
  FOR Laufvar:=1 TO Zahl DO
    Ergebnis:=Ergebnis*Laufvar;
  END;
  WriteCard(Zahl, 3);
  WriteString(" Fakulttaet ist ");
  WriteCard(Ergebnis, 6);
  WriteLn;
END Fakulttaet.
```

```
MODULE Runterzaehlen;

FROM InOut IMPORT WriteInt, WriteLn;

VAR Laufvar : INTEGER;

BEGIN
  FOR Laufvar:=10 TO -10 BY -3 DO
    WriteInt(Laufvar, 3);
    WriteLn;
  END;
  WriteInt(Laufvar, 3);
  WriteLn;
END Runterzaehlen.
```

```
MODULE Endlos;

FROM InOut IMPORT WriteString;

VAR i : INTEGER;

BEGIN
  WriteString("Nochmal ");
  FOR i:=1 TO 10 DO
    WriteString("und nochmal... ");
    i:=1;
  END;
END Endlos.
```

Listing Modula-Kurs

```
MODULE Loop;

FROM InOut IMPORT WriteString, WriteLn;
FROM SYSTEM IMPORT BITSET;

VAR Maus[00DFF016H] : BITSET;
    i : INTEGER;

BEGIN
  WriteString("Starte Endlosschleife");
  WriteLn;
  WriteLn;
  LOOP
    IF NOT(10 IN Maus) THEN
      EXIT;
    END;
    WriteString("Endlos ");
  END;
  WriteLn;
  WriteLn;
  WriteString("Endlosschleife beendet");
  WriteLn;
END Loop.
```

```
MODULE While;

FROM InOut IMPORT WriteInt, WriteLn;

VAR i : INTEGER;

BEGIN
  i:=1;
  WHILE (i<=10) DO
    WriteInt(i, 3);
    WriteLn;
    i:=i+1;
  END;
  WriteInt(i, 3);
  WriteLn;
END While.
```

```
MODULE Wurzel;

FROM InOut IMPORT WriteLn, WriteString;
FROM RealInOut IMPORT ReadReal, WriteReal;
FROM MathLib0 IMPORT sqrt;

VAR Zahl : REAL;

BEGIN
  WriteString("Geben Sie eine Zahl ein: ");
  ReadReal(Zahl);
  IF Zahl>=0.0 THEN
    WriteString("Wurzel ");
    WriteReal(Zahl, 6, 3);
    WriteString(" ist ");
    WriteReal(sqrt(Zahl), 6, 3);
  ELSE
    WriteString("Zahl ist kleiner als Null!");
  END;
  WriteLn;
END Wurzel.
```

Listing Modula-Kurs

Commodore hat es nun endlich geschafft: Während alle Welt noch über die Versionen 1.4 und 1.5 des AmigaDOS speulierte, hat Commodore mit dem Amiga 3000 gleich eine Stelle vor dem Komma zugelegt und zählt die Versionen nun ab 2.x – für unseren Kurs wollen wir uns jedoch mit den inzwischen betagten Versionen 1.2 und 1.3 zufriedengeben... Sie haben es in der letzten Folge wohl schon bemerkt: Obwohl wir in der ersten Folge des Kurses auf eine (für den Insider schon etwas betagte) Version des AmigaDOS Bezug genommen haben, konnten wir es uns in der letzten Folge nicht verkneifen, schon auf eine etwas neuere Version Bezug zu nehmen. Es geht hier um die Beziehungen zwischen Version 1.2 und 1.3 von AmigaDOS. Schauen wir uns in den beiden folgenden Teilen des Kurses nun an, was Commodore Neues zu bieten hat – oder hatte ...

Von 1.2 zu 1.3 – kann eine Stelle nach dem Punkt wirklich soviel ausmachen? Bei Commodore schon, zumal dort bei der Zählung der diversen Varianten eines Betriebssystems wesentlich feinere Unterscheidungen an der Tagesordnung sind. Die Bestimmung der aktuellen, das heißt beim Booten geladenen Version des Betriebssystems sollte kein größeres Problem darstellen – denkt man.

Die Praxis sieht jedoch etwas anders aus. Zum einen besteht AmigaDOS eigentlich aus zwei Teilen: Dem sogenannten Kickstart, welches auf dem A1000 noch über Diskette geladen, bei den neueren Rechnern jedoch schon in der Hardware (ROM) verewigt wird, und den übrigen Teilen, die auf der Systemdiskette vorhanden sind. Zu diesen Teilen gehören zum einen die DOS-Befehle, zum anderen aber auch Gerätetreiber und die »Libraries«. Sofern man nicht genau weiß, welche Version von Kickstart im Rechner eingebaut ist, sollte man einen Blick auf die »Diskettenhand« beim Einschalten oder auf die Meldung »Release« während des »Bootens« werfen. Hier wird die entsprechende Version angezeigt. Die Version der Workbench steht im allgemeinen auf der Diskette. Was ist aber, wenn mit einer fremden Diskette gebootet wurde? Klarheit könnte letztendlich eine Anfrage beim

Michael Anton

Das AmigaDOS – leicht und verständlich!

Neu dabei bei 1.3

Nach der Vorstellung des Amiga 3000 hat Commodore zwar eine neue Generation des AmigaDOS eingeführt, wir bleiben in unserem Kurs jedoch bei den etwas betagteren Vertretern dieses Betriebssystems. Diese Folge unseres CLI-Kurses soll die Version 1.3 von AmigaDOS näher beleuchten.

Kursfahrplan

Teil 1: – Einführung in das CLI

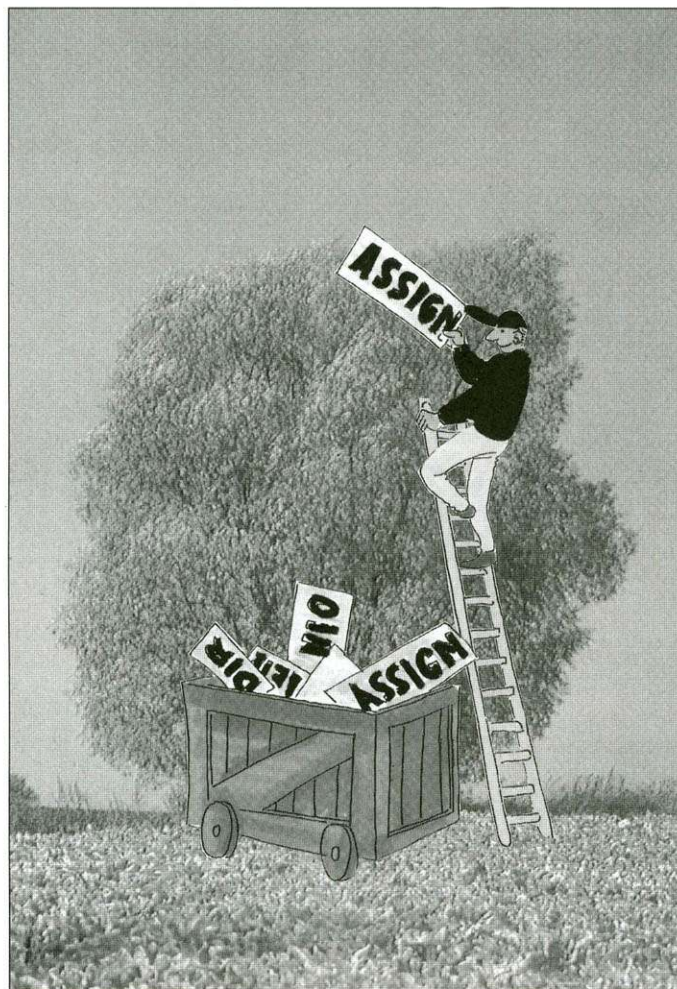
Teil 2: – Was kann AmigaDOS

Teil 3: – Stapeldateien

Teil 4: – Neu dabei bei 1.3

Teil 5: – Alternativen zum CLI

Teil 6: – Multitasking



Amiga selbst schaffen. Der Befehl hierzu lautet »VERSION«. Probieren Sie's mal aus, und Sie werden eine Überraschung erleben – mit 1.2 und 1.3 haben die ausgegebenen Zahlen absolut nichts zu tun. Aber im Prinzip ist's doch ganz einfach: Eine 33 vor dem Komma steht für Version 1.2, eine 34 für 1.3. Die Stellen nach dem Komma geben die Nummer der Zwischenversion an. Da das Betriebssystem in der Anfangsphase öfters geändert wurde, ist diese Art der Numerierung recht sinnvoll. Je höher diese Zahlen, desto neuer ist die verwendete Version. Stabile und relativ neue Versionen für Kickstart sind 33.180 beziehungsweise 34.5, für die Workbench 33.56 oder 34.21. Ab 1.3 bietet der Befehl »VERSION« außer der eigentlichen Anzeige noch andere Funktionen. So können mit ihm auch die Versionen der Treiber und »Libraries« abgefragt werden, wenn der Name der entsprechenden Datei als Parameter übergeben wird und diese Datei im entsprechenden Verzeichnis vorhanden ist.

VERSION printer.device

würde die entsprechenden Daten für den Druckertreiber angeben. In Scripts kann darüber hinaus auch abgefragt werden, ob eine bestimmte Version vorhanden ist, indem die benötigte Version als zweiter Parameter übergeben wird. Über den Returncode wird dies angegeben, wobei ein Wert von 0 bedeutet, daß die Version der »Library« oder des »Device« dem übergebenen Wert entspricht oder größer (gleichbedeutend mit »neuer«) ist, Returncode 5 bedeutet, daß nur eine ältere Version vorliegt. (Die Auswertung der Returncodes kennen wir schon von der letzten Folge.) Eine solche Abfrage kann manchmal nützlich sein, da einige Programme nicht mit älteren »Libraries« zusammenarbeiten. Lohnt sich nun der Umstieg auf die neuere Version? Mit ziemlicher Sicherheit. Einen kleinen Vorgeschmack haben Sie ja schon in der letzten Folge bekommen, diese und die nächste Folge werden auf die Vorteile der Version 1.3 noch näher eingehen. Der Weg zur neuen Version ist nicht sehr kompliziert. Für knapp 80 DM ist ein Update-Set erhältlich – bestehend aus drei Disketten und einem Handbuch. Wenn Sie Glück haben, finden Sie

Die Steuerung von MORE

[SPACE]	Seite vorblättern
[RETURN]	Nächste Zeile
[BACKSPACE]	Seite zurückblättern
<	Textanfang
>	Textende
%xx	zur Stelle xx% springen
q, [CTRL+C]	Beenden
/(String)	Suche Zeichenfolge mit...
.(String)	...und ohne Beachtung von Groß- und Kleinschreibung wiederhole Suche
n	Editor starten, der Name muß in der Umgebungsvariablen »EDITOR« definiert sein.
[SHIFT+E]	

wir uns nun innovativeren Aspekten der Version 1.3 zu.

...und Innovationen...

Viele Befehle wurden in 1.3 mit zusätzlichen Optionen ausgestattet, die mehr oder weniger sinnvolle Operationen erlauben. So wurde »AS-SIGN« mit neuen Optionen ausgestattet, die die Arbeit mit Geräten verbessern. Die Option »REMOVE« erlaubt es, mit »MOUNT« angemeldete

Geräte wieder abzumelden. Mit der Option »EXISTS« kann überprüft werden, ob ein bestimmtes Gerät überhaupt vorhanden ist. Wenn ja, so wird 0 als Returncode übergeben, was in Scripts, wie bereits bekannt, abgefragt werden kann.

Auch »COPY« wurde überarbeitet, mit der Option »DATES« bleiben Datum und Uhrzeit der Kopien auf ihrem alten Wert, »COM« bewirkt, daß auch Kommentare zu den Dateien mitkopiert werden. Die

Tabelle: Die Möglichkeiten von »MORE«

vielleicht auch auf den Disketten kommerzieller Produkte eine recht vollständige Version des eigentlichen Systems, im Hinblick auf eine komplette Version inklusive Kickstart- und Extras-Diskette sowie der Anleitung sei jedoch die Investition in das offizielle Set empfohlen!

Schönheitsoperationen...

Das Spektrum der diversen Änderungen reicht vom simplen Face-Lifting über neue Optionen für bereits bekannte Befehle bis hin zu völlig neuen Befehlen, natürlich wurden auch bekannte Fehler der älteren Versionen ausgebessert. Manchem Programm wurden nur neue Schlüsselwörter verpaßt, die an der eigentlichen Funktion nichts ändern und somit Makulatur sind. Ihre einzige Funktion besteht darin, die Syntax von Befehlen übersichtlicher zu machen – oft arten sie jedoch in unnötige Tipparbeit aus. Nehmen wir mal den bereits

bekannten Befehl »ASK« dessen Syntax

ASK "Frage"

ist. In der Version 1.3 kann (muß aber nicht!) zusätzlich noch das Schlüsselwort »PROMPT« angegeben werden, also

ASK PROMPT "Frage"

Ähnliches auch bei »FAIL-AT«, wo vor der Fehlergrenze zusätzlich noch »RCLIM« angegeben werden kann oder ein "überflüssiges" »DEVICE« in Verbindung mit »INFO«. Nicht unbedingt ein großer Fortschritt, man sollte sich gut überlegen, ob man sich solche tippintensiven und dennoch überflüssigen Eingaben angewöhnt. Etwas mehr Sinn machen Erweiterungen, wie sie bei »TYPE« vorgenommen wurden. Dort sind nun statt »OPT N« und »OPT H« auch die Parameter »HEX« und »NUMBER« erlaubt, um ein bestimmtes Ausgabeformat zu bewirken. So etwas macht natürlich keine besonderen Verbesserungen eines Betriebssystems aus, wenden

Die Optionen von EVAL

Genereller Aufruf:

EVAL Wert1 Operation Wert2 TO Datei LFORMAT
Format

Werte:

Zulässig sind nur ganze Zahlen. HEX-Zahlen müssen durch vorangestelltes '0x' oder '#x' gekennzeichnet werden, Oktalzahlen mit '#' oder '0'. (Vorsicht also bei der Angabe von Dezimalzahlen mit führenden Nullen!) Zeichen mit vorangestelltem Hochkomma werden als ASCII-Codes ausgewertet.

Rechenoperationen:

+	Addition
-	Subtraktion
*	Multiplikation
/	Division
MOD	Rest bei Modulo-Division
&	logisch AND
	logisch OR
XOR	logisch XOR
EQV	logisch Äquivalenz
<<	bitweise links schieben
>>	bitweise rechts schieben

Bei Schiebefehlern bedeutet Wert2 die Anzahl der Bits

Ausgabeformate:

%Xn	n-stellige HEX-Zahl
%On	n-stellige Oktalzahl
%In	n-stellige Dezimalzahl
%Tn	n-facher Tabulatorvorschub
%I	Dezimalzahl
%C	ASCII-Zeichen
*N	Zeilenvorschub nach Ausgabe

Tabelle: Rechenfunktionen von »EVAL«

Auch in Ihnen schlummern verborgene Talente



SIE sind leidenschaftlicher Spieler.

SIE programmieren natürlich gern selbst Spiele.

Lassen **SIE** Ihre programmierten Spiele nicht in der Schublade oder in irgendeiner Ecke Spinnweben ansetzen.

Wenn **SIE** nicht wissen, wohin mit Ihrem Listing, schicken **SIE** es an den

DMV-Verlag
Redaktion AMIGA DOS
Postfach 250
Stichwort: Spieleprogrammierung
3440 Eschwege

Option »CLONE« ist eine Kombination aus »DATES« und »COM«. Minimale Änderungen wurden auch an »DIR« und »ECHO« vorgenommen, sie sind jedoch ohne größeren Wert. Praktisch dagegen ist die Option »QUICK« von »FORMAT«, mit der bereits formatierte Disketten schnell gelöscht werden können.

Ähnliches auch bei »INSTALL«, hier kann mit »NOBOOT« eine Diskette als nicht bootfähig deklariert werden, mit »CHECK« kann überprüft

werden, ob der Bootblock dem Standard entspricht – eine einfache Art von Virenschutz. »LIST« wurde mit einigen zusätzlichen Parametern zur Gestaltung der Ausgabe versehen; näheres kann dem Handbuch entnommen werden.

...und völlig Neues!

Soviel zu den wichtigsten Neuerungen bekannter Befehle in der Version 1.3. Wieweit diese verwendbar sind, muß

die Praxis im Einzelfall zeigen. AmigaDOS 1.3 hat aber noch mehr zu bieten!

Neben mehr oder weniger praktischen Erweiterungen der alten Befehle kann die Version 1.3 auch mit völlig neuen Befehlen aufwarten. Dies können einfache Zusatzprogramme sein, die nur selten Verwendung finden, wie die Befehle »FF« und »FIXFONTS« zur schnelleren Ausgabe von Texten oder der schnelleren Arbeit mit Zeichensätzen, hinter ihnen können sich je-

doch auch sehr praktische Erweiterungen verbergen.

Ablösung für »TYPE« liefert der neue Befehl »MORE«. Mit ihm können Dateiinhalte bequem durchblättert werden, das Glücksspiel mit dem Anhalten der Ausgabe entfällt in Zukunft. Nebstehende Tabelle zeigt die Optionen, die innerhalb von »MORE« verfügbar sind, der Aufruf erfolgt mit

MORE Dateiname

In manchen Fällen kann auch der Befehl »CMD« recht nütz-

Glossar

In dieser Folge des Kurses wurden einige Tatsachen kurz angesprochen, die doch noch etwas genauerer Erklärung bedürfen, zumindest ansatzweise...

Es soll hier keine komplette Einführung in die Interna des Amiga geliefert werden, sondern nur ein kleiner Einblick. Dies ist nötig, um einige im Kurs genannte Begriffe näher zu erläutern, zum anderen ist es auch ein kleiner Trick, um einige Befehle, die sich der Einordnung in den Kurs störrisch widersetzen, unterzubringen.

Libraries und Devices

Die Bestandteile von AmigaDOS sind in oft recht kleine Einheiten aufgesplittet, und der Großteil von ihnen muß von der Diskette nachgeladen werden. Dies macht die Arbeit zwar etwas langsam, erleichtert aber die Pflege des Systems, da bei Änderungen an einem Teil nicht gleich das ganze System neu gebaut werden muß. Wo liegen nun aber die Leistungen des Systems begraben? In einem gelungenen Zusammenspiel aus Hard- und Software, dessen Konzept dem Programmierer erlaubt, mit einfachsten Mitteln komplexe Vorgänge auszulösen. Auf der Hardware-Seite stehen die sogenannten Devices zur Verfügung, Steuerprogramme für bestimmte Geräte. Diese Treiber, die im Verzeichnis »DEVS:« stehen, werden bei Bedarf vom (Boot-) Laufwerk nachgeladen. So wird beispielsweise vor dem Start einer Druckerausgabe der Treiber »Printer.Device« nachgeladen, soweit er noch nicht im Speicher ist, für Diskettenoperationen wäre das »Trackdisk.Device« zuständig. Allerdings gibt es auch Geräte, die nicht diesem Automatismus unterliegen. Diese müssen separat angemeldet wer-

den, was über das Programm »MOUNT« geschieht. Prominente Beispiele solcher Geräte sind »RAD:«, »AUX:« oder »SPEAK:«, die ab 1.3 verfügbar sind. Beim Aufruf von »MOUNT« mit dem Gerätenamen als Parameter, beispielsweise

MOUNT SPEAK:

wird der Treiber für die Sprachausgabe geladen, und »SPEAK:« steht danach zur Verfügung. (Erfolgt ein Zugriff auf ein solches Gerät vor der Anmeldung, erscheint die Meldung "device not mounted" auf dem Bildschirm.) Allerdings ist es mit einem einfachen Aufruf von »MOUNT« nicht getan, statt dessen muß an bestimmter Stelle eine genauere Beschreibung des Gerätes stehen. Zuständig ist die Textdatei »MountList« im Verzeichnis »DEVS:«, die eine passende Definition enthalten muß. Ein Blick auf diese Datei mag vielleicht verwirrend sein, aber Sie sollten sich keine grauen Haare wachsen lassen: Für die wichtigsten Geräte sind bereits Einträge vorhanden, bei Neuanschaffungen von Hard- oder Software stehen meist irgendwo Beispiele für notwendige Einträge in die »MountList« – entweder auf der Diskette oder in der Anleitung. (In der originalen Startup-Sequence von 1.3 werden die Geräte meist schon automatisch geladen!) Ein anderes Phänomen des Amiga sind seine Libraries. Sie stellen im wesentlichen eine Sammlung von Unterprogrammen dar, auf die Betriebssystem und Programmierer gleichermaßen zugreifen können. So muß, um beispielsweise ein Fenster zu öffnen, nicht jedesmal ein Programm mit den notwendigen Kleinarbeiten geschrieben werden. Es reicht statt dessen, wenn der Program-

mierer dem Betriebssystem die Anweisung gibt, aus einer bestimmten Bibliothek eine bestimmte Routine auszuführen. Dieses Konzept macht den Amiga zu einer besonders einfach zu programmierenden Maschine, auf der sich unterschiedliche Programmiersprachen in der gleichen Form profilieren können. (Zumindest in der Theorie, die Praxis setzt dennoch erhebliche Kenntnisse voraus, aber das ist eine ganz andere Geschichte...)

Speicherverwaltung

Sie werden bestimmt schon über Begriffe wie »Chip-Mem« oder »Fast-Mem« gestolpert sein. »Mem« steht für Memory, also Speicher. Soweit kein Problem, aber was sollen die Feinheiten in der Unterscheidung? Nun, sie werden durch die Hardware des Amiga nötig, denn Speicher ist dort nicht immer gleich Speicher. Das liegt daran, daß im Amiga neben dem Hauptprozessor auch noch einige Co-Prozessoren, sogenannte Custom-Chips, ihre Dienste verrichten. Das »Chip-Mem« ist nun ein Bereich des Speichers, auf den sowohl Haupt- als auch die Co-Prozessoren zugreifen können. Dies sind im Normalfall 512 kByte Speicher, bei neueren Geräten auch 1 MByte. Jedes Mehr an Speicher ist jedoch nur für den Hauptprozessor zugänglich, die für Grafik und Sound zuständigen Co-Prozessoren haben von diesem »Fast-Mem« nichts, obwohl gerade sie durch die Komplexität ihrer Aufgaben möglichst viel Speicher benötigen. Was sie nun genau im Gerät haben, und wieviel davon noch verfügbar ist, zeigt Ihnen der Befehl »AVAIL« (nur 1.3). Die Unterscheidung zwischen

Chip- und Fast-Mem kann zu Komplikationen führen. Zum einen sind ältere Programme nicht auf ein Mehr an »Chip-Mem« eingerichtet, der Betrieb auf neueren Geräten kann so zu Abstürzen führen. Andererseits vergeben viele Programme den Speicherplatz nach dem Kriterium "Was gerade vorne ansteht, wird auch benutzt". Dies ist insofern tragisch, als an vorderster Stelle wertvolles »Chip-Mem« steht und somit auch an Anwendungen "verschwendet" werden kann, bei denen normalerweise auch »Fast-Mem« genügen würde.

Als Beispiel seien die Optionen »RESIDENT« und »RAM:« genannt. Zur besseren Ausnutzung des Speichers wurden in der Version 1.3 einige Befehle zur effektiveren Verwaltung des Speichers vorgesehen. Der Befehl »NOFASTMEM« war schon bei 1.2 verfügbar, mit ihm wird vorhandenes Fast-Mem ganz einfach abgeschaltet. Neu dagegen ist »FASTMEMFIRST«, mit dem die Vergabe von Fast-Mem erzwungen werden kann.

Dies ist aber nur bei älteren Speichererweiterungen nötig, neuere Zusätze sind nur selten problematisch. »MERGEMEM« dient dazu, den Speicherplatz auf verschiedenen Karten zu möglichst großen Bereichen zusammenzufassen, da AmigaDOS solche Bereiche besonders gern hat. Bei neueren Erweiterungen, die sich selbständig in das System einbinden (Auto-Config) werden diese Befehle nur selten benötigt. Falls Sie Probleme mit manchen Programmen haben sollten, können Sie jedoch versuchen, die Probleme mit diesen Befehlen zu lösen.

lich sein. Mit ihm können Ausgaben, die an die serielle oder parallele Schnittstelle gehen, in Dateien oder an andere Geräte umgeleitet werden. Die Syntax ist

CMD Port Datei

wobei als Port nur »SERIAL« oder »PARALLEL« zulässig ist, als Datei können sowohl echte Dateien als auch Geräte (siehe letzte Folge) genommen werden.

CMD SERIAL df0:Protokoll

schreibt in der Datei "Protokoll" auf, was an die serielle Schnittstelle gerichtet war, mit

CMD PARALLEL SPEAK:

wird hörbar, was nicht gedruckt werden soll. Es empfiehlt sich, »CMD« stets als Hintergrundtask (»RUN CMD...«) zu starten, aber dazu in der übernächsten Folge mehr.

Nicht unbedingt praktisch ist der Befehl »WHICH«, mit dem der Fundort einer bestimmten Datei als kompletter Pfad ausgegeben wird. Allerdings erfolgt die Suche nur in der Liste der residenten (siehe unten) Programme und des mit »PATH« eingestellten Suchwegs. »WHICH« kann leider nicht dazu verwendet werden, Dateien "irgendwo" zu suchen. Einzig in Scripts kann er praktisch sein, wenn vor dem Aufruf eines Programms überprüft werden soll. Ein Returncode von 0 bedeutet, daß das Programm vorhanden ist. Ein sehr praktischer Neuling ist der Befehl »RESIDENT«, der insbesondere bei Rechnern ohne Festplatte enormen Zuwachs an Geschwindigkeit bringen kann. Wie bereits bekannt, müssen alle Befehle des AmigaDOS von einem Da-

tenträger geladen werden, da das CLI selbst keine Befehle erkennt. Dies führt oft zu Verzögerungen – vor allem dann, wenn man nur mit Disketten und gar nur einem Laufwerk arbeitet.

Nie wieder Disketten?

Der Weg zur schnelleren Arbeit führte bislang über die RAM-Disk, auf die einfach die wichtigsten Kommandos kopiert und dann von dort geladen wurden. Aber jetzt geht es noch schneller, denn mit »RESIDENT« können Programme sofort in den Arbeitsspeicher geladen und dort bei Bedarf ausgeführt werden. Da das zusätzliche Kopieren von der RAM-Disk in den Speicher entfällt, ist diese Lösung sogar noch schneller – aber diese Zeit ist kaum meßbar: Ein mit »RESIDENT« geladenes Programm steht sofort zur Verfügung.

Der einfachste Weg, ein Programm resident zu machen, ist

RESIDENT Programm

Wer gerne viel tippt, kann dahinter auch noch »ADD« schreiben, das ist jedoch freigestellt. Soll ein residentes Programm wieder entfernt werden, so geht dies über die Option »REMOVE«:

RESIDENT Programm REMOVE

Als Spezialität kann dem geladenen Programm gleich noch ein neuer Name gegeben werden, indem sowohl der neue Name als auch die Datei, in der das Programm steht, übergeben werden. Dabei ist normalerweise die Reihenfolge zu beachten (zuerst der neue, dann der alte Name), notfalls kann jedoch auch mit den

Schlüsselwörtern NAME und FILE gearbeitet werden:

RESIDENT NAME Zeige FILE Type

Dieses Beispiel macht den Type-Befehl resident, der nun unter dem Namen »ZEIGE« aufgerufen werden kann. Die Möglichkeiten, die sich durch die Optionen ergeben, können Sie schon ahnen...

Leider hat die Sache einen kleinen Haken – es können nämlich nicht beliebige Programme resident gemacht werden. Zum einen lohnt es sich natürlich kaum, sehr große Programme ständig im Speicher zu halten (es sei denn, man besitzt 4 MByte oder mehr davon), zum anderen stellt das Betriebssystem gewisse Anforderungen an die zu ladenden Programme. Daß diese Anforderungen erfüllt werden, kennzeichnet das (P)ure-Attribut (siehe letzte Folge), ist dieses Flag bei einem Programm nicht gesetzt, verweigert »RESIDENT« die Übernahme. Mit dem Zusatz »PURE« kann diese jedoch erzwungen werden:

RESIDENT Vorsicht PURE

Allerdings geht man dann das Risiko ein, daß dieses Programm früher oder später einen Absturz verursacht. Die meisten DOS-Befehle und Utilities sind in der Hinsicht jedoch unbedenklich, bei anderen Programmen helfen nur Experimente.

Intelligenter Scripts

Speziell für "Hochstapler" hat die Version 1.3 einige Schmankele in der besseren Programmierung von Scripts zu bieten. Mit »EVAL« steht eine Art Taschenrechner für den CLI zur Verfügung, mit dem recht viele Operationen

möglich sind – allerdings nur mit ganzen Zahlen.

Nebenstehende Tabelle zeigt eine Übersicht über die Operationen von »EVAL«. Die Ergebnisse können sowohl im CLI ausgegeben als auch in Dateien umgeleitet werden. Wenn Sie also wissen möchten, wieviel 1+1 ist, brauchen Sie nur

EVAL 1+1

eingeben. Ebenfalls neu sind die sogenannten Environment- oder Umgebungsvariablen. Es verhält sich dabei ähnlich wie bei "normalen" Programmiersprachen, denen Werte frei zugewiesen werden können, hierzu dienen die Befehle »SETENV« und »GETENV«. Steht in einem Script dann der Name einer solchen Variablen, so wird an diese Stelle deren Inhalt gesetzt. Dies sollte jedenfalls der Normalfall sein, derzeit handelt es sich bei diesen Variablen jedoch noch um ein Provisorium, welches manchmal mit Vorsicht zu genießen ist. Im Prinzip sind sie eine recht nützliche Angelegenheit, der Umgang mit ihnen erfordert jedoch etwas Fingerspitzengefühl.

Ausblick

Ich hoffe, daß Sie einen ersten Einblick in die Leistungsfähigkeit der "neuen" DOS-Version bekommen haben. Ein Umstieg lohnt sich wohl auf alle Fälle, denn es werden mit ihr nicht nur alte Fehler behoben, sondern auch neue Funktionen eingeführt. Damit wollen wir die Version 1.3 bis zur nächsten Folge ruhen lassen.

(cd)



AMIGA -- BTX MULTITERM !!! PRO !!!

Der absolute BTX Software-Decoder.

Preise:

Multiterm an A-Koppler / Modem	125,-DM
Multiterm Pro	149,-DM
Multiterm an Post-Modem DBT03	195,-DM
Multiterm Pro	229,-DM
Sep. Interface Amiga-DBT-03 nur	75,-DM

Das richtige Modem !!!!

Best 2400 Plus (Btx-fähig)*	429,-DM
Discovery 2400C*	359,-DM
U.S. Robotics HST 14.400*	1560,-DM
U.S. Robotics Dual Standard*	2390,-DM

Weitere Modelle Robotics/Lightspeed etc. auf Anfrage
* Betrieb am Postnetz der BRD und West-Berlin bei Strafe verboten.
512K Speicher mit Uhr f. A500 189,-DM
Info über MS-DOS Rechner und weitere Neuheiten anfordern.

Kopierschutz passe', jetzt kommt: Project D

Disketten Utility der Sonderklasse

Das Original Kopierprogramm direkt aus USA.
Mit Update-Service.

incl. deutscher Anleitung 80,-DM

DIGI-View 4.0 298,-DM
Pagestream 1.8 339,-DM
Pagesetter II 189,-DM
weitere Software zu Superpreisen !!!!
Kickstart ROM 1.3 59,-DM

Neuheiten !!!

Interne PC-Karte f. Amiga 500 788,-DM
Interne 20 MB Festplatte " " 1299,-DM

Für XT / AT-Karten
Flugsimulator 4.0
Die neueste Version von Microsoft.
Direkt aus USA. 129,-DM

M. Kirschbaum Medienberatung
Schubertstr. 3, 4320 Hattingen
Tel. + Btx: 02324/82249 --- Fax: 02324/83722

SECOND HAND COMPUTER

Ankauf * Verkauf * Vermittlung * Inzahlung

Gebraucht-Computer:

- * Zubehör
- * Neugeräte
- * alle Marken
- * Konkursware-Ankauf
- * Ankauf defekter Geräte

Angebote solange Vorrat

ATs z.B.: 12 MHz, NEU, 512 KB, 102Tast., 200-Watt-Netz. 999,-

386-SX z.B.: 16 MHz, NEU, 1 MB, 102Tast., 200-Watt-Netz. 1549,-

386er: COMPAQ, IBM, TANDON, No Name (16 bis 33 MHz) auf Anfrage

FESTPLATTEN: z.B. 760 MB, ESDI oder SCSI, neuwertig 4795,-

weitere Festplatten von 10 bis 350 MB ab 195,-

WO SONST ?

Wir kaufen und verkaufen:

- * Homecomputer **NEU:**
- * XTs und ATs
- * Büroanlagen
- * Anrufbeantworter

Jetzt auch in der DDR!

CO-PROZESSOREN: z.B. 80387-25 798,-

DRUCKER: 9- und 24-Nadel-Drucker ab 198,-

LASERDRUCKER: neuwertige und gebrauchte Marken-geräte ab 1995,-

DISCOVERY MODEM: 2400 Baud, neu, (ohne FTZ, Betrieb strafbar) 378,-

ANRUFBEANTWORTER: mit Fernanfrage, neu (ohne FTZ, Betrieb strafbar) 179,-

Weitere gebrauchte und neue Geräte sowie Software auf Anfrage!

ALPHA 2000 GmbH, 24 Std. Info: 069-443000, Fax 44 30 22, 6000 Frankfurt/M. 1, Ingolstädter Str. 27

ALPHA 2001 GmbH, 24 Std. Info: 05 61-52 50 66, 3501 Niestetal (bei Kassel), Witzenhäuser Str. 10

ALPHA 2002, DDR-1512 Werder/H., Unter den Linden 17, Tel.: 26 49

ALPHA 2003, DDR-6500 Gera, Sachsenplatz 10, Tel.: 2 83 27

ALPHA 2006, DDR-6840 Pößneck, Mühlstr. 2, Tel.: 23 66

Edgar Meyzies

Assembler ist keine Hexerei

Auf dem Weg zum Profi

Unser erster Assemblerkurs wurde von Ihnen, unseren Lesern, stark beachtet. Danke für Ihr Interesse und für Ihr engagiertes Feedback. In Zuschriften und auf der Hotline wurde angeregt, den Kurs fortzusetzen. Voilà, die erste Folge unserer Bemühungen halten Sie in den Händen.

In sechs Ausgaben wollen wir versuchen, Ihnen den Weg zum Champion in der Assemblerprogrammierung zu ebnen. Dabei werden wir auf Erfahrungen setzen, die Sie im vorherigen Kurs [1] oder anderweitig gesammelt haben. Es werden jedoch auch Grundlagen vermittelt und Themen behandelt, die uns als Fragen oder Vorschläge erreichten. Ein kleines Projekt bildet dann den Rahmen.

Zielrichtung

Mit dem Kurs verfolgen wir mehrere Ziele. Wir wollen Ihnen Freude an der Assemblerprogrammierung bereiten, die sich am ehesten bei systematischer Arbeit mit guten Werkzeugen einstellt. Ferner wollen wir Sie zum aktiven Mitmachen herausfordern, beim Entwerfen, Codieren und Testen.

Verfeinern wir nun das Ziel. Im Verlauf des Kurses soll ein File-Requester (AmigaDOS-Requester, kurz: »ADR«) mit besonderen, noch zu beschreibenden Leistungsmerkmalen entstehen. Das kleine Projekt dient als »Vehikel«, die Anwendung einfacher aber bewährter Methoden der Software-Technik praktisch zu üben, ohne dabei den immer wieder erforderlichen Ablauf von Entwickeln, Codieren und Testen zu ignorieren. Der Ablauf geht aus dem Kursfahrplan hervor, der nur wesentliche Stationen der einzelnen Ausgaben markieren kann.

Jeder Teil der Folge wird einige Grundlagen über die Arbeit mit dem Prozessor »68000« bieten. Dem Programmmentwurf wird ein hoher Stellenwert beigemessen. Programmablaufpläne werden als Basis für die Codierung eingesetzt.

Wir streben an, unseren Code erst im jeweils folgenden Heft zu veröffentlichen, um Ihnen den Spaß am Codieren und am Austesten nicht zu nehmen.

Wie im vergangenen Kurs wird es uns nicht darauf ankommen, besonders trickreich zu programmieren. Auch die Optimierung (Codelänge und Laufzeitverhalten) bleibt von untergeordneter Bedeutung. Dafür betonen wir die Aspekte Anschaulichkeit, Nachvollziehbarkeit und Änderbarkeit.

Erschrecken Sie nicht bei dem Wort Methode. Es folgt keine langatmige, theoretische Abhandlung über Möglichkeiten für das Vorgehen bei der Entwicklung von Software. Die Praxis und einfache Verfahren sind gefragt. Es ist jedoch wertvoll, sich vor Augen zu

führen, daß der Prozeß der Entstehung von Software prinzipiell gemäß Abbildung 1 abläuft.

Sechs Phasen im Leben eines Programms

Die ersten beiden Phasen (Schritte) dienen dazu, Klarheit über das Entwicklungsziel zu gewinnen. Sie sind bei Auftragsarbeiten von extrem hoher Bedeutung. Aber auch bei Ausübung unseres Hobbys ist es von Vorteil, sich selbst darüber im klaren zu sein, was man eigentlich erreichen will. Überflüssig, mögen Sie einwenden. Das habe ich doch alles im Kopf! Schon recht, aber ist die Realisierung so präzise durchdacht, daß sie es wert ist, sofort in Assemblerstatements gegossen zu werden?

Methoden

Würden Sie zum Beispiel mit dem Bau eines Gartenhauses ohne Bauplan beginnen? Um diesen zu erstellen, würden Sie Ihre Vorstellungen durch-

denken und schließlich im Plan konkretisieren. Bei der Zeichnung des Planes werden mitunter Widersprüche deutlich, oder Sie werden angeregt, nochmals über Ihre bisherigen Ideen nachzudenken.

Software < > Gartenhütte?

Gibt es einen vernünftigen Grund, die Entstehung von Software anders zu behandeln als den Bau einer Gartenhütte? Etwa weil Software im Gegensatz zu Balken unsichtbar ist und ein »Verschnitt« dadurch nicht augenfällig wird? Wir meinen nein und treten deshalb vehement für den Gedanken ein, uns Zeit für eine Analyse der Hobby-Programmierung zu nehmen. Der methodische Schritt kann wesentlich dazu beitragen, die Komplexität des Vorhabens richtig einzuschätzen, den Weg für eine sichere Realisierung zu beschreiten und Flickwerk mit den damit verbundenen Folgen zu vermeiden.

Haben Sie in Abbildung 1 die Rückkopplung zwischen den einzelnen Phasen bemerkt? Damit soll zum Ausdruck gebracht werden, daß auch bei einer Annäherung der oberen und unteren Bereiche Rückwirkungen auftreten können. Schließlich ist es unvermeidbar, daß sich Vorstellungen über das Entwicklungsziel verändern, wenn man sich damit auseinandersetzt. Weiterhin können unvorhergesehene technische Schwierigkeiten auftreten. Je intensiver einzelne Schritte durchdacht werden und je besser man seinen Amiga kennt, desto unwahrscheinlicher wird der »Rückfall«.

Programmmentwurf

Der dritte Schritt im Entstehungsprozeß von Software wurde mit Programmmentwurf bezeichnet. Bei komplexen Projekten ist es zweckmäßig, die Phase in Systementwurf und in Programmmentwurf aufzubrechen. Der Systementwurf erfaßt im wesentlichen

Kursfahrplan

Teil 1: Einführung und Grundlagen

Teil 2: Benutzeroberfläche entwickeln

Teil 3: Steuerung realisieren

Teil 4: Dynamische Datenstrukturen

Teil 5: Requester mit Funktionen

Teil 6: Programmschnittstelle schaffen

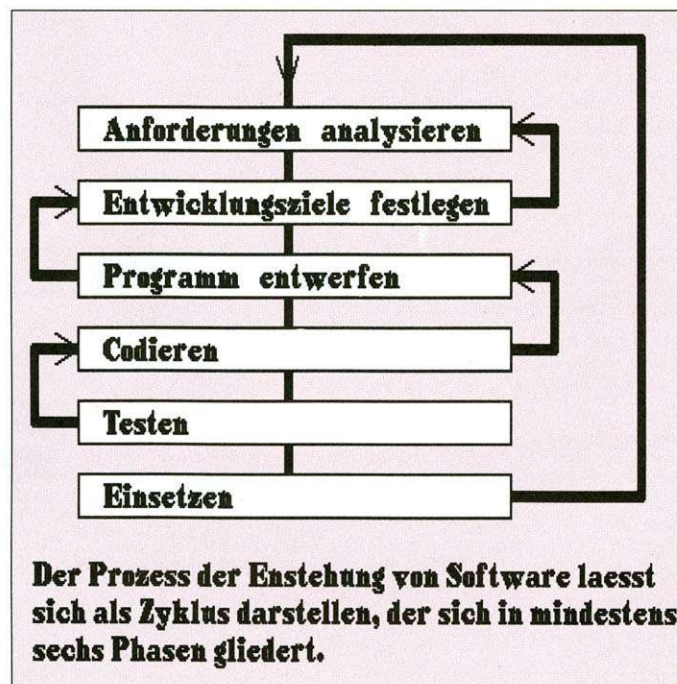


Abb. 1. Die Entstehung eines Programms

das Zusammenspiel von Programmen, um die angestrebte Funktionalität zu erreichen. Der Programmwurf selbst beschreibt die Struktur und den Ablauf eines Programms so, daß der Bauplan für die Codierung entsteht. In dem Plan sind natürlich auch Reaktionen auf Ein- und Ausgänge zu berücksichtigen.

Schrittweise Verfeinerung

Wie beim Hausbau kann es Gesamtpläne und Feinentwürfe geben. Für die Realisierung größerer Projekte wird empfohlen, die Möglichkeiten der "schrittweisen Verfeinerung" einzusetzen, da sich aus dem Vorgehen unmittelbar die Programmstruktur ergibt.

Wir wollen Ihre Freude an der Assemblerprogrammierung auf jeden Fall fördern und unter keinen Umständen durch Formalismus ersticken. Deshalb fordern wir auch nicht, daß die Analyse- und die Entwurfsphase immer ausdrücklich zu durchlaufen sind. Beim versuchsweisen Austeilen einer Systemroutine zum Beispiel wären die Schritte doch wohl überzogen. Die Entscheidung, von welcher Komplexität ab die Takte explizit zu durchlaufen sind, muß jeder selbst fällen. Die Praxis hat jedoch gezeigt, daß sich der Programmwurf schon bei kleinsten Projekten lohnt und den Zeitbedarf für

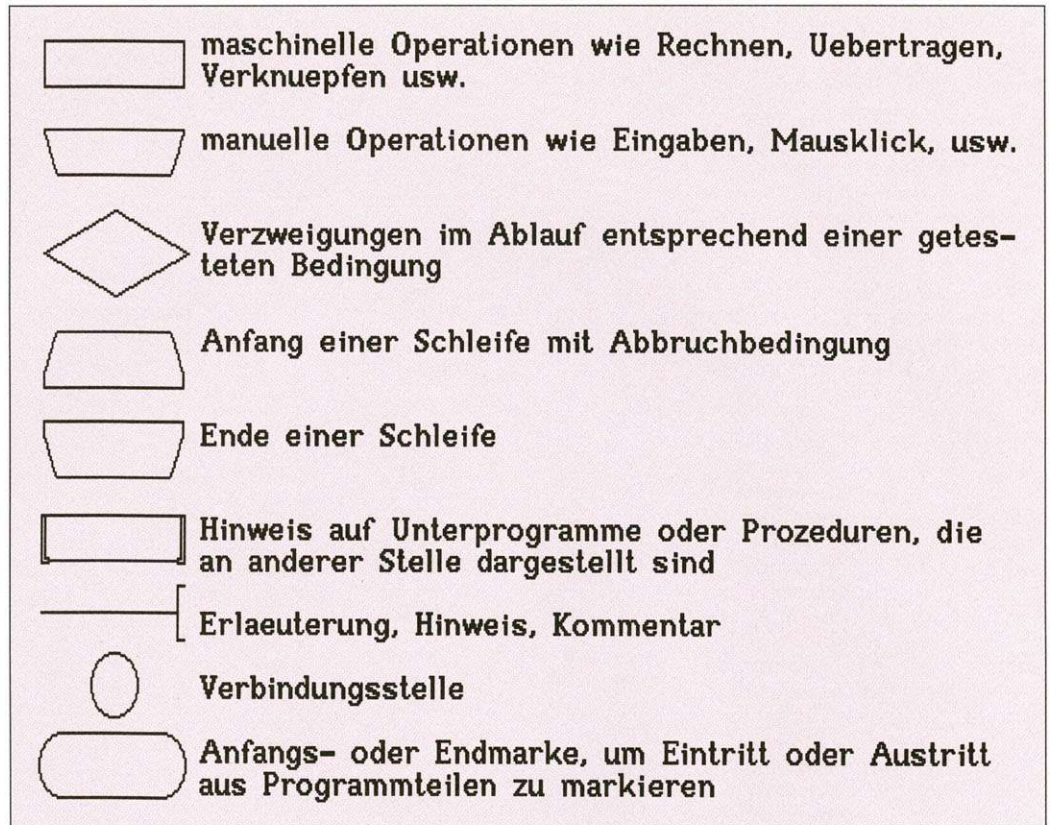


Abb. 2. Symbole für den Ablaufplan

die Fehlersuche reduziert. Auf eine einfache Methode, Programme zu entwerfen, gehen wir im Abschnitt "Programmabläufe planen" ausführlich ein. Neben der Scheu, unnötig erscheinenden Aufwand zu be-

treiben, kann es aber auch Defizite im Verständnis von Systemroutinen geben, die mitunter davon abhalten, einen Programmwurf zu erstellen. Wie soll man etwas ausdrücken, wenn man noch nicht weiß, wie es geht, wenn man während der Programmierung noch lernen muß? Genau hier greift die Methode der schrittweisen Verfeinerung. Detaillieren Sie Ihren Plan dann nur bis auf die Ebene dessen, was zu tun ist. Vertragen Sie das Wie auf später, wenn Sie sich einen Weg für das Vorgehen erarbeitet haben. Führen Sie aber Notizen über Ihre Experimente, um sich nicht im Kreis zu drehen. So können Sie sich Enttäuschungen ersparen und Ihre Erfahrungen für andere Projekte festhalten. Eventuell ergibt sich daraus sogar ein Beitrag für unser Magazin. Wir würden uns darüber freuen.

Ein bißchen Ironie

Im Phasenmodell der Entstehung von Software gelangen wir nun zur Codierung, einer sehr beliebten Beschäftigung, die das Image prägt und in den Augen anderer mit dem "touch" des Künstlerischen oder gar des Genialen verbunden ist. (Selbst-)Ironie, laß nach! Haben Sie sich eigentlich schon gefragt, warum

dieser Schritt schlicht mit Codierung bezeichnet wurde und nicht mit Programmierung? Wir haben die Bezeichnung bewußt gewählt, um zu verdeutlichen, daß die kreative Programmierung sehr viel mehr umfaßt als (gegenüber anderen zu verschweigende) stundenlange Selbstkasteiung am Terminal und die "Überstrapazierung" von Assembler sowie Linker. Sie ahnen es schon, daß der Autor die Schritte vor der Codierung der Programmierung zurechnet und ihnen eine hohe Bedeutung beimißt.

Systematik und Spiel

Es wird einfach die Behauptung aufgestellt, daß ein schrittweises, systematisches Vorgehen zu besseren Lösungen, auch aus dem Blickwinkel der Dokumentation, führt als das häufig anzutreffende Vorgehen von Praktikern, erst einmal anzufangen, um sich allmählich zum zunächst noch unscharfen Ziel zu hangeln. Die Versuchung ist groß. Es wird nicht verkannt, daß dieser Weg reizvoll ist und eher spielerisch zu Lösungen führt, die durchaus tragfähig sein können. Beim Spiel ist der Zeitaufwand von untergeordneter Bedeutung. Es ist aber auch ein schönes Gefühl,

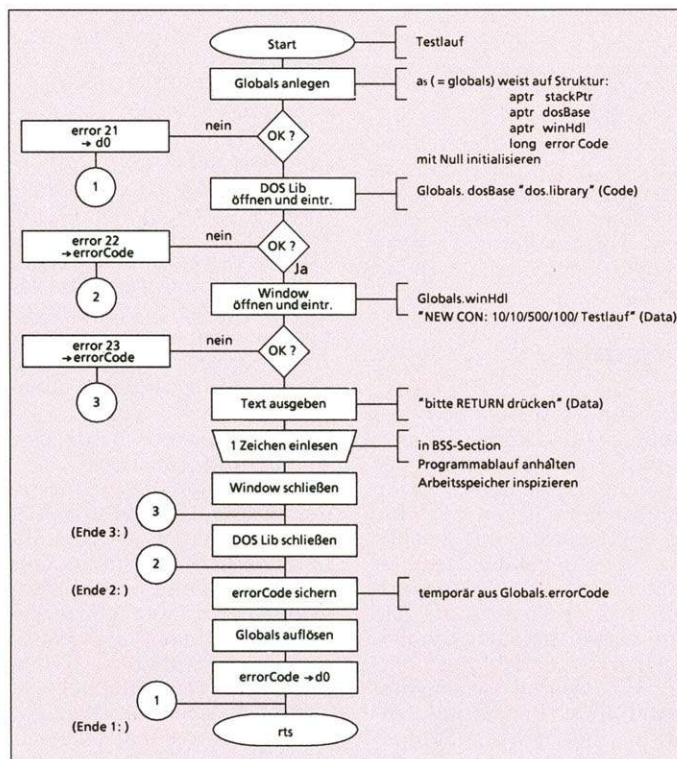


Abb. 3. Der Ablaufplan für Testlauf 1

die Früchte systematischer Arbeit zu ernten.

Testaufwand verringern

Die vorletzte Phase im Lebenszyklus (Entstehung und Nutzung) von Software dient dem Test. Es ist schwer, selbst erstellte Software angemessen zu testen. Die Fehler finden meistens andere viel leichter und schneller. Um dennoch zu brauchbaren Ergebnissen beim Test eigener Werke zu gelangen, bedarf es einer guten Vorbereitung. Ein detaillierter Programmwurf kann bei entsprechender Codierung den Testaufwand verringern. Aus dem Entwurf ergeben sich die Testfälle im kleinen und aus den Ergebnissen der Analysephase die im großen. Es ist sehr nützlich, bereits in der zweiten und dritten Stufe des Prozesses darüber zu befinden, wie das Programm in Extremfällen reagieren soll, zum Beispiel bei fehlerhaften Eingaben oder bei Speicher-mangel. So vermeidet man, daß Mängel erst beim Testen offensichtlich werden und man erneut in die Codierphase eintreten muß. Es hat sich auch bewährt, bereits in der Analyse- und Entwurfsphase Testideen festzuhalten.

Werkzeuge

Auch für diesen Kurs bleiben wir bei bewährten Software-Werkzeugen aus dem PD-Bereich. Leider gibt es noch kein "Mini-CASE" (Computer Aided Software Engineering), das die Analyse- und die Entwurfsphase hinreichend unterstützt. An den Editor stellen wir keine besonderen Ansprüche.

Makroassembler »A68K«

Als Makroassembler findet der in unserem ersten Kurs bewährte »A68K« [2] Verwendung, der gut mit dem vorgesehenen Linker und Debugger harmonisiert. Auf den Assembler sind wir in [3] ausführlich eingegangen. Drucken Sie bei Bedarf die Anleitung »A68K.doc« aus, um Anschluß zu gewinnen.

Linker »BLINK«

Auch in diesem Kurs setzen wir den PD-Linker »BLINK« in der Version 6.7 [4] ein. Der Linker arbeitet so problemlos, daß das Fehlen einer ausführlichen Dokumentation unkräftig ist. Eine Beschreibung der Optionen findet man in [5].

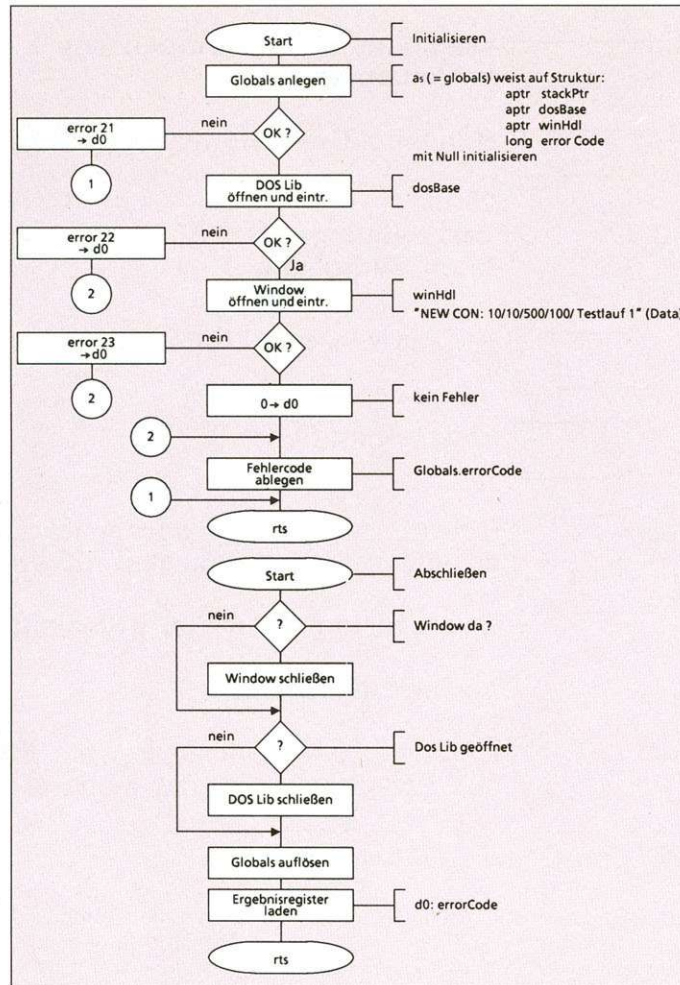


Abb. 4. Der strukturierte Ablaufplan

Symbolischer Debugger

»DEBUG«

Neu in unserem Werkzeugkasten ist der symbolische Debugger »DEBUG« (Version 2.10) [6]. Der Debugger arbeitet nur mit der 68000er CPU. Der Assembler kann mit der Option »-d« veranlaßt werden, Symbol-Hunks (Code-Blöcke mit den von uns im Programm verwendeten Namen und die so bezeichneten Stellen im Arbeitsspeicher) zu erzeugen. »DEBUG« nutzt diese Informationen. Die Länge der lauffertigen Programme nimmt gewaltig zu. Mit Unterstützung des Debuggers ist es möglich, ein Programm schrittweise ablaufen zu lassen, zum Beispiel bis zu einem angegebenen Label. Auch Breakpoints können gesetzt werden. Der Debugger ist in einer Textdatei ausführlich in englischer Sprache dokumentiert. In Tabelle 2 geben wir einen Überblick über die vielseitigen Funktionen des Debuggers, um Ihnen das Werkzeug schmackhaft zu machen. Sie werden sich schnell einarbeiten.

Include-Dateien

Nur bedingt den Werkzeugen zuzurechnen sind die Include-Dateien, die im Aufsteigerkurs heranwachsen und die wir weiterverwenden sowie ausbauen wollen. Sie finden die Dateien in der **Data-box** zu dieser Ausgabe. Aus Platzgründen müssen wir leider von einem Abdruck absehen. Heute allerdings kommen wir noch ohne die Includes aus.

Planung

Aus den verschiedenen Methoden, Programme zu entwerfen beziehungsweise das Ergebnis dieser Bemühungen darzustellen, haben wir eine der einfachsten ausgewählt, den Programmablaufplan, der zum Beispiel in [7] ausführlich beschrieben wird. Die von uns verwendeten Symbole gehen aus Abbildung 2 hervor. Wir werden sie entsprechend ihrer Bedeutung einsetzen und uns die Freiheit nehmen, mit unterschiedlicher Detaillierung zu arbei-

ten. Die Ablaufpläne werden wir durch Beschreibungen der verwendeten Datenstrukturen ergänzen.

Modell

Brennen Sie schon darauf, endlich loszulegen, die Pflichtübung von Analyse und Entwurf hinter sich zu bringen, um schließlich in die Tasten des Amiga zu greifen? Vorher ist jedoch die gemeinsame Basis ein wenig zu beschreiben. Außerdem haben sich für den Amiga Konventionen etabliert, die zwingend zu beachten sind.

Registergebrauch

Die Inhalte der Register »a0«, »a1« sowie »d0« und »d1« werden von Routinen des Betriebssystems verwendet und nicht gesichert. Es ist deshalb davon auszugehen, daß ihr Inhalt zerstört wird. In eigenen Routinen sollte diese Konvention aufgegriffen werden. Das bedeutet konkret, daß bei anderen Registern als den genannten, routinemäßig zu sichern und ihre Inhalte wiederherzustellen ("restaurieren" hat sich dafür eingebürgert) sind, sofern sie verwendet werden.

Die Befolgung der Konvention gibt uns die Sicherheit, daß wir alle Register beliebig einsetzen können (um zum Beispiel auf häufig benutzte Speicherstellen schnell zuzugreifen), ohne uns über die Bewahrung von Registerinhalten sorgen zu müssen.

Die Systemroutinen liefern Ergebnisse vorwiegend in »d0« und »d1«. Listing 1 enthält Beispiele dazu. Eine vollständige Übersicht ist in [8] abgedruckt. Die Gültigkeit der Funktionsergebnisse ist explizit zu testen (zum Beispiel mit »tst« beziehungsweise »move« mit anschließender Abfrage des Statusregisters). Auch diese Konvention werden wir für eigene Programme übernehmen.

Die Routinen des Betriebssystems (und sonstiger Libraries) sind über Sprungvektoren zu erreichen, mit der Basisadresse der jeweiligen Bibliothek in »a6«. Ein anderer Register ist dafür nicht zu verwenden. Vor dem Aufruf von Systemroutinen ist die jeweilige Library explizit zu öffnen und bei Programmende zu schließen. Die dafür benötigten Routinen enthält die stets geöffnete Library »Exec« mit der festen Basisadresse vier.

Der Vollständigkeit halber soll nicht unerwähnt bleiben, daß »a7« als Stackpointer (sp) besetzt ist und uns somit eigentlich nur die Adreßregister »a0« bis »a5« frei zur Verfügung stehen.

Anfangsadressen von Datenstrukturen

Der Anfang von Datenstrukturen ist auf gerade Adressen zu legen. Die Ausrichtung erreicht man zum Beispiel durch Verwendung der Assembleranweisungen "cnp 0,2" (Programmzähler auf gerade Adresse ohne Offset bringen) oder "even". Für Datenstrukturen, die von DOS oder den Custom-Chips verwendet werden, ist sogar die Ausrichtung auf Langwortadressen ("cnp 0,4") erforderlich. Zeichenketten können auch an Byte-Adressen beginnen, sofern auf sie nur Byte für Byte zugegriffen wird.

Wir werden vorwiegend mit dynamischen Datenstrukturen arbeiten, das heißt erst während des Programmablaufes Speicherplatz für unsere Objekte anfordern. Die entsprechende Routine »Exec .AlocMem« läßt die angeforderten Speicherblöcke stets an Langwortadressen beginnen, so daß die Anforderungen der vorherigen Absätze automatisch erfüllt werden. Bei der Anforderung von Speicherblöcken werden wir diese mit »null« initialisieren lassen, um mit einem definierten Zustand zu beginnen.

Direkte Adressierung von Hardware

Trotz der erreichbaren Zeiterparnis werden wir der Versuchung widerstehen, Hardware direkt zu adressieren. Das Betriebssystem bietet hinreichende Unterstützung, auf die Hardware "multitasking-ge-

recht" zuzugreifen. Außerdem sollen unsere Programme auch noch auf dem Amiga 3000 laufen.

Stack dimensionieren

Ohne Stack läuft nichts. Jeder Prozeß hat einen eigenen Stapel, der nach unten wächst und auf dessen unteres Ende der Stackpointer (»sp«) weist. Standardmäßig legt das Betriebssystem einen Stack von 4000 Bytes an. Für unsere Programme ist das ausreichend.

Reentrant programmieren

Bei der Programmierung von Software für ein multitasking- und damit auch multiuser-fähiges System ist es eine gute Übung, "reentrant" (auch mit "shareable" oder "ablaufinvariant" bezeichnet) zu programmieren. Dadurch kann

ein Programm für mehrere Nutzer quasi gleichzeitig arbeiten. Mit "reentrant" (Versuch einer wörtlichen Übersetzung: "wiedereintrittsfähig") wird die Eigenschaft eines Programms bezeichnet, von mehreren Tasks gleichzeitig ("shareable") so genutzt werden zu können, als wenn es nur für die jeweilige Task in den Arbeitsspeicher geladen wäre. Das Programm darf somit nur auf die Objekte (Variablen) schreibend zugreifen, die der gerade ablaufenden Task zuzuordnen sind. Jede Task findet nach einem Task-Wechsel genau dieselben Bedingungen vor, die vor dem Wechsel bestanden ("reentrant"). Das Betriebssystem des Amiga bietet ein exzellentes Beispiel dafür.

Es liegt auf der Hand, daß keine statischen globalen Variablen verwendet werden dürfen. Vielmehr muß jede Anwendung über einen eigenen Satz an globalen Variablen verfügen, der in einer dynamisch verwalteten Datenstruktur anzulegen ist. In unseren Beispielen kommen wir darauf zurück.

Objekte, die nicht (schreibend) verändert werden, das heißt Konstanten, Zeichenketten oder Images, können selbstverständlich global an festen Adressen (in einem einfachen Satz) gehalten werden. Es wird nur lesend darauf zugegriffen.

Lassen Sie uns nun gemeinsam den ersten Testlauf vorbereiten. Halt, nicht gleich den Amiga einschalten, sondern

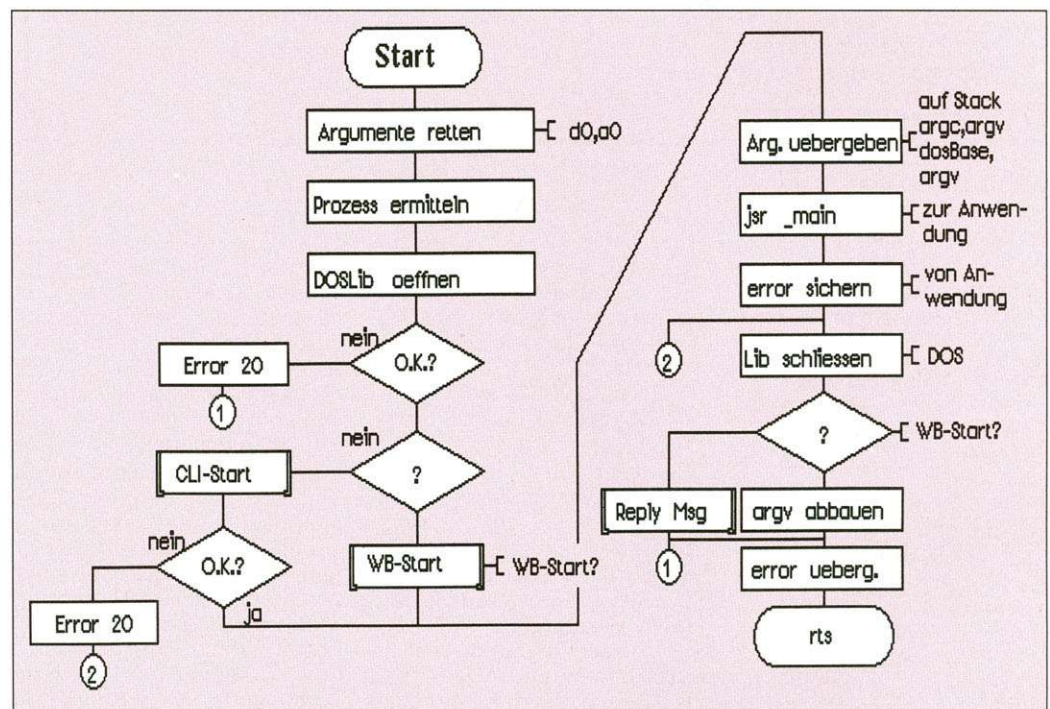


Abb. 5. Der weitere Ablaufplan

MVC

Musik Video Computer

Public-Domain-
Dschungel?
Nicht bei uns!!!

Fordern Sie unsere 4 deutschen Katalogdisketten an (DM 6,- in Briefmarken).
Alle gängigen Serien auf Lager! Natürlich topaktuell!
Und jetzt der Hammer: Jede 3,5"-Disk. nur DM 1,90
Jede 5 1/4"-Disk. nur DM 1,10
Sonderserien plus DM 0,60

Wir liefern Ihren Auftrag spätestens 1 Tag nach Eingang aus!!!
Daß wir nur mit Verify kopieren, ist für uns selbstverständlich!
Testen Sie uns doch einfach!!!

MVC

Musik Video Computer — Alles für und mit AMIGA
Hammer Str. 103 · 4730 Ahlen · Telefon/BTX 023 82/25 03
Telefax 023 82/25 04

Funkbilder für IBM-kompatible PCs XT/AT, C64 und 128,
AMIGA und Atari ST
Fernschreiben, Morse und
Radio-Kurzwellen-Nachrichten.

Haben Sie schon einmal das Piepsen von Ihrem Radio auf Ihren Bildschirm sichtbar gemacht? Hat es Sie schon immer interessiert wie man Wetterkarten, Meteosat-Bilder, Wetter-Nachrichten, Presseagenturen, Botschaftsdienste usw. auf einem Computer sichtbar macht? Ja? —, dann lassen Sie sich eine Einsteiger-Information schicken oder bestellen Sie einfach gleich.

Steckfertige Karten mit eingebautem Filterkonverter. Alle gängigen Betriebsarten, selbsterkennende Auswertung und Abstimmung. Stufenlose Shiften und Baudrate. Sondereinheit für verschlüsselte Sendungen und Codeanalyse. Drucken, Speichern, automatische Aufzeichnung. Senden und Empfangen von Funkfern schreiben, Morsezeichen und Faksimile-Bilder. Aufzeichnen, Überarbeiten, Speichern und Drucken.

Unser Angebot — eine Komplett-Ausrüstung mit Anleitung für den Einsteiger für Funkfern schreiben, Morse- und Bilder-Empfang. Modul einstecken, mit Lautsprecheranschluß verbinden, einschalten und los geht's.

Super-Sonder-Angebot BONITO-Supercom ab 298,00 DM
Bitte Info Nr. 2 anfordern bei

Bonito, Peter Walter

Gerichtsweg 3, 3102 Hermannsburg, Telefon 0 50 52/60 52

B Breakpoints setzen
C Parameter fuer Programmstart uebergeben
D Disassemblieren
F Speicher fuellen
G Programm ausfuehren
I Register der CPU initialisieren
L Programm laden, um zu debuggen
M Speicherbereich in Langworten anzeigen
N Basis fuer Zahlendarstellungen bestimmen
R CPU-Register anzeigen
S Liste aller Symbole anzeigen
T Trace-Modus einschalten
W Ausgabefenster ein-/ausschalten
X Debugger verlassen
Z eine Unterroutine ueberspringen
. Speicherinhalte aendern
> Redirection
< Eingaben von Datei lesen
? Help-Text anzeigen

DEBUG

Der symbolische Debugger "DEBUG" bietet eine Reihe leistungsfaehiger Hilfen, die geeignet sind, die Programmentwicklung wirksam zu unterstuetzen.

Zu finden ist er unter anderem auf der Fish-Disk 251 in der Public Domain.

Hier finden Sie eine Uebersicht ueber die Tastaturbefehle

Abb. 6. Die Funktionen von »Debug«

zu Papier und Bleistift greifen; weniger, um zu codieren, als den Versuch zu unternehmen, die eingangs propagierten Methoden einzuüben.

Der erste Testlauf

Anforderungen

Im ersten Schritt soll, Sie erinnern sich, die Anforderung an das Programm erfaßt werden. Was halten Sie von folgender Aufgabenstellung: Ein einfaches Programm ist zu erstellen, das

- eine oder mehrere Systemroutinen aufruft,
- im Ablauf innehält, um den Arbeitsspeicher zu inspizieren, und
- Möglichkeiten bietet, den symbolischen Debugger zu testen.

Entwicklungsziele

Die Aufstellung der Anforderung wirkt wie ein Selbstzweck. Das Prinzip wird jedoch deutlich. Aus dem Ergebnis der Anforderungsanalyse, der obigen Aufstellung, sind die Entwicklungsziele abzuleiten:

Ein CLI-Programm entwickeln, das

- ein Dos-Window öffnet,
- darin einen Text ausgibt,
- auf Tastendruck das Fenster schließt,
- reentrant ist und
- mit Labels versehen einen Symbol-Hunk erzeugt.

Die Aufstellung der Entwicklungsziele ist nichts anderes als die Konkretisierung der Anforderungen. Wenn Nutzer und Entwickler dieselbe Person sind, dann verschmelzen zumeist beide Phasen, und man begnügt sich mit der Formulierung der Entwicklungsziele.

Bei der Erarbeitung der Ziele könnten zum Beispiel folgende erste Testideen aufgekommen sein:

- Wird der angeforderte Arbeitsspeicher "sauber" zurückgegeben?
- Ist das Programm tatsächlich reentrant?
- Werden für die möglichen Sektionsarten (»CODE«, »DATA«, »BSS«) jeweils eigene Hunks (siehe dazu [9]) angelegt?
- Ist es möglich, einen »DATA-Hunk« in das CHIP-Memory laden zu lassen?

Auf jeden Fall sollte man die Testideen aufschreiben oder in einer Datei festhalten. Weitere Testvorstellungen ergeben sich aus dem...

Programmmentwurf

Bei unserem einfachen Beispiel wirkt auch der formale Akt des Programmmentwurfs überflüssig. Aber wo ist die Grenze zu ziehen? Von welchem Komplexitätsgrad ab lohnt sich der Overhead? Wie bereits herausgestellt, muß jeder für sich selbst die Frage

beantworten. Bedenken Sie bitte bei Ihrer Entscheidung, daß der Entwurf die Strukturierung beziehungsweise Modularisierung fördern kann. Hinzu kommen der Gewinn für die Dokumentation und für das Testkonzept sowie eine Reduzierung des Zeitbedarfs für das Debugging.

Was halten Sie davon, wenn Sie nun selbst einen Programmablaufplan zur Erreichung der oben beschriebenen Entwicklungsziele mit den Symbolen aus Abbildung 2 entwerfen? Unsere Vorstellungen finden Sie in Abbildung 3. Wer von uns hat etwas vergessen? Ist Abbildung 3 detaillierter als Ihr Plan, oder ist es gerade umgekehrt? Es ist ei-

gentlich nicht möglich, feste Regeln für den Detaillierungsgrad aufzustellen.

Um die Übersichtlichkeit zu bewahren, aber auch um die Komplexität zu reduzieren, ist es häufig von Vorteil, mit dem Symbol für Unterprogramme beziehungsweise Prozeduren (vergleiche Abbildung 2) zu arbeiten. Ein Programm erhält dadurch Struktur, und man vermeidet es, sich zu früh an Details festzubeißen. Damit wollen wir Sie überreden, für das oben vorgegebene Entwicklungsziel einen zweiten Ablaufplan zu erstellen, der zwei oder drei Unterprogramme aufweist.

Zweite Runde

Unseren zweiten Ablaufplan finden Sie in Abbildung 4. Sind Sie zu einem ähnlichen Ergebnis gekommen? Können Sie die Vorteile des zweiten Vorgehens erkennen? Gibt es auch Nachteile, und wenn ja, welche? Eine Struktur gemäß Abbildung 4 werden wir für alle künftigen Ablaufpläne verwenden. Sie war im Prinzip auch schon in unserem Aufsteigerkurs anzutreffen. Haben Sie bemerkt, daß Assembler bisher noch keine wesentliche Rolle gespielt hat? Wir sind absichtlich so vorgegangen und meinen, daß Programmmentwürfe in weiten Teilen sprachunabhängig gehalten werden können. Viele Listings wären viel leichter in eine andere Sprache zu übertragen, wenn ein Programmmentwurf beigegeben wäre. Hiermit wollen wir den ersten Teil unseres Kurses beenden. Sie finden im nächsten Teil die Listings, mit denen wir im weiteren Verlauf arbeiten wollen. Dort wenden wir uns dann auch der Codierung zu.

(jb)

Literatur und Software

- [1] AMIGA DOS 1/90 bis 5/90, sämtliche Ausgaben noch lieferbar
- [2] Kick PD 266 (Vers 2.61) oder Fred Fish 168 (Vers 2.42)
- [3] AMIGA DOS 1/90
- [4] Kick PD 266 oder Fred Fish 110
- [5] Das Amiga Guru-Buch, 1989, S. 146 ff.
- [6] Fred Fish 251
- [7] Software Engineering kurz und bündig, 1986
- [8] AMIGA DOS 1/90, S. 108 ff.
- [9] Das Amiga Guru-Buch, 1989, S. 466 ff.
- [10] AMIGA, Programmieren in Maschinensprache, 1989
- [11] M68000-Familie, Teil 1, 1985
- [12] Das Amiga Guru-Buch, 1989, S. 379 ff.
- [13] Amiga Intern Band 2, 1989
- [14] AMIGA ROM KERNEL, Libraries & Devices, 1990

Liebe Leser,
in dieser Rubrik finden Sie neben Informationen zu den in der AMIGA DOS vorgestellten Programmen und Produkten Rat und Hilfestellung zu Ihren kleineren und größeren Programmierproblemen. Wir sind natürlich jederzeit bemüht, die eingehenden Leserfragen zu beantworten. Doch haben Sie bitte Verständnis, daß wir nicht alle eingehenden Briefe persönlich beantworten können. Oft erreichen uns mehrere Briefe zum gleichen Thema, einer davon wird dann stellvertretend für die anderen in unserer Zeitschrift beantwortet.

Dringende Probleme, die das Heft betreffen, lassen sich möglicherweise besser telefonisch regeln. Rufen Sie dienstags unsere Hotline an. Von 17 bis 20 Uhr stehen wir Ihnen mit Rat und Tat zur Seite, wenn Sie eine der folgenden Nummern wählen:
0 56 51/8 09-7 40 (bis 744)

Ihre AMIGA-DOS-Redaktion



SIE FRAGEN, WIR ANTWORTEN

In unserer Leserbriefecke veröffentlichen wir Anfragen, die für die Leser der AMIGA DOS interessant sind. Falls Sie uns schreiben wollen, tun Sie dies an die folgende Adresse:

DMV-Verlag
Redaktion AMIGA DOS
Leserservice
Postfach 250
3440 Eschwege

Wenn Sie eine Antwort zur Problemecke haben, schreiben Sie dies unter dem Kennwort 'Problemecke'. Eine Bitte haben wir noch: Bitte schreiben Sie Ihre vollständige Adresse nicht nur auf den Umschlag, sondern auch in den Brief. Legen Sie am besten einen an Sie selbst adressierten, ausreichend frankierten Rückumschlag bei. Nun aber zu den Briefen.

Kommt er, oder kommt er nicht?

Ich, 15 Jahre alt, habe mir zu dem in der AMIGA DOS 6/90 abgedruckten Brief über den »Sleeper Virus« Gedanken gemacht. Meiner Meinung nach ist er nur Wichtigtuerei, da folgende Dinge auf irgendeine Art Probleme bereiten müßten:

1) Angeblich »cruncht« dieser Virus sechs Tracks. Zum Lesen der Diskette müßte sie der Virus allerdings wieder entcrunchen. Sollte der Virus zu-

fällig nicht aktiviert sein, könnte der Amiga diese Tracks nicht lesen. Da ein Programm nicht einfach durch Einlegen einer Diskette geladen wird (außer beim Booten), dürfte man häufiger ohne den Virus arbeiten (es sei denn, alle Disketten wären verseucht), und dann dürfte der Amiga Probleme beim Lesen haben – das müßte eigentlich auffallen.

2) Die angeblichen »Programmierer« des »Sleeper-Virus« behaupten, daß der Virus irgendwann im Dezember zuschlägt. Wie soll denn der Virus an die korrekte Zeit kommen, wenn der Amiga keine akkugepufferte Uhr hat. Entweder wäre der Virus bei den Geräten schon aktiv geworden, oder er ruht vergessen irgendwo auf Diskette.

Obwohl von diesem »Virus« eigentlich wohl nie Gefahr ausgehen wird (sicher bin ich mir allerdings auch nicht...), sollte man trotzdem beim Austausch von Disketten vorsichtig sein.

Florian Severin
Lindlar

Wie schon in Heft 6 gesagt, wenn's ein Scherz war, ist es gut, wenn es ernst gemeint war, soll die Erzeuger des Mistvieches der Blitz treffen. Viren können einem den Spaß am Computern verderben, vor allen Dingen Anfängern, die sich nicht ausreichend schützen können. Deshalb hier

nochmal der gute Rat: Eine PD-Diskette mit Anti-Kleinvieh-Programmen kostet nicht viel, zahlt sich aber aus.

(Red.)

Berichtigungen

Seit der Startausgabe 01/90 bin ich regelmäßiger Leser der Zeitschrift AMIGA DOS. Als Umsteiger vom Amstrad CPC habe ich zuvor bereits Ihre »PC International« gelesen. Normalerweise schreibe ich auch keine Leserbriefe, aber in Ihrer Ausgabe 07/90 waren eine Reihe von Fehlern, beziehungsweise falsche Auskünfte zu den Leserbriefen, so daß ich Ihnen ein paar Berichtigungen zukommen lasse.

1.) Leserbriefe:

a) Amiga 500 als PC: Eine interne Steckkarte für den A500 wird bereits seit einiger Zeit vertrieben, und zwar bei den Firmen SCS Schomburg, Bahnhofstr. 38, 2800 Bremen, und Roßmüller Computertechnik, Neuer Markt 21, 5309 Meckenheim, von denen auch bald eine externe PC-Karte zu erwarten ist.

b) Assembler: Neben deutschen Anleitungen in diversen anderen Computermagazinen findet man die Anleitung zum A68K auch im »Dritten großen Amiga Public Domain Buch« vom Verlag »technicSupport«, das für 49,- DM erhältlich ist.

2.) Textverarbeitungen: Ein kurzer Widerspruch in Text und Kurzreferenz: »Vizawrite 2.0« ist entgegen der Darstellung in der Kurzreferenz in der Lage, Bilder einzulesen.

3.) Public-Domain-Spieleshows
a) »Zerg 1.0«: Dieses Programm ist wirklich so gut, wie Sie es darstellen, nur leider... nicht fertig! Das Formulieren von Zaubersprüchen und einige andere Sachen funktionieren nämlich noch nicht, so daß das Spiel nicht bis zu Ende gespielt werden kann, man darf also auf die endgültige Fassung gespannt sein.

b) »Dungeon Flipper«: Böse Falle! Hat sich der Oase-Herausgeber schon bei Euch gemeldet? Dieses Programm gehört nicht zur Public Domain. Die Oase-PD-Reihe geht bis zur Nummer 99, ab der Nummer 100 handelt es sich um professionelle Software mit Preisen zwischen 20,- DM und 59,- DM. »Dungeon Flipper« kostet zum Beispiel 29,- DM und kann auch nur über die Oase-Reihe bei Wolf, Deipe Steige 187, 4420 Coesfeld, bezogen werden.

Ich habe Verständnis, daß ein Autor, Redakteur oder wie auch immer wegen des Schreibens kaum dazu kommt, andere Computerhefte zu lesen oder »abzukupfern«, allerdings sollte man (frau) erwarten, fachgerechte Auskünfte zu erhalten. Euer doch eigentlich recht positives Computer-

Die Problemecke

Manche Probleme lassen sich selbst in der Redaktion nicht zufriedenstellend lösen. Zu diesem Zweck haben wir die Problemecke eingeführt, in der Leser spezielle Fragen an andere Leser richten können. Wer eine Antwort auf die Fragen hat, schickt diese bitte unter dem Stichwort "Problemecke" an die Redaktion (Anschrift siehe Einleitung).

Technische Fragen

Da durch die Festplatte der zweite Einschub für das 3,5-Zoll-Laufwerk belegt war, habe ich mir ein nachträgliches externes 3,5-Zoll-Laufwerk gekauft. Da der A2000 aber intern für zwei Laufwerke ausgelegt ist, läßt sich das externe nur unter dem Namen »DF2:« ansprechen. Das ist weiter nicht tragisch, doch leider überprüft ein Kopierprogramm, das ich besitze, erst Laufwerk »DF1:«, bevor es andere Laufwerke zuläßt. Aus diesem Grund bringt auch weder ein Eintrag in der Mountlist noch ein Anmelden mit dem Mount-Befehl eine Verbesserung. Gibt es eine Möglichkeit, hardware-mäßig im Rechner das erste externe Laufwerk beim A2000 als »DF1:« anzumelden? Auch vor Lötarbeiten würde ich dabei nicht zurückschrecken.

Meine zweite Frage bezieht sich auf den A500. Bei der Kickstart 1.2 habe ich mit gutem Erfolg die resetfeste RAM-Disk »VD0:« aus der PD verwendet. Als der

A2000 mit der Kickstart 1.3 kam, wollte ich die dort enthaltene neue RAM-Disk »RAD:« sofort nutzen, stellte jedoch fest, daß nach einem Reset der Inhalt gelöscht wurde. Das gleiche passierte jetzt aber auch mit der »VD0:«, obwohl beide ordnungsgemäß angemeldet wurden. Auch der Tip, der dem A2000 beigelegt wurde, nämlich die »RAD:« mit der Option »r« anzuwählen, funktionierte nicht. Lediglich die resetfeste Hardcopy-Funktion unter »TurboPrintII« klappt. Woran kann das liegen?

Die dritte Frage bezieht sich auf meine Rechneruhr; sie funktioniert nicht mehr richtig. Nach dem "Hochfahren" der Festplatte wird die Uhr von der Echtzeituhr gesetzt, nach zirka einer Minute allerdings springt der Zeiger wieder zurück, und alles beginnt von vorne. Nach jedem Setzen mit »Setclock opt load« wird sie zwar wieder initialisiert, läuft aber dann wieder nur eine Minute. Auch hier wäre ich für einen Tip dankbar.

Matthias Röhm
Iserlohn

Sidecar macht Probleme

Ich bin Besitzer eines Amiga 500 mit zweitem Laufwerk, einem MByte Speicherplatz und dem 1060er Sidecar mit 512 kByte. Nun möchte ich das Arbeiten mit diesem System noch effektiver gestalten, daher habe ich einige Fragen an Experten:

1) Kann ich mein zweites Laufwerk (3,5") vom Sidecar aus

nutzen, wenn es am Amiga angeschlossen bleibt?

2) Wie kann ich das interne Amiga-Laufwerk vom Sidecar aus ansprechen, und in welchem Format arbeitet es dann?

3) Ist es möglich, den Speicher des Sidecars auf 640 kByte zu erweitern, ohne eine Steckkarte zu kaufen (mit anderen RAMs)?

4) Kann ich vom Sidecar aus den meist ungenutzten Amiga-Speicher als RAM-Disk benutzen?

5) Kann man vom Amiga aus das 5,25-Zoll-Laufwerk des Sidecars ansteuern?

6) Welche Festplatte empfehlen Sie mir für Amiga und Sidecar (Amiga- und PC-Betrieb)?

7) Kann ich die Daten im Sidecar-Speicher im Falle eines Amiga-Absturzes retten?

8) Kann ich meinen Monitor (Commodore 1901, umgebaut für Amiga-Scart-Anschluß) mit PC-Grafikkarten (CGA, EGA, VGA) verwenden?

9) Wie kompatibel ist das Sidecar zum IBM-XT?

Oliver Gottschalch
Krefeld-Fischeln

Zu den Fragen 6 und 8 folgendes:

Festplatten lassen sich durch Anlegen von Partitionen ("Stücken") von verschiedenen Systemen her nutzen, so die meisten Festplatten und Filecards für den Amiga. Ob dies allerdings auch für das Sidecar zutrifft, können wir mangels Besitz eines solchen nicht sagen. Der Monitor 1901 kann nur den CGA-Modus des PC verwalten, bei EGA und VGA-Karten brauchen Sie andere Monitore. Bei den restlichen Fragen sind wir auf die

Mithilfe unserer Leser angewiesen.

(Red.)

Frage zu Assembler

Ich habe eine Frage zu einem Problem, das während des Assemblerkurses auftaucht ist: Wie bekomme ich den String "DOS" als erstes in ein Programm, damit ich ihn als Bootblock verwenden kann? Der »A68K« macht – sogar wenn meine erste Zeile im Programm zum Beispiel »dc.b "DOS",0« heißt – immer irgendwelche Zeichen an den Anfang des Programms.

Florian Severin
Lindlar

Drucker LC 24-10 und Textomat

Seit sechs Monaten besitze ich einen Amiga 500 und einen Drucker (STAR LC 24-10).

Zur Grundausstattung des Amiga gehörte unter anderem das Textprogramm »Textomat« von Data Becker.

Mein Problem liegt in der Erstellung des Druckformats.

Ich benutze meistens DIN-A4-Papier und habe Schwierigkeiten mit dem Seitenvorschub.

Schon bei der ersten Seite führt der Drucker einen Vorschub von acht bis zehn Zeilen aus, obwohl ich bei der Formateinstellung den oberen Rand auf »1« eingestellt habe. Bei der zweiten Seite kommt dann schon ein Vorschub bis zur nächsten Seite. Was mache ich falsch?

Horst Hoppe,
Witten

heft läuft sonst Gefahr, das Niveau eines bekannten Boulevardblattes zu erlangen und die ernsthaften Leser zu verlieren, oder sollte es sich etwa um das Suchspiel "Wer findet die meisten Bugs?" gehandelt haben? Habe ich damit vielleicht gewonnen? Vielleicht einen "Ehrenplatz" im Redaktionsteam?

Zum Schluß eine kleine Anregung: Warum gibt Ihr nicht eine PD-Serie mit den in Eurem Heft erschienenen PD-Programmen heraus? Dadurch würde man Schwierigkeiten beim Bezug der vorgestellten PDs vermeiden und entspre-

chenden Leserfragen zuvorkommen. So ein Fehler wie mit »Dungeon Flipper« dürfte dann der Vergangenheit angehören.

So, damit beende ich den überlangen Brief, bessert Euch!

Ralf Ludwig
Lehre

Recht hat er, der Ralf Ludwig, zumindestens in einigen Dingen. Die Beschreibung des A68K ist inzwischen auch auf PD-Disketten erhältlich und die unterschiedliche Auffassung über das Können des Programms »Vizawrite« war

auch nicht nötig. Bei den PC-Karten für den A500 mußten wir jedoch vom aktuellen Stand ausgehen, und der ist immer zwei Monate vor dem Erstverkaufstag.

»Dungeon Flipper« ist übrigens tatsächlich nicht Public Domain, wir entschuldigen uns dafür beim PD-Versand Wolf und unseren Lesern. So, der Buße ist genug getan, wir geloben Besserung.

(Red.)

Kein Klick mit »NoClick«

Zu den Leserbriefen "Fragenkatalog" und "Zuviel Pink im

Druck" aus der Ausgabe 07/90 möchte ich folgendes bemerken:

Es gibt ein PD-Programm namens "NoClick" von der Fish-Disk Nr. 246 oder 241, welches das lästige Klicken der Laufwerke abschaltet. Diese Diskette kann unter anderem auch bei mir für 2,20 DM zuzüglich Versandkosten bezogen werden, auf Wunsch auch mit einer kurzen deutschen Anleitung zum Programm.

"Zuviel Pink im Druck" läßt sich mit Hilfe des Programms »TurboPrint Professional« beseitigen. Mit diesem Programm kann man die Farben

für den Ausdruck genau auf den Drucker abstimmen, dadurch erscheinen die Farben genau so auf dem Drucker, wie sie dargestellt werden.

Holger Dingelstädt
Wilhelmshavener Str. 30
8510 Fürth

Stellvertretend für die vielen anderen Briefe, die uns zu den Themen erreicht haben (wofür wir uns herzlich bedanken), haben wir den von Herrn Dingelstedt abgedruckt, so das sich Leser, die sich am Klicken noch stören, daran orientieren können.

(Red.)

Ein Schalter für den Amiga

Um einen Schalter in Amiga-BASIC darzustellen, kann folgendes Programm verwendet werden:

```
SUB Schalter (x%, y%, x1%, y1%,
text$) STATIC
LINE (x%+10, y%+5) -(x1%+10,
y1%+5), 2, bf
LINE (x%, y%)-(x1%, y1%), 3, bf
LINE (x%+4, y%+2)-(x1%+4,
y1%+2), 2, b
LINE (x%+8, y%+4)-(x1%+8,
y1%+4), 2, b
xadr=&WINDOW(8)+36:
yadr=&WINDOW(8)+38
t%=y1%-y%
t1%=y%+t%/2
POKEW xadr,x%+20
POKEW yadr,t1%+3
COLOR 1,3
PRINT text$
COLOR 1,0
END SUB
```

Der Aufruf erfolgt mit:

Schalter 150, 30, 470, 60,
"Hier muß Ihr Text stehen"

Achim Maugert und
Thomas Edlinger,
Trofaiach (Österreich)

Also, auf ins AmigaBASIC und ausprobiert.

(Red.)

Amiga 500 und PC

Ich besitze seit anderthalb Jahren einen Amiga 500, in dieser Zeit habe ich ihn immer weiter ausgebaut.

Ich habe jetzt ein Zweitlaufwerk, eine Speichererweiterung auf 1 MByte und einen 24-Nadel-Drucker. Da ich mich an der Gewerbeoberschule auf Informatik spezialisieren will, sehe ich mich gezwungen, meinem Amiga eine PC-Karte zu verpassen. Nun zu meiner Frage:

Wenn ich mir das »PC-Power-Board« kaufe, das für den

Amiga 500 angeboten wird, wie kann ich zusätzliche IBM-kompatible-Steckkarten anschließen, oder muß ich auf den Amiga 2000 umsteigen, der schon solche Steckplätze hat? Finanziell würde ich die Lösung mit dem Amiga 500 vorziehen.

Gernot Kramer,
Bozen (Italien)

Aus finanzieller Sicht würde sich die Anschaffung des »Power-PC-Board« lohnen, aber wo Licht ist, da ist auch Schatten. Der Schatten liegt in diesem Fall bei den Steckplätzen, die zwar kompatibel zum PC sind, aber nur Einsteckkarten mit halber Höhe zulassen. Wenn Sie mit einem PC arbeiten wollen, dann sollten Sie sich einen billigen PC zulegen oder auf einen gebrauchten Amiga 2000 umsteigen.

(Red.)

BASIC-Start, die zweite

Ich finde es ja sehr lobenswert, daß man den Einsteigern helfen will, aber es sind doch noch einige Verbesserungen vorhanden.

So würde ich zum Beispiel von der Diskette »Extras1.3« die Dateien aus »FD1.3« erst einmal in die für »AmigaBASIC« geforderten »bmap«-Dateien mit dem Programm »ConvertFD« (ExtrasD) umwandeln und diese Dateien ins Verzeichnis »Libs« auf der BASIC-Startdiskette kopieren.

1. AmigaBASIC kann ja mit den FD-Dateien sowieso nichts anfangen.

2. Die konvertierten Dateien sind viel kürzer, und man spart Platz auf der Diskette.

Wenn man wie ich nur im »CLI« arbeiten möchte, dann stören auch die vielen Dateien mit »info«, die AmigaBASIC generiert, wenn ich mein Programm abspeichere.

Dies kann ganz leicht verhindert werden, indem ich die »Icon.Library« aus dem Verzeichnis »Libs« lösche, und schon habe ich wieder mehr Platz.

Allerdings muß die »Startup-Sequence« geändert werden. Ein Beispiel:

```
»Setmap D«
»RUN AmigaBASIC«
```

Entweder »RUN« oder »Runback« verwenden, falls dies im Verzeichnis »C« vorhanden ist. So kann ich, ohne AmigaBASIC zu beenden, Amiga-

BASIC nach hinten klicken und im »CLI« weiterarbeiten. Ich würde auch noch aus dem Verzeichnis »C« der Workbench 1.3 die Datei »FF« (Fast-Fonts) in mein Verzeichnis »C« kopieren und in die »Startup-Sequence« einbinden:

```
»FF -0«
```

Dies ergibt eine schnellere Textausgabe.

Klaus-Jürgen Hampel,
Sibbesse

Es klappert

Seit Beginn dieses Jahres bin ich stolzer Besitzer eines Amiga 500 mit der Workbench 1.3 und ohne jegliche Erweiterung. Mein einziges Laufwerk wird daher sehr belastet, und es treten Probleme auf.

1. Mein Laufwerk ist im Vergleich mit anderen A500 auffällig laut, und die Disketten sind schwerer einzulegen als bei anderen. Dazu kommt, daß bei meinem Laufwerk eine Art Abdeckklappe fehlt, das heißt, die Mechanik und Elektronik der Floppy ist freisichtbar, was bei Freunden nicht der Fall ist.

2. Seit einiger Zeit gibt meine Floppy seltsame Klapper- und Schleifgeräusche von sich, die sich durch einen sanften Klaps auf das Gehäuse wieder legen, aber trotzdem immer wiederkommen.

Manuel Bock,
Eggenfelden

Entdeckungen über fehlende Teile im Amiga sind nichts Besonderes, da die Firma Commodore Änderungen ohne Ankündigung vornimmt.

Gravierend waren Änderungen von der Version 1.2 auf 1.3, die sich unter anderem auf die Laufwerke bezogen. Dies ist aber alles noch im normalen Rahmen und kein Grund zur Beunruhigung. Schleifgeräusche kommen sehr oft von minderwertigem Diskettenmaterial.

Das Klappen des Diskettenlaufwerks kann ein Schaden im Innern sein, sollte beim Fachhändler überprüft werden.

(Red.)

Geräusche, die zweite

1. In diesen Tagen habe ich das Spiel »Summer Edition« von der Firma Epyx erworben, welches allerseits als »gut« getestet wurde. Obwohl das Spiel einwandfrei läuft, treten

bei den einzelnen Ladevorgängen starke Geräusche auf, ganz anders als bei meinen anderen Disketten (Spiele wie Programme). Eine Rückfrage beim Vertreiber ergab, daß diese Geräusche vom starken Kopierschutz verursacht werden und mein Amiga keinen Schaden nehmen könne. Höchstens könne die Diskette defekt werden. Man würde sie dann umtauschen. Stimmt diese Aussage hinsichtlich der auftretenden Geräusche?

2. Ich habe von Ihnen eine Databox bekommen und möchte einzelne Programme auf eine andere Diskette kopieren. Unter anderem habe ich beim Kopieren der »Icons« und anschließend Öffnen des Fensters keine »Icons« gefunden. Können Sie mir weiterhelfen?

Björn Simon,
Rodgau

Zur Frage 1: Es gibt tatsächlich Kopierschutztechniken, die so stark sind, daß man glaubt, die Floppy würde jeden Moment auseinanderfliegen. Erfreulicherweise geht die Tendenz des Kopierschutzes in eine andere Richtung (Abfragen aus dem Handbuch und so weiter).

Zur Frage 2: Um Dateien zu kopieren, müssen Sie ins »CLI« oder in die »Shell«. Von dort können Sie mit dem Befehl »Copy« einzelne oder mehrere Dateien kopieren. Beispiel:

```
COPY DF0:Texte/test.txt TO
DF1:Texte/test.txt
```

In unserem Beispiel bedeutet dies, daß vom Laufwerk »DF0:« im Verzeichnis »Texte« die Datei »test.txt« auf das Laufwerk »DF1:« in das Verzeichnis »Texte« geschrieben wird.

Um Programme mit einem »Icon« zu versehen, müssen Sie auf der »Extras«-Diskette in das Verzeichnis »Tools« gehen und dort den »IconEd« aufrufen. Mit diesem Programm ist es möglich, die Programme oder Dateien mit einem Icon zu versehen.

(Red.)

Die Redaktion behält sich vor, Leserbriefe in gekürzter Form wiederzugeben. Sollten zu einem Problem mehrere Briefe eingehen, so wird ein Brief stellvertretend für alle anderen abgedruckt.

Oliver Heggelbacher

UND JETZT – EIN HIT TTTT TTTT TTTT

Radio-Show mit der "Jingle-Machine"

Mit gesampelten Sounds kann man zum Beispiel Musikstücke erstellen – wenn man musikalisch ist. Oder man nutzt sie für die Vertonung von eigenen Bandaufnahmen, seien es Audio- oder Videoexperimente. Fehlt nur noch das Wie. Nun, hier ist es – die Jingle-Machine.

Dieses Programm hat die sinnvolle Aufgabe, gesampelte Geräusche, Sprüche, Musikstücke oder was auch immer in den unterschiedlichsten Variationen abzuspielen, um damit sogenannte "Jingles" (bekannt vom Radio) zu erzeugen. Das Programm ist insbesondere für Amiga-Benutzer gedacht, die einen Sound-Sampler ihr eigen nennen, wobei solch ein Gerät meiner Meinung nach für jedermann erschwinglich und eine sehr lohnende Anschaffung ist.

Es lassen sich mit dem Programm zum Beispiel sehr einfach selbstgedrehte Filme vertonen, indem die benötigten Geräusche von einer Geräusch-CD gesampelt und dann mit diesem Programm an den passenden Stellen eingespielt werden.

Timingprobleme dürfte es dann wohl kaum mehr geben. Ich selbst benutze die »Machine«, wenn wir unsere vom Radio mitgeschnittenen Kassettenkonserven neu abmischen und dabei zwischen den Stücken Jingles und Effekte einstreuen. Das Besondere an dem Programm ist, daß es

auch das FAST-MEM, sofern vorhanden, benutzt und die Samples erst beim Abspielen in einen Pufferspeicher des CHIP-MEMs umkopiert.

Zum Betrieb braucht man die Arp-Library

Das Programm benutzt die ARP-Library V33+, welche beim Devpac Assembler mitgeliefert wird und auch als Public Domain erhältlich ist. Die Datei »arp.library« muß deshalb auf der Bootdiskette vorhanden sein.

Nach dem Start des Programms wird auf dem Bildschirm eine Tabelle aufgebaut, in der 48 einzelne Samples Platz haben. Ein Sample kann durch Anklicken der entsprechenden Nummer ausgewählt werden. Ist ein Tabelleneintrag noch nicht belegt, wird anstelle des Sample-Namens "not used" angezeigt. Am unteren Bildschirmrand wird der Hauptbildschirm aufgebaut.

Durch Anklicken der Box »EDIT« gelangen Sie in den Editiermodus. Hier können Sie nun einzelne IFF-Samples

laden, indem sie das dunkelgraue Feld hinter »NAME:«, wo der jeweils aktuelle Dateiname angezeigt wird, anklicken.

Das Sample darf nicht länger als 128 kByte sein, längere Dateien werden entsprechend gekürzt. Mit den beiden Schieberreglern »RATE« und »VOL« können Sie dann die Abspielgeschwindigkeit und die Lautstärke des Samples einstellen, der Loop-Schalter

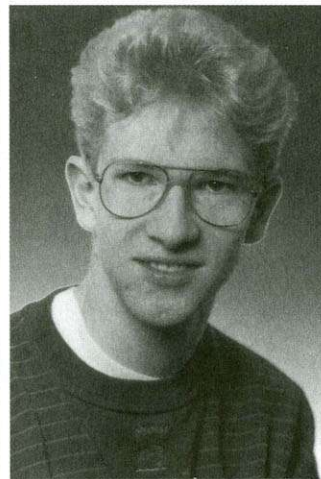
bestimmt, ob das Sample wiederholt werden soll. Am rechten Rand sind folgende Menüpunkte vorhanden:

1. »SAVE SET«
Hiermit können Sie die aktuelle Einstellung (die geladenen Samples und deren Parameter) sichern.
2. »LOAD SET«
Eine mit »SAVE SET« abgespeicherte Set-Datei wird ge-

Programm des Monats

Jingle-Machine

von Oliver Heggelbacher



Name: Heggelbacher
Vorname: Oliver
geb.: 25.05.1972
Beruf/Schule: Schüler des Gymnasiums im Bildungszentrum Markdorf
Hobbys: Elektronik und Basketball

Der Anfang des eigenständigen Programmierens begann mit einem Commodore VC20 und einem Sinclair Spectrum, die allerdings nicht mein Eigentum waren.

Der erste eigene Computer war ein Amstrad CPC 464, den ich zuerst in BASIC, danach in Assembler programmierte. Im August 1987 erwarb ich dann einen Amiga 500. Auch hier schlug ich mich zunächst mit dem BASIC herum, bevor ich dazu übergang, hauptsächlich Maschinenprogramme zu entwickeln. Ich habe mich zwar anfangs auch in C versucht, da meine C-Programme jedoch häufiger abstürzten als meine Versuche mit Assembler, kam ich schnell wieder davon ab. In unserer Schule programmiere ich außerdem auf MS-DOS-Computern, hauptsächlich in Turbo Pascal und dBase.

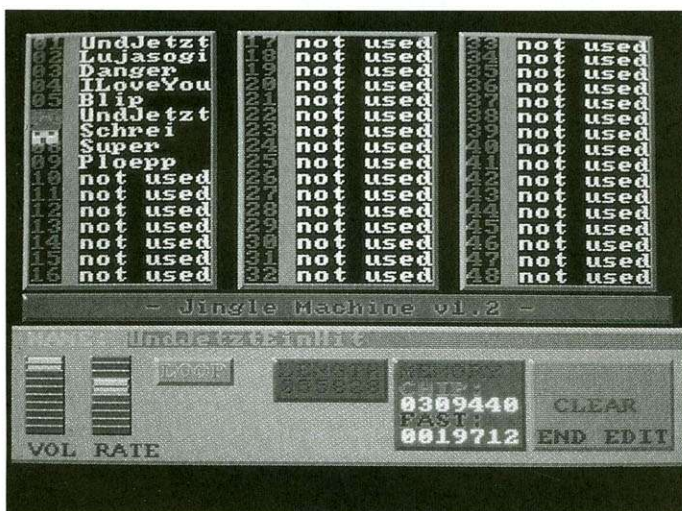


Bild 1. Die Jingle-Machine – jeder Platz ist für ein Sample gedacht

laden, und die darin definierten Sample-Dateien werden ebenfalls eingelesen.

3. »CLEAR«

Diese Funktion bewirkt das Rücksetzen der 48 Sample-Felder und Freigabe des belegten Speichers.

4. »END EDIT«

Zurück zum Hauptbildschirm.

Bei den Datei- und Speicheroperationen kann es zu folgenden Fehlermeldungen kommen:

»Cannot open file:« – Beim Laden/Speichern wurde ein ungültiger beziehungsweise nicht vorhandener Dateiname angegeben.

»Not an IFF-8SVX file:« – Die zu ladende Sample-Datei ist nicht im IFF-Format.

»No BODY chunk found:« – Die Datei enthält keine Sound-Daten.

»Not a SET file:« – Die bei »LOAD SET« angegebene Datei ist keine mit »SAVE SET« abgespeicherte Datei.

»Out of Memory:« – Es ist nicht mehr genügend freier Speicher vorhanden. Diese Meldung tritt auch auf, wenn nicht genügend zusammenhängender Speicher vorhanden ist. Abhilfe ist dann eventuell dadurch möglich, daß Sie das aktuelle Set sichern und dann wieder mit »LOAD SET« laden, da dann der gesamte Speicher neu organisiert wird.

Sowohl bei »EDIT« als auch im Hauptbildschirm kann ein Sample abgespielt werden. Dies geschieht entweder durch Tastendruck oder durch die rechte Maustaste. Wenn

Wichtig!!!

Bei der Jingle-Machine mußten wir erstmals ein wenig von unserer Maxime, nur ganze Programme abzudrucken, abweichen. Nicht, daß wir Ihnen, verehrte Leser(innen), nicht das ganze Programm gönnen, nein, es ist schlicht und einfach zu lang für eine AMIGA DOS. Teil zwei des Listings finden Sie daher erst in der nächsten Ausgabe, der 10/90.

Was machen Sie nun mit der ersten Hälfte? Ganz einfach. Nehmen Sie sich einen normalen ASCII-Editor (wer noch keinen hat, kann dazu »ED« aus dem C-Verzeichnis nehmen), und tippen Sie Teil 1 ab. Wenn Sie fertig sind, speichern Sie das File als »JingleMachine.ASC« auf einer Diskette Ihrer Wahl. Teil 2 wird dann ganz in der gewohnten Weise an den ersten Teil angehängt; also »JingleMachine.ASC« in den Editor (oder in den Assembler) laden und weiter abtippen, danach noch mal abspeichern, allerdings diesmal als »JingleMachine.S«. Dieser Quelltext kann dann vom Assembler in Maschinencode umgewandelt werden.

Ach ja, die Ungeduldigen unter Ihnen können natürlich das komplette Programm sofort auf der Databox bekommen; dort finden Sie neben dem kompletten Quelltext auch das fertig assemblierte File. Den Geduldigeren unter Ihnen empfehlen wir in der Zwischenzeit ein Getränk Ihrer Wahl und die AMIGA DOS als Entspannungslektüre.

man die Tastatur benutzt, so wählt man am besten eine der Sondertasten ([SHIFT], [CTRL], [ALT] und so weiter), da diese keine Repeat-Funktion besitzen.

Wie man Jingles abspielen kann

Im Hauptbildschirm bieten sich folgende Einstellungsmöglichkeiten:

1. Kanalanzeige

Am linken Rand befindet sich eine Kanalanzeige (»CHAN«), die den Status (Lautstärke) der einzelnen Kanäle anzeigt. Die einzelnen Kanäle können durch Anklicken der Kanalnummer ab- und angeschaltet werden. Die Effekte »MONO« und »PHASE« benutzen als

Hilfskanal allerdings auch abgeschaltete Kanäle.

2. Volume (»VL«)

Einstellung der Gesamtlautstärke.

3. Decay (»DC«)

Mit diesem Regler wird der Lautstärkeabfall beim Spielen eines Samples eingestellt.

4. Phase (»PH«)

Es werden Phasenverschiebungen erzeugt, indem der gleiche Ton auf einem anderen Kanal mit minimaler Verzögerung abgespielt wird, wobei sehr interessante Effekte entstehen. Je weiter der Regler von der Mittelstellung entfernt ist, desto größer ist die Verschiebung. Die Phasenverschiebung muß durch Anklicken der grünen Box unter

dem Regler eingeschaltet werden.

5. Lesley (»LE«)

Der Ton wandert zwischen dem linken und rechten Kanal hin und her. Mit dem Regler wird die Geschwindigkeit, mit der dies geschieht, eingestellt. Ist der Regler am unteren Anschlag, so wird das Lesley ausgeschaltet.

6. Filter (»FILT«)

Der eingebaute Tiefpaßfilter des Amiga wird ein beziehungsweise ausgeschaltet.

7. »HOLD«

Das gespielte Jingle wird beim Loslassen der Taste nicht mehr abgebrochen, sondern klingt ganz aus.

8. »MONO«

Der Klang wird sowohl auf dem linken als auch auf dem rechten Kanal gespielt, dies ist insbesondere bei der Verwendung des Lesleys sinnvoll.

9. Der Echoteil

Diese Option dient dem nachträglichen "Verhallen" der Samples. Um sie einzuschalten, muß man am rechten Rand der Echosektion die Anzahl der Wiederholungen wählen, wobei »C« für Dauerwiederholung steht. Ausgeschaltet wird das Echo, indem die Anzahl der Wiederholungen auf null gestellt wird. Daneben stehen noch zwei Regler zur Verfügung, »DL« und »VL«. »DL« bestimmt die Verzögerung, der Regler »VL« die Lautstärkeabnahme der einzelnen "Echos".

(jb)

Listings

```

1 ;*****
2 ;*** Jingle Machine v1.2 von Oliver Heggelbacher ***
3 ;*** Assembler: Devpac v2.14D ***
4 ;*** ARP.library v33+ und passendes INCLUDE-File ***
5 ;*** wird benötigt ***
6 ;*****
7 OPT O+,OW-
8
9 ; Assemblieren des Schriftscrollings ?
10 Scrolling SET 1
11
12 ;----- Includes -----
13 INCDIR "RAD:include/","include/"; eventuell a
14 npassen
15 INCLUDE arp/arpbase.i
16 INCLUDE exec/exec lib.i
17 INCLUDE intuition/intuition lib.i
18 INCLUDE graphics/graphics lib.i
19 INCLUDE hardware/custom.i
20 UCopIns equ $14
21 RSRESET
22 UCopList rs.b 0
23 ucl Next rs.l 1
24 ucl FirstCopList rs.l 1
25 ucl CopList rs.l 1
26 ucl_SIZEOF rs.w 0
27
28 DMABase equ $dff000

```

Listing: JingleMachine.S



```

29 CIAApra equ $bfe001
30
31 ;----- Strukturen -----
32 ; Sample-Struktur
33 RSRESET
34 used rs.b 1 ; 0 = nicht verwendet
35 vol rs.b 1 ; Lautstärke, max. 255
36 rate rs.b 1 ; Abtaste, max. 255
37 loop rs.b 1 ; Wiederholung, 0 = nein
38 adr rs.l 1 ; Speicheradr
39 len rs.l 1 ; Samplelänge
40 fname rs.b 34 ; Dateiname, max. 32 Zeichen
41 sdir rs.b 34 ; Pfad
42 slen rs.b 0
43 sanz equ 48
44
45 ; Gadget-Struktur
46 RSRESET
47 gtyp rs.b 1 ; 1=keine Reakt.,2=On/Off,3=Blink
48 ,4=Regler,5=On
49 gstate rs.b 1 ; ein/aus bzw. Reglerstellung
49 gx rs.b 1
50 gy rs.b 1
51 gbr rs.b 1
52 gho rs.b 1
53 gx1 rs.w 1
54 gy1 rs.w 1
55 gx2 rs.w 1
56 gy2 rs.w 1
57 glen rs.b 0
58 ganz equ 100
59

```

Listing: JingleMachine.S

Listing des Monats

```

60 ; Sound-Struktur
61 RSRESET
62 chact rs.b 1 ; 0=aus , 255=eingeschaltet
63 chused rs.b 1 ; 255=benutzt, 254=Hilfskanal fue
64 r PHASE
65 chvol rs.w 1
66 chrep rs.w 1
67 chper rs.w 1
68 chph rs.w 1
69 chlen rs.b 0
70 chanz equ 4
71 ;----- Makros -----
72 PUSH MACRO
73 move.l \1,-(sp)
74 ENDM
75
76 POP MACRO
77 move.l (sp)+,\1
78 ENDM
79
80 PUSHM MACRO
81 movem.l \1,-(sp)
82 ENDM
83
84 POPM MACRO
85 movem.l (sp)+,\1
86 ENDM
87
88 CALLARP MACRO
89 move.l ArpBase,a6
90 jsr LVO\1(a6)
91 ENDM
92
93 CLOSEARP MACRO
94 move.l ArpBase,a1
95 CALLEXC CloseLibrary
96 POPM a0/d0
97 ENDM
98
99 MAKEGAD MACRO
100 moveq #\1,d0
101 moveq #\2,d1
102 moveq #\3,d2
103 moveq #\4,d3
104 moveq #\5,d4
105 bsr InitGad
106 ENDM
107
108 DISPNUM MACRO
109 moveq #\1,d1
110 moveq #\2,d2
111 moveq #\3,d3
112 moveq #\4,d4
113 bsr PrNum
114 ENDM
115
116 MAKEFRA MACRO
117 moveq #\1,d0
118 moveq #\2,d1
119 moveq #\3,d2
120 moveq #\4,d3
121 moveq #\5,d4
122 bsr MakeFr
123 ENDM
124
125 COPYSTR MACRO
126 .cl\@ move.b (\1)+,(\2)
127 tst.b (\2)+
128 bne.s .cl\@
129 ENDM
130
131 FILEREQ MACRO
132 lea \1,a1
133 lea \2,a2
134 lea \3,a5
135 bsr FRequest
136 ENDM
137
138 MESSAGE MACRO
139 lea .mte\@,a0
140 bsr PrError
141 bra.s .mjm\@
142 .mte\@ dc.b "\1",0
143 EVEN
144 .mjm\@
145 ENDM
146
147 ERROR MACRO
148 MESSAGE <ERROR: \1>
149 move.l ScreenHd,a0
150 CALLINT DisplayBeep
151 ENDM
152
153 ***** Initialisierung, Hauptschleife *****
154 SECTION JMCode,CODE
155
156 Init OPENARP
157 beq NoARP
158 move.l a6, ArpBase
159 bsr FillJingles
160 bsr OpenScreen
161 tst.b d0
162 bne NoMem
163 IFD Scrolling
164 bsr InitScroll
165 ENDC
166 bsr InitSInt
167 bsr InitEff
168 move.b #-1,rnum
169 move.b #-1,jbnum
170

```

Listing: JingleMachine.S

```

171 bsr BuildJD
172 bsr JinGadgets
173
174 moveq #0,d0
175 .loop1 bsr GetChAdr
176 clr.b chused(a1)
177 clr.w chvol(a1)
178 addq.b #1,d0
179 cmp.b #chanz,d0
180 blo.s .loop1
181
182 bsr SetNMode
183 MAKEFRA 1,18,38,18,10
184 move.b #-1,peflag
185 bsr DelError
186
187 MainLoop
188 bsr Play
189 cmp.b #1,abortfl
190 bne.s Exit
191 bsr Change
192 bra.s MainLoop
193
194 Exit bsr Requester
195 tst.b d0
196 bne.s MainLoop
197
198 bsr ResetEff
199 bsr ResetSInt
200 bsr ClearMem
201 IFD Scrolling
202 bsr RemScroll
203 ENDC
204 bsr CloseScreen
205 NoMem CLOSEARP
206 NoARP rts
207
208 FillJingles
209 lea Sample,a0
210 move.l #sanz*slen-1,d0
211 .fj2 clr.b (a0)+
212 dbra d0,.fj2
213 rts
214
215 ***** Jingles editieren *****
216 Change
217 bsr BuildEdit
218 bsr SetGadgets
219 clr.b abortfl
220 .ch2 bsr MouseControl
221 bsr.s TestSetKeys
222 tst.b abortfl
223 beq.s .ch2
224 rts
225
226 TestSetKeys
227 cmp.b #48,d0
228 blo JinSel2
229 cmp.b #-2,d0
230 beq KeyDown
231 cmp.b #-3,d0
232 beq KeyUp
233 cmp.b #51,d0
234 blo VarRefr
235 cmp.b #51,d0
236 beq LoadSam
237 cmp.b #52,d0
238 beq LoadSet
239 cmp.b #53,d0
240 beq SaveSet
241 cmp.b #54,d0
242 beq.s CleanUp
243 cmp.b #55,d0
244 beq.s EndEdit
245 rts
246
247 EndEdit move.b #255,abortfl
248 rts
249
250 CleanUp bsr Requester
251 tst.b d0
252 bne.s .abort
253 bsr ClearMem
254 bsr FillJingles
255 bsr InstallSam
256 bsr PrMem
257 .abort rts
258
259 JinSelect
260 PUSH d0
261 move.b jnum,d0
262 bsr GetGadPt
263 bsr InvSW2
264 POP d0
265 move.b d0,jnum
266 bsr GetGadPt
267 bsr InvSW2
268 rts
269
270 JinSel2
271 bsr JinSelect
272 JSelect bsr.s PrsName
273 bsr.s GetSPoint
274 move.l len(a0),d0
275 moveq #6,d1
276 moveq #16,d2
277 moveq #23,d3
278 moveq #15,d4
279 moveq #11,d5
280 PUSHM a0-a1
281 bsr PrNum
282 POPM a0-a1

```

Listing: JingleMachine.S




```

283      moveq    #48,d0
284      move.b   vol(a0),d1
285      PUSHM    a0-a1
286      bsr      Switch
287      POPM     a0-a1
288      moveq    #49,d0
289      move.b   rate(a0),d1
290      PUSHM    a0-a1
291      bsr      Switch
292      POPM     a0-a1
293      moveq    #50,d0
294      move.b   loop(a0),d1
295      PUSHM    a0-a1
296      bsr      Switch
297      POPM     a0-a1
298      rts
299
300 GetSPoint
301      clr.w    d7
302      move.b   jnum,d7
303      mulu     #slen,d7
304      lea      Sample,a0
305      lea      (a0,d7),a0
306      rts
307
308 PrSName moveq    #0,d0
309      move.b   jnum,d0
310      mulu     #slen,d0
311      lea      Sample,a1
312      lea      (a1,d0),a1
313      lea      JStr+6,a2
314      lea      fname(a1),a3
315      COPYSTR  a3,a2
316 .loop2  move.b   #32,-1(a2)
317      tst.b    (a2)+
318      bne.s    .loop2
319      lea      JStr,a0
320      bra      Print
321 JStr    dc.b    1,7,20,2,13,4
322      dcb.b    33,32
323      dcb.b    0,255
324      EVEN
325
326 VarRefr move.b   gstate(a0),d1
327      bsr      GetSPoint
328      cmp.b    #48,d0
329      bne.s    .var1
330      move.b   d1,vol(a0)
331      cmp.b    #49,d0
332      bne.s    .var2
333      move.b   d1,rate(a0)
334      cmp.b    #50,d0
335      bne.s    .var3
336      move.b   d1,loop(a0)
337      rts
338
339 ***** Diskzugriffe *****
340
341 ;----- Diskettenmacros -----
342 FMRead  equ 1005
343 FMWrite equ 1006
344
345 OPENDAT MACRO
346      move.l   #DatName,d1
347      move.l   #FM\1,d2
348      CALLARP  Open
349      move.l   d0,FileHd
350      ENDM
351 CLOSDAT MACRO
352      move.l   FileHd,d1
353      CALLARP  Close
354      ENDM
355
356 WRITDAT MACRO
357      move.l   \1,d2
358      move.l   \2,d3
359      move.l   FileHd,d1
360      CALLARP  Write
361      ENDM
362 READDAT MACRO
363      move.l   \1,d2
364      move.l   \2,d3
365      move.l   FileHd,d1
366      CALLARP  Read
367      ENDM
368
369 WRCODE  MACRO
370      move.l   #\1,DCode
371      WRITDAT  #DCode,#4
372      ENDM
373 RECODE  MACRO
374      READDAT  #DCode,#4
375      move.l   DCode,d0
376      cmp.l    #\1,d0
377      ENDM
378
379 WRLONG  MACRO
380      move.l   \1,DCode
381      WRITDAT  #DCode,#4
382      ENDM
383 GETLONG MACRO
384      READDAT  #DCode,#4
385      move.l   DCode,\1
386      ENDM
387
388 JUMPDAT MACRO
389      move.l   \1,d2
390      move.l   FileHd,d1
391      moveq    #0,d3
392      CALLARP  Seek
393      ENDM
394

```

Listing: JingleMachine.S



```

395 ;----- SET speichern/laden
396 SaveSet FILEREQ SetFN,SetDR,SetReq
397      tst.b    d0
398      beq.s    .abort
399      OPENDAT  Write
400      beq.s    .abort
401      WRCODE   "JmSt"
402      WRITDAT   #Sample,#slen*sanz
403      CLOSDAT
404      .abort    rts
405
406 LoadSet  move.l   #"Load",SetRTex
407      FILEREQ  SetFN,SetDR,SetReq
408      move.l   #"Save",SetRTex
409      tst.b    d0
410      beq.s    .abort
411      OPENDAT  Read
412      beq.s    .cof
413      RECODE   "JmSt"
414      bne.s    .nsfil
415      bsr      ClearMem
416      READDAT  #Sample,#slen*sanz
417      CLOSDAT
418      bsr.s    InstallSam
419      .abort    rts
420      .cof      ERROR <Cannot open file>
421      rts
422      .nsfil    CLOSDAT
423      ERROR <Not a SET file>
424      rts
425
426 InstallSam
427      moveq    #0,d0
428      .loop    PUSH    d0
429      bsr      PrSample2
430      bsr      JinSelect
431      bsr      JSelect
432      tst.b    used(a0)
433      beq.s    .nused
434      lea      fname(a0),a3
435      lea      sdir(a0),a4
436      bsr      MixPath
437      bsr.s    GetDat
438      cmp.b    #13,d0
439      beq.s    .lsucc
440      POP      d0
441      bsr      PrSample2
442      PUSH     d0
443      .lsucc   bsr      PrMem
444      .nused   POP      d0
445      addq.b    #1,d0
446      cmp.b    #sanz,d0
447      blo.s    .loop
448      moveq    #0,d0
449      bsr      JinSelect
450      rts
451
452 ;----- einzelnes Sample laden
453 LoadSam bsr      GetSPoint
454      FILEREQ  fname(a0),sdir(a0),SamReq
455      tst.b    d0
456      beq.s    .abort
457      PUSH     a0
458      bsr      FreeSample
459      POP      a0
460      move.b   #255,vol(a0)
461      move.b   #128,rate(a0)
462      bsr.s    GetDat
463      .swoff   bsr      JSelect
464      bsr      PrMem
465      moveq    #0,d0
466      move.b   jnum,d0
467      bsr      PrSample2
468      .abort    rts
469
470 ;----- IFF-8SVX Datei lesen
471 GetDat   clr.b    used(a0)
472      PUSH     a0
473      OPENDAT  Read
474      beq      .fnf
475      RECODE   "FORM"
476      bne      .niff
477      GETLONG  d0 ; Formlen
478      RECODE   "8SVX" ; Kennung
479      bne      .niff
480      .sebody  RECODE  "BODY" ; Body suchen
481      beq.s    .bodyfound
482      GETLONG  d1 ; Chunklen
483      tst.l    d0
484      beq      .nbod ; Dateiende erreicht
485      addq.l    #1,d1
486      bclr     #0,d1
487      JUMPDAT  d1 ; Chunk ueberspringen
488      bra.s    .sebody ; weitersuchen
489      .bodyfound
490      GETLONG  d0 ; Body-Laenge
491      POP      a0
492      cmp.l    #20000,d0
493      blo.s    .nottl
494      move.l    #1ffff,d0
495      PUSH     d0
496      MESSAGE  <File truncated>
497      POP      d0
498      .nottl   bclr     #0,d0
499      bclr     #1,d0
500      move.l    d0,len(a0)
501      bsr      AllocSample
502      PUSH     a0
503      tst.b    d0
504      beq.s    .oom
505      READDAT  adr(a0),len(a0)
506      tst.l    d0

```

Listing: JingleMachine.S

Listing des Monats

```

507      beq      .crf
508      CLOSDAT
509      POP      a0
510      move.b   #255,used(a0)
511      move.b   #255,jbnum
512      moveq    #13,d0
513      rts
514 .oom      ERROR <Out of memory>
515      bra     .pa0
516 .fnf      ERROR <Cannot open file>
517      bra     .pa0
518 .niff     ERROR <Not an IFF-8SVX file>
519      bra.s   .cdat
520 .nbod     ERROR <No BODY chunk found>
521      bra.s   .cdat
522 .crf      ERROR <Cannot read file>
523 .cdat     CLOSDAT
524 .pa0      POP      a0
525      rts
526
527 ; Filerequester des ARP-Library aufrufen
528 FRequest
529      move.l   fr_File(a5),a3
530      move.l   fr_Dir(a5),a4
531      PUSHM   a0-a4
532      PUSH    a5
533      COPYSTR a1,a3
534      move.l   ScreenHd,a0
535      CALLINT ScreenToBack
536      POP     a0
537      CALLARP FileRequest
538      PUSH    d0
539      move.l   ScreenHd,a0
540      CALLINT ScreenToFront
541      POP     d0
542      POPM    a0-a4
543      tst.b   d0
544      beq.s   .frab
545      PUSHM   a1-a4
546      COPYSTR a3,a1
547      COPYSTR a4,a2
548      POPM    a1-a4
549      bsr.s   MixPath
550      moveq    #-1,d0
551 .frab     rts
552
553 MixPath  lea     DatName,a5
554      tst.b   (a4)
555      beq.s   .nopath
556      COPYSTR a4,a5
557      tst.b   -(a5)
558      cmp.b   #"/",-1(a5)
559      beq.s   .nopath
560      cmp.b   #":",-1(a5)
561      beq.s   .nopath
562      move.b   #"/",(a5)+
563 .nopath  COPYSTR a3,a5
564      rts
565
566 ***** Speicherverwaltung *****
567 SamMType equ 1
568 BufMType equ 3
569
570 AllocSample
571      move.l   SBufLen,d0
572      move.l   d0,SBufOld
573      bsr     FreeSBuf
574      move.l   len(a0),d0
575      moveq    #SamMType,d1
576      PUSH    a0
577      CALLEEXEC AllocMem
578      POP     a0
579      move.l   d0,adr(a0)
580      beq.s   .memfull
581      move.l   SBufLen,d0
582      cmp.l   len(a0),d0
583      bhl.s   .smaller
584      move.l   len(a0),d0
585      move.l   d0,SBufLen
586 .smaller
587      bsr     AllocSBuf
588      beq.s   .nobufmem
589      moveq    #-1,d0
590      rts
591 .memfull
592      bsr.s   AllocSBuf
593      moveq    #0,d0
594      rts
595 .nobufmem
596      move.l   adr(a0),a1
597      move.l   len(a0),d0
598      PUSH    a0
599      CALLEEXEC FreeMem
600      POP     a0
601      move.l   SBufOld,d0
602      move.l   d0,SBufLen
603      bsr.s   AllocSBuf
604      moveq    #0,d0
605      rts
606
607 FreeSample
608      tst.b   used(a0)
609      beq.s   .abort
610      bsr.s   FreeSBuf
611      PUSHM   a0/d0
612      move.l   adr(a0),a1
613      move.l   len(a0),d0
614      CALLEEXEC FreeMem
615      POPM    a0/d0
616      move.l   #0,len(a0)
617      move.b   #0,used(a0)
618      PUSH    d0

```

Listing: JingleMachine.S

```

619      bsr.s   AllocSBuf
620      POP     d0
621 .abort    rts
622
623 FreeSBuf
624      PUSHM   a0/d0
625      move.l   SBufAdr,a1
626      move.l   SBufLen,d0
627      tst.l   d0
628      beq.s   .abort
629      CALLEEXEC FreeMem
630 .abort    POPM   a0/d0
631      rts
632
633 AllocSBuf
634      move.l   SBufLen,d0
635      tst.l   d0
636      beq.s   .abort
637      moveq    #BufMType,d1
638      PUSH    a0
639      CALLEEXEC AllocMem
640      POP     a0
641      move.l   d0,SBufAdr
642      tst.l   d0
643      bne.s   .abort
644      move.l   d0,SBufLen
645 .abort    rts
646
647 ClearMem
648      bsr.s   FreeSBuf
649      clr.l   SBufLen
650      moveq    #0,d0
651      lea     Sample,a0
652 .loop     bsr     FreeSample
653      lea     slen(a0),a0
654      addq.b   #1,d0
655      cmp.b   #sanz,d0
656      blo.s   .loop
657      bra     DelError
658      rts
659
660 ***** Jingles spielen *****
661 Play     bsr     BuildPlay
662      bsr     PlayGadgets
663      bsr     PlPrefs
664      clr.b   abortfl
665      move.b   #255,dvflag
666 .play2    bsr     MouseControl
667      bsr     TestPlayKeys
668      tst.b   abortfl
669      beq.s   .play2
670      clr.b   dvflag
671      bsr     PGStore
672      rts
673
674 MAKEPR   MACRO
675      moveq    #1,d0
676      moveq    #2,d1
677      bsr     PInit
678      ENDM
679
680 PlPrefs  tst.b   prflag
681      bne     PGRestore
682      move.b   #255,prflag
683      MAKEPR  48,-1
684      MAKEPR  49,-1
685      MAKEPR  50,-1
686      MAKEPR  51,-1
687      moveq    #0,d0
688 .cloop    bsr     GetChAdr
689      move.b   #255,chact(a1)
690      addq.b   #1,d0
691      cmp.b   #chanz,d0
692      blo.s   .cloop
693      MAKEPR  52,-1
694      MAKEPR  54,-1
695      MAKEPR  65,-128
696      moveq    #59,d0
697      moveq    #-1,d1
698      bsr     Switch
699      clr.b   EchoRep
700      clr.b   EchoDel
701      clr.b   EchoVol
702      clr.b   Hold
703      clr.b   Mono
704      clr.b   PhAct
705      clr.b   Lesley
706      rts
707
708 PGStore  lea     GadBuf,a1
709      moveq    #48,d0
710      bsr     GetGadPt
711 .pgs2     move.b   gstate(a0),(a1)+
712      lea     qlen(a0),a0
713      addq.b   #1,d0
714      cmp.b   #gused,d0
715      blo.s   .pgs2
716      rts
717
718 PGRestore
719      lea     GadBuf,a1
720      moveq    #48,d0
721 .pgr2     move.b   (a1)+,d1
722      PUSH    a1
723      bsr     Switch
724      POP     a1
725      addq.b   #1,d0
726      cmp.b   #gused,d0
727      blo.s   .pgr2
728      rts
729
730 PInit

```

Listing: JingleMachine.S




```

731      bsr      Switch
732 TestPlayKeys
733      cmp.b     #48,d0
734      blo      JinSelect
735      cmp.b     #-2,d0
736      beq      KeyDown
737      cmp.b     #-3,d0
738      beq      KeyUp
739      cmp.b     #48,d0
740      blo      PJSel
741      moveq     #0,d1
742      move.b    gstate(a0),d1
743      cmp.b     #52,d0
744      blo.s     ChanSel
745      beq.s     SMainVol
746      cmp.b     #53,d0
747      beq.s     SetMDecay
748      cmp.b     #54,d0
749      beq.s     SetFilter
750      cmp.b     #55,d0
751      beq      SetHold
752      cmp.b     #56,d0
753      beq.s     JEdit
754      cmp.b     #57,d0
755      beq      SetEDel
756      cmp.b     #58,d0
757      beq      SetEVOL
758      cmp.b     #64,d0
759      blo      SetERep
760      beq.s     SetMono
761      cmp.b     #65,d0
762      beq.s     SetPhase
763      cmp.b     #66,d0
764      beq.s     SetLes
765      cmp.b     #67,d0
766      beq.s     SetPhAct
767      cmp.b     #68,d0
768      beq.s     ExitPrg
769      rts
770
771 JEdit      move.b    #1,abortfl
772      rts
773 ExitPrg
774      move.b    #-1,abortfl
775      rts
776 PJSel
777      move.l     a0,SampPt
778      rts
779 ChanSel
780      sub.b     #48,d0
781      bsr      GetChAdr
782      eor.b     #255,chact(a1)
783      rts
784 SMainVol
785      move.b    d1,MainVol
786      rts
787 SetMDecay
788      move.b    d1,Decay
789      rts
790 SetFilter
791      move.b    d1,Filter
792      beq.s     FiltOff
793      bclr      #1,CIAApra
794      rts
795 FiltOff    bset      #1,CIAApra
796      rts
797 SetMono
798      move.b    d1,Mono
799      rts
800 SetPhase
801      neg.b     d1
802      move.b    d1,Phase
803      rts
804 SetPhAct
805      move.b    d1,PhAct
806      rts
807 SetLes
808      move.b    d1,Lesley
809      rts
810 SetHold
811      move.b    d1,Hold
812      rts
813
814 SetEDel    move.b    d1,EchoDel
815      rts
816 SetEVOL    move.b    d1,EchoVol
817      rts
818 SetERep    sub.b     #59,d0
819      move.b    EchoRep,d2
820      cmp.b     d0,d2
821      beq.s     .nonset
822      move.b    d0,EchoRep
823      move.b    d2,d0
824      add.b     #59,d0
825      moveq     #0,d1
826      bra      Switch
827 .nonset    rts
828
829 ***** Ausgabe der Sounds *****
830 KeyUp      tst.b     Hold
831      bne.s     .thold
832      clr.b     EchoSta
833      moveq     #chanz-1,d0
834 .loop1     bsr      GetChAdr
835      bsr      FlushChan
836      dbra     d0,.loop1
837 .thold     rts
838
839 KeyDown    clr.w     lestate
840      bsr      GetSPoint
841      move.l     a0,ASampPt
842      bsr      CopySam

```

Listing: JingleMachine.S



```

843      move.b    vol(a0),EchoVL2
844      bsr.s     InstallTone
845      bsr      InitTimer
846      clr.w     EchoCou
847      clr.w     d1
848      move.b    EchoDel,d1
849      mulu     d1,d1
850      lsr.w     #7,d1
851      move.w     d1,EchoC2
852      move.b    EchoRep,EchoSta
853      rts
854
855 InstallTone
856      move.l     ASampPt,a0
857      tst.b     used(a0)
858      beq.s     .nocav
859      moveq     #0,d0
860      move.b    actchan,d0
861 .search    bsr.s     SelChan
862      tst.b     chact(a1)
863      beq.s     .noton
864      tst.b     chused(a1)
865      beq.s     .found
866 .noton     cmp.b    actchan,d0
867      bne.s     .search
868 .noton2    bsr.s     SelChan
869      tst.b     chact(a1)
870      beq.s     .noton2
871 .found
872      move.b    d0,actchan
873      bsr      FlushChan
874      move.b    EchoVL2,d1
875      move.b    loop(a0),d2
876      bsr      FillChan2
877      tst.b     Mono
878      beq.s     .sdma
879      addq.b    #2,d0
880      and.b     #3,d0
881      bsr      GetChAdr
882      bsr      FlushChan
883      move.b    EchoVL2,d1
884      move.b    loop(a0),d2
885      bsr      FillChan2
886 .sdma      bsr      AudDMAOn
887 .nocav     rts
888
889 SelChan    addq.b    #1,d0
890      cmp.b     #chanz,d0
891      blo.s     .selc2
892      moveq     #0,d0
893 .selc2     bra      GetChAdr
894
895 WAITADM    MACRO
896 .wadml\@
897      move.w     vhposr(a3),d1
898      cmp.b     #50,d1
899      blo.s     .wadml\@
900 .wadm2\@
901      move.w     vhposr(a3),d1
902      cmp.b     #50,d1
903      bhs.s     .wadm2\@
904 .wadm3\@
905      move.w     vhposr(a3),d1
906      cmp.b     #50,d1
907      blo.s     .wadm3\@
908      ENDM
909
910 FlushChan
911      tst.b     chused(a1)
912      beq.s     .abort
913      clr.w     chvol(a1)
914      clr.b     chused(a1)
915      lea      DMABase,a3
916      clr.w     d1
917      clr.w     d2
918      bset     d0,d1
919      bset     d0,d2
920      lsl.w     #7,d2
921      move.w     d1,dmacon(a3)
922      move.w     d2,intena(a3)
923      move.w     d2,intreq(a3)
924      WAITADM
925 .abort     rts
926
927 FillChan2
928      bsr.s     FillChan
929      tst.b     PhAct
930      beq.s     .nophase
931      PUSH     d0
932      PUSHM    d1-d2
933      eor.b     #3,d0
934      bsr      GetChAdr
935      bsr      FlushChan
936      POPM     d1-d2
937      bsr.s     FillChan
938      move.b    #254,chused(a1)
939      POP      d0
940 .nophase
941      rts
942
943 FillChan
944      move.b     #255,chused(a1)
945      move.w     d1,d7
946      lsl.w     #8,d7
947      move.w     d7,chvol(a1)
948      move.b     d2,chrep(a1)
949      move.l     SBufAdr,ac_ptr(a2)
950      move.l     len(a0),d7
951      lsr.l     #1,d7
952      move.w     d7,ac len(a2)
953      moveq     #0,d3-
954      move.b     rate(a0),d3

```

Listing: JingleMachine.S

Listing des Monats

```

955      bsr.s   SetRate
956      bsr     SetDMA
957      rts
958
959 SetRate eor.b   #255,d3
960        mulu   d3,d3
961        lsr.w   #7,d3
962        addi.w  #124,d3
963        move.w  d3,chnper(a1)
964        move.w  d3,ac_per(a2)
965        rts
966
967 SetVol  move.w  chvol(a1),d1
968        lsr.w   #8,d1
969        tst.b   Lesley
970        beq.s   .noles
971        move.w  lestate,d2
972        lsr.w   #7,d2
973        cmp.w   #256,d2
974        blo.s   .seph
975        andi.w  #255,d2
976        eor.b   #255,d2
977 .seph  cmp.b   #1,d0
978        beq.s   .rchn
979        cmp.b   #2,d0
980        beq.s   .rchn
981        eor.b   #255,d2
982 .rchn  mulu   d2,d1
983        lsr.w   #8,d1
984 .noles  clr.w   d2
985        move.b  MainVol,d2
986        mulu   d2,d1
987        lsr.w   #8,d1
988        lsr.b   #2,d1
989        move.b  d1,ac_vol(a2)
990        tst.b   dvflag
991        bne.s   DrawVol
992        rts
993
994 DrawVol move.l  Plane3,a4
995        lea     184*40+1(a4),a4
996        clr.l   d2
997        move.b  d0,d2
998        add.l   d2,a4
999        lsr.w   #1,d1
1000       eor.b   #31,d1
1001       bclr    #0,d1
1002       moveq   #0,d2
1003       moveq   #-1,d3
1004 .dvloop  cmp.b  d2,d1
1005         bne.s  .dv12
1006         move.b #10000001,d3
1007 .dv12   move.b  d3,(a4)
1008         lea     80(a4),a4
1009         addq.b  #2,d2
1010         cmp.b  #32,d2
1011         blo.s  .dvloop
1012         rts
1013
1014 SetDMA  move.w  dmareg,d7
1015         bset    d0,d7
1016         move.w  d7,dmareg
1017         rts
1018 AudDMAon
1019         lea     DMABase,a3
1020         move.w  dmareg,d1
1021         lsl.w   #7,d1
1022         move.w  d1,intena(a3)
1023         WAITADM
1024         move.w  dmareg,d1
1025         bset    #15,d1
1026         move.w  d1,dmacon(a3)
1027         tst.b   chrep(a1)
1028         bne.s   .samrep
1029         WAITADM
1030         move.w  dmareg,d1
1031         lsl.w   #7,d1
1032         move.w  d1,intreq(a3)
1033         bset    #15,d1
1034         move.w  d1,intena(a3)
1035 .samrep  clr.w   dmareg
1036         rts
1037 dmareg  dc.w 0
1038
1039 CopySam tst.l   SBufLen
1040         beq.s   .abort
1041         move.b  jnum,d1
1042         cmp.b  jbnum,d1
1043         beq.s   .abort
1044         tst.b   used(a0)
1045         beq.s   .abort
1046         move.b  d1,jbnum
1047         move.l  adr(a0),a3
1048         move.l  SBufAdr,a4
1049         move.l  len(a0),d1
1050         subq.l  #1,d1
1051         lsr.l   #2,d1
1052 .copyl  move.l  (a3)+,(a4)+
1053         dbra   d1,.copyl
1054         .abort  rts
1055
1056 GetChAdr
1057         moveq   #0,d7
1058         move.b  d0,d7
1059         mulu   #chlen,d7
1060         lea     Channel,a1
1061         lea     (a1,d7),a1
1062         moveq   #0,d7
1063         move.b  d0,d7
1064         mulu   #ac_SIZEOF,d7
1065         lea     DMABase+aud,a2
1066         lea     (a2,d7),a2

```

Listing: JingleMachine.S



```

1067      rts
1068
1069 ;----- Effekt-Interrupt -----
1070 InitEff
1071        PUSHM   d1-d7/a0-a6
1072        lea     CiaaResource,a1
1073        moveq   #0,d0
1074        CALLEXC OpenResource
1075        move.l  d0,CiaaBase
1076        move.l  d0,a6
1077        lea     Interrupt,a1 ;Effekt-Interrupt Struktur
1078        moveq   #0,d0
1079        jsr     -6(A6)        ; TimerA installieren
1080        move.l  d0,d5
1081        bsr.s   InitTimer    ; Timer starten
1082        moveq   #0,d0
1083 EndStart
1084        tst.l   d5            ; Ergebnis von Resource
1085        sne     d0
1086        POPM    d1-d7/a0-a6
1087        rts
1088
1089 ResetEff
1090        PUSHM   d1-d7/a0-a6
1091        move.l  CiaaBase,a6
1092        lea     Interrupt,a1
1093        moveq   #0,d0
1094        jsr     -12(A6)       ; Interrupt entfernen
1095        moveq   #0,d0
1096        POPM    d1-d7/a0-a6
1097        rts
1098
1099 InitTimer
1100        move.b  #10000001,$bfee01 ;Timer starten
1101        lea     CINTDelay,a1
1102        move.b  1(a1),$bfe401     ;Timer A low
1103        move.b  0(a1),$bfe501     ;Timer A high
1104        rts
1105
1106 IntCode  PUSHM   a0-a6/d1-d7
1107          bsr.s   DoEffect
1108          POPM    a0-a6/d1-d7
1109          moveq   #0,d0
1110          rts
1111          ;Effekt erzeugen
1112          ;kein Fehler
1113
1114 SECTION JMDData,DATA
1115 Interrupt
1116        dc.l    0            ;letzter Node
1117        dc.l    0            ;nächster Node
1118        dc.b    2            ;Node Type = Interrupt
1119        dc.b    0            ;Priorität
1120        dc.l    InterruptName ;Name
1121        dc.l    0            ;Zeiger auf Daten
1122        dc.l    IntCode      ;Interrupt Routine
1123
1124 InterruptName
1125        dc.b    "JingleInt",0
1126
1127 CiaaResource
1128        dc.b    "ciaa.resource",0
1129
1130 SECTION JMCode,CODE
1131 DoEffect
1132        bsr.s   Echo
1133        bsr     DoLes
1134        moveq   #0,d0
1135        bsr.s   ChanEff
1136        moveq   #1,d0
1137        bsr.s   ChanEff
1138        moveq   #2,d0
1139        bsr.s   ChanEff
1140        moveq   #3,d0
1141        bsr.s   ChanEff
1142        rts
1143
1144 ChanEff  bsr     GetChAdr
1145          tst.b   chused(a1)
1146          beq.s   .chnotused
1147          bsr.s   VolDec
1148          bsr     DoPhase
1149          .chnotused
1150          bsr     SetVol
1151          rts
1152
1153 VolDec  bsr     GetChAdr
1154          clr.w   d1
1155          move.b  Decay,d1
1156          lsl.w   #1,d1
1157          move.w  chvol(a1),d2
1158          sub.w   d1,d2
1159          bcc.s   .snzero
1160          bsr     FlushChan
1161          clr.w   d2
1162          move.w  d2,chvol(a1)
1163          rts
1164
1165 Echo    move.b  EchoSta,d1
1166          tst.b  d1
1167          beq.s  .noecho
1168          move.w  EchoCou,d2
1169          addq.w  #1,d2
1170          cmp.w   EchoC2,d2
1171          bhs.s   .timere
1172          move.w  d2,EchoCou
1173          rts
1174          .timere  clr.w  EchoCou
1175                  cmp.b  #4,d1
1176                  beq.s  .echcont
1177                  subq.b #1,d1
1178                  move.b d1,EchoSta
1179                  .echcont

```

Listing: JingleMachine.S

mailsoft

DAS Software-Versandhaus
in der
SCHWEIZ

Die neuesten Games und
Anwenderprogramme ab Lager
lieferbar.

Sofort Gratisliste anfordern!

AMIGA PUBLIC DOMAIN

Fish	-360	Taifun	-140
RPD	-215	Oase	-51
Kickstart	-300	Cactus	-38
Panorama	-32d	SAFE	-39

mailsoft

Postfach, 8330 Pfäffikon ZH
Tel. 01/9 50 56 11

DATATRANS

Endlich die Daten Übertragungsmöglichkeit:

Vom Commodore C64 / C 128 zum AMIGA 500 / 1000 / 2000

Das Kit besteht aus folgenden Komponenten:
1.) Software Diskette für C 64 / C 128
2.) Übertragungskabel Länge ca. 2 Meter
3.) Anpassungsprogramm für die Empfangsseite
4.) Ausführliche Bedienungsanleitung
5.) Tel. Hotline für spezielle Fragen



Unser Übertragungs-Kit ist auch für PC's +
ATARI-ST's bestimmt!

Preis 60,- DM incl. Versandkosten.

Händler Anfragen erwünscht!

HVK Kunze Hollandstr. 119 4500 Dortmund 12
Tel. 0231/259090
FAX 0231/201565

(c) by Wide Soft

5000

AMIGA Public-Domain

80 SUPER AMIGA-PD Serien

Ultraschnell-Aktuell-Preiswert-Zuverlässig

Fish, Kickstart, TBAG, Franz, Taifun, Antares...

Seit 1985 haben wir AMIGA-PD

Incl. 3,5 MF2DD Disk

ab 1 - 79 Disk a 2,00 DM

3,5 MF2DD schon ab 1,60 DM

Incl. 5,25 Disk

10 - 99 a 1,20, ab 100 a 1,00 DM

Wir kopieren mit doppeltem Verify.

AMIGA PD incl. Markendisk auf Anfrage.

Katalogdiskette 3,- DM, bei Vorkasse in Briefm.

oder fordern Sie unser Info-Material an 1,- DM

Montag-Freitag 10.00-20.00, Samstag 11.00-16.00

Tel: 040/6428225, Telefax: 040/6426913

Vork. + 4,- DM, Nachn. + 8,- DM, Ausland + 16,- DM

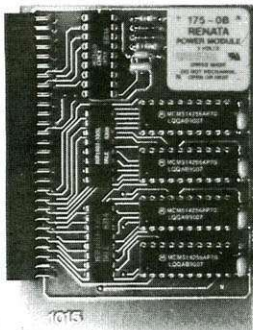
R. Dombrowski

Postfach 710462

2000 Hamburg 71

AMIGA 500 512 k

Speichererweiterung



- Megabit Technologie
- Abschaltbar inkl. Schalter
- Uhr und Batterie

DM 119,-

ohne Uhr

DM 109,-

Hardware Design Udo Neuroth
Postfach 525, 4250 Bottrop
Tel. 02041/20424 ab 16 Uhr

OMEGA Datentechnik

AMIGA 3000 auf Anfrage

DRAM 514256 ZIP.....19,-

für AMIGA 3000 und A 2620 und A2630

DRAM 511000 - 10 Megabit.....17,50

512kB RAM mit Uhr.....125,-

Profisampler 56kHz V2.....125,-

Profisampler stereo.....169,-

AMIGA 2000 RAM-Karten

MicroBotics 2MB...759,- 4MB...1075,-

Supra 2MB...698,- 4MB...999,- 8MB...1599,-

A 2286 AT Karte.....1698,-

ALF2 60MB Filecard.....1298,-

ALF2 84MB SCSI Filecard..1598,-

128 kB RAM-Karte für PC-Karte.145,-

Professional 68030 Karte...1498,-

mit Coproz. 68881 20MHz & 1MB...2398,-

NEC Multisync 3D.....1495,-

OMA V 1.7 Makro-Assembler.....148,-

PageStream 1.8.....298,-

OMEGA Datentechnik 0441 / 71109

Junker Str. 2, 2900 Oldenburg

Daten- und Organisationssysteme

Hard- und Softwarevertrieb

Ihr AMIGA-Fachhändler
im Bergischen Land!



ANGEBOTE

Amiga-Reisware-Maus	89,-
Drehs-BTX/VTX-Manager kompl.	189,-
2-MB-Speichererweiterung für A 500	639,-
8-MB-Speichererweiterung für A 2000	1.998,-
40-MB-Hardcard GVPII-Quantum	1.798,-

Achtung - Achtung - Achtung

Wir reparieren Ihren defekten Amiga preisgünstig

Neu - Neu - Neu - Neu - Neu - Neu

Computershop "Die Diskette"

Friedrich-Engels-Allee 186 - 5600 Wuppertal 2

Öffnungszeiten (Büro + Ladengeschäft)

Mo - Fr 10.00 - 18.00 - Sa 10.00 - 14.00

Sedanstraße 136 * 5600 Wuppertal 2

Tel. 02 02 / 50 15 00 * Martin Kramer

AMIGA - PD CENTER

8500 DISKETTEN

ANGEBOT DES MONATS

KATALOG-SET

5 voll gepackte Disketten incl. Aquarium!
Hochaktuell und komplett in deutsch!!!
Integrierter Update-Service!

ANTI - VIRUS

komplett neu überarbeitete AVD (Anti-Viren-
Disk mit diversen Updates (VirusX), vielen
neuen Progr. und deutschen Anleitungen!

TIME - SPEZIAL

Sonderausgabe der TIME-Serie - 2 Disketten-
mit diversen Updates, neuen Programmen
und natürlich mit neuer ST-Musik!

alles komplett für nur DM 30,- (Vorkasse!)

A.P.S. -e-lectronic-
Sonnenborstel 31
D-3071 Steimbke
Tel.: 05026/1700
FAX 05026/1615



Info-Disk mit akt. Angeboten, News,
Preisanschreiben, Soundtracks,
Serien-Infos, u.s.w. für DM 2,- (VK)

AURIGA TECHNOLOGIE

Mainaustr. 38

8000 München 60

Tel.: 089/8203651

Tel. Bestellannahme Mo. - Fr. 8.00 bis 18.00

unter Tel.: 0821/462511 in Augsburg

Speichererweiterung A2000:

Amiga 2000 Interne Speicherkarte bis 8MB, auf-

rüstbar in 2MB Schritten mit 411000 oder 511000.

autoconfigurierend, Testsoftware, Handbuch

2MB bestückt: DM 699,- 4MB: DM 999,-

FileCards Amiga 2000:

ALF2 SCSI-Kontrollern, autoboot ab Kickstart 1.3.

hohe Datentransferrate, einfache Bedienbarkeit

durch mitgelieferte ALF2-Software

mit Quantum P40S, 19/11ms, 42MB 1585,-

mit Seagate ST1096, 24ms, 83MB 1599,-

Atari-ST-Emulator MEDUSA DM 489,-

GFA-Basic3.5 Interpr.+Compiler DM 329,-

M2Amiga Modula-2 Compiler V3.3 DM 329,-

Drucker:

Citizen SWIFT24, 24 Nadeln DM 799,-

Farboption SWIFT24 DM 120,-

Einzelblatteneinzug SWIFT24 DM 245,-

FontCard SWIFT24, je Stück DM 99,-

Speichererweiterungen

ANRUUFEN !!!! LOHNT SICH !!!!

TAGESPREISE !!!!!

Amiga 500 - 512 KB intern, abschaltbar	ab DM 159,-
Amiga 500 - 1 MB intern, absch., Uhr	ab DM 398,-
Amiga 500 - 2 MB intern, absch., Uhr	ab DM 598,-
Amiga 1000 - 2 MB extern, absch.,	ab DM 798,-
Amiga 2000 - 2 MB intern, absch.,	ab DM 898,-

Alle Speichererweiterungen sind autoconfigurierend, abschaltbar und
mit sehr schnellen RAM's (100ns und schneller) ausgerüstet!

Durch Megabit-Technologie minimaler Strombedarf

****12 Monate Garantie****

Laufwerk 3.5" intern f. Amiga 2000 DM 169,-

Laufwerk 3.5" extern, durchgeschl. Bus, abschaltb. DM 198,-

B & S Computer-Vertriebs GmbH

Beethovenstr. 33 : 4172 Straelen 1

Tel: 02834/1249 : Fax: 02834/6979

PUBLIC DOMAIN CENTER
Postfach 3142
5840 Schwerte

Volltreffer ...

Jede nur

1,90 DM

Info anfordern!

Jörg R. Hiller
fhn computer

public domain software
● **2.40/1.20** ●

jede Disk 3.5"/5.25" 2DD (100% errorfrei)
Alle PD's sind etikettiert (Name, Nr., Inhalt)
Versand in der Regel innerhalb 3 Tagen nach Bestelleingang
Fish, Kickstart, RPD, Taifun, Chiron, ACS Tornado, Faug
Franz, Cactus -> Katalogdisk DM 5,-/Liste gegen Rückporto

Amiga Hardware

Amiga 2000	1989.-	Amiga XT Karte	750.-
3 1/2" LW extern	198.-	PAL Genlock	568.-
5 1/4" LW extern	259.-	Y-C Genlock	1139.-
8MB/2MB A2000	858.-	Deluxe View 4.0	398.-
3 1/2" Disks je	1.49	5 1/4" Disks je	0.59

Weiteres auf Anfrage. Preise in DM zzgl. Versandkosten.

feinauer - hillier
Offenbacher Landstr.14
6450 Hanau 7 / Tel.: 06181/650328

**ALSDORFER
PUBLIC DOMAIN CENTER**

Wir liefern u. a. die Serien:
ACS, Amicus, AmSel, Auge, Bavarian, Cactus, Faug, Fish,
Franz, Kickstart, Panorama, RPD, Ruhr, Saar, Taifun u.v.a.

Unsere Preise inkl. Etiketten

3,5" 2DD No Name	5,25" 2D No Name
1- 50 à 2,50 DM	1- 50 à 1,50 DM
51-100 à 2,40 DM	51-100 à 1,40 DM
ab 101 à 2,30 DM	ab 101 à 1,30 DM

Leerdisketten inkl. Etiketten

3,5" 2DD No Name	5,25" 2D No Name
10- 50 à 10 St. 15,50 DM	10- 50 à 10 St. 8,50 DM
51-100 à 10 St. 14,50 DM	51-100 à 10 St. 7,50 DM
ab 101 à 10 St. 13,50 DM	ab 101 à 10 St. 6,50 DM

Innerhalb **48 Std.** verläßt Ihre Bestellung unser Haus.
Versandkosten 8 DM + NN, Ausland nur Vorkasse + 15 DM

WALTER KAMINSKI

Geilenkirchener Straße 4 - 5110 Alsdorf
Telefon 02404/22963

SUPERPACK 50

50 PD-Programme der Extraklasse!
Return to Earth, Kampf um Eriador, Risk, Broker,
Paranoid, Lucky Loser, Faktura, MS-Text, Videodatei
Plattenliste, Superliga, Haushaltsbuch, MCAD, Wi-
zard of Sound, CLI-Pack, Virus Stop, Werner-Spiel,
Latein, R.O.M., StarTrek, Alf, Core Wars, Label,
Amiga-Paint, Giroman, Blizzard, VirusControl, Te-
trix, Moria, Battleforce, Peters Quest, Super-Bilder,
Billard, Einkommensteuer, DSORT III, Fix-Disk, Uni-
versaldatei, Quickmenü, Diskey, Mandelbrot, Silver
Bilder, Astronomie, Super-Print, Calc, Atlantis,
Schach, Labelpaint! Fast alle mit dt. Anleitungen!
Alle zusammen nur DM 79,-

GRATIS-KATALOGISK

NEU SUPERPACK II (DM 49,-)

Banner, Power Packer, Xytronic, ASDG-RAM, Sonix-
Player u. -Musik, Fußballmanager, Steinschlag, MR-
Backup, Bibel-Quiz, Bootmaster, Boulder 1.3, Roll On,
Paccy, Pente, Tumbler Street, Labelprint, Thundercopy,
Trucking, SchreibM, Hyperadress, Ultra-Paint, Quiz-
master, Pinnball, Database-Wizard, Jazzbench!

PATRICK PAWLOWSKI, Software-Servise
Ellerbruch 19, 2177 Wingst, 04778/7294
Versandkosten: Vorkasse DM 3,50/NN DM 6,00



Wie?... Wir verschenken keine Amigas!
Aber wir haben immer gute Angebote!
Von A 500 bis A 2500, ob Filecard ...
oder PC-AT-Karten oder Zubehör

Was?... Alles für den Amiga bis Professional!
Anrufen lohnt sich!!!

Wo?... PV. Computer-Shop
Bahnhofstr. 278, 8623 Wetzikon (ZH)
Tel. 019 30 79 54
Schweiz

HARDWARE • SOFTWARE • BERATUNG

G N E

5.25" TEAC Profilaufwerk extern 279,-
1,67 MB unformatiert, 880 KB im AMIGAFORMAT, 100% kompatibel zu internem 3,5"-
Diskchange, abschaltbar, 40/80 Track unschaltbar original standardmäßiger durchgeschlit-
fener Bus, Amigatarbenes extrem robustes Stahlblechgehäuse, direkt MS-DOS + PC/XT/AT
Karten kompatibel, mindestens 80 cm langes zugestelltes Rundkabel, Stromversorgung
über Amiga, passender BOOTSELEKTOR im Preis von 279,- enthalten, inklusive WHITE
PROTECT Schalter, qualitätsgespritzt, ausführliche deutsche Bedienungsanleitung

5.25" TEAC intern A2000 239,-
Technische Daten wie 5.25" extern, wahlweise DF1 oder DF2 kostet 259,- wegen
zusätzlicher Modifikationsplatte unkompatibler Einbau innen! 2 Minuten OHNE LÖTEN!!!

3.5" TEAC Profilaufwerk extern 229,-
Technische Daten wie 5.25" Laufwerk extern, White Protect Schalter und Bootselektor im
Preis enthalten!

3.5" TEAC intern A2000 169,-
Technische Daten kompatibel zu 3.5" Laufwerk extern für A500 als DFO, für A2000 als DFO-
DF1, für A1000 als DFO! unkompatibler Einbau innerhalb 2 Minuten OHNE LÖTEN!!!

BOOTSELEKTOR DF1/DF2/DF3 15,-
SOUNDVERTEILER 19,-
A500 512KB + Uhr/abschaltbar 179,-

STAR LC 24-10 Farbband STAR LC 24-10: 9,- DM
STAR XB 24-10 1398,-

NEC P2 PLUS Farbband NEC P2200: 8,50 DM
NEC P6 PLUS Farbband NEC P6/P6+: 9,- DM
Umschaltplatte mit Kick 1.3 98,-

Wir liefern nur deutsche Geräte mit Seriennummer und Super-Service!
1 Jahr Garantie auf alle Produkte dieser Anzeige!

GNE - GREBE NEUMANN ELEKTRONIK

Am Stein 10, 5419 Raubach, 02684-5566/5572

Telex: 869987 Fax: 02684/5448

HÄNDLERANFRAGEN ERWÜNSCHT!!!

DONAU-SOFT

24 h-Schnellversand

Neutrale Disketten

3,5" 2DD (100 % errorfrei)

	von Sentinel	von SONY/Collossus
bis 99 Stück	1,60 DM	2,00 DM
ab 100 Stück	1,40 DM	1,85 DM
ab 500 Stück	1,25 DM	1,70 DM

Laufwerke mit allen Extras

3,5" intern	155,- DM
3,5" extern, abschaltbar, Busdurchführung	209,- DM
5,25" extern, wie 3,5" + 40/80-Trackumschaltung	269,- DM

Sim City (dt.)	79,- DM	GFA-Basic 3.5	208,- DM
B.A.D.	77,- DM	GFA-Compiler 3.5	129,- DM
DPaint III dt.	240,- DM	512 KB-Erw. (A500)	199,- DM
Zoetrope 1.1	189,- DM	2 MB-Erw. (A500)	598,- DM

Vorkasse: + 5,- DM, Nachnahme: + 8,- DM, Ausland: + 10,- DM

MAIK HAUER

Postfach 14 01, 8858 Neuburg Fax: 0 84 31/4 98 00
Tel.: 0 84 31/4 97 98 (bis 22 Uhr) BTX: "Donau-Soft"

MacSoft - AMIGA SHOP
Hardware - Software - Schulung - PD

Public Domain
DISKETTE
AUF 2 DD NUR **4,- DM**

Über 4500 PD-Disk! Immer aktuell!
24-Std.-Versand-Service
Katalog-Disketten anfordern, 5,- DM
Selber abholen, NN gespart!
Hardware-Zusammenstellung auf Wunsch.
Lassen Sie sich Ihren
persönlichen Amiga anfertigen.
Fragen Sie nach unseren
Amiga-Einsteiger-Kursen.

Telefon 02 31 / 51 60 10

Mo.-Fr. 10-13, 15-20 Uhr, Sa. 10-16 Uhr

Hannövrstr. 82, 4600 Dortmund 1

BTX * mac soft amiga #

AMIGA PD

Riesiges Angebot ausgesuchter PD Software

- Jedes Programm mit DEUTSCHER ANLEITUNG
- SUPER SICHER - Alles 100 % Virentfrei

zum Sonderpreis ab 1,-

Immer auf dem NEUESTEN STAND
● Zu jeder AMIGA-AUSGABE die getesteten
PD PROGRAMME auf DREI 3,5" Disketten
in einem Monatlichen ABO zusammengefaßt für:
DM 19,90 per Bankeinzug DM 22,90 per Nachnahme

Jetzt einmalig für Sie zum Schnäpppreis 9,90

- Fordern Sie unseren umfangreichen Katalog für nur DM 3,- an.

Hotline 0781 - 57734

von 18.00 - 20.00 Uhr

GERMAN · SOFT

PD - UTE - URLBAUER

Abt. C-64/C-128/AMIGA PD

Am Rothweg 9 · D-7600 Offenburg 18

Tel. 0781 / 57734



Ihre Hotline für AMIGA-DOS-Tip

Telefon
0 89 / 4 39 10 87 bis -89

Telefax
0 89 / 4 39 10 80

AMIGA D.E.L.T.A. - S.O.F.T. AMIGA C64/128 PC PC C64/128

Software - Hardware - Public Domain
Frank Krüger · Postfach 45 00 23 · 5305 Alfter 1

Public-Domain-Disketten

Amiga	über 6000 Stck.	ab DM 2,18
PC	über 1500 Stck.	ab DM 2,25
C64/128	über 200 Stck.	ab DM 1,95
PD-ABO	jeden Monat 10 aktuelle Disks	nur DM 29,00

Wir führen alle Serien - 24 Std. Versandservice

Vokabeltrainer (Amiga): Englisch, Französisch,
Latein, Italienisch, Spanisch
Je Disk mit ca. 5000 Vokabeln.....DM 49,-

Weitere Top-Programme auf Lager !!

Versand per NN (+ 6,- DM) oder Scheck (+ 2,- DM)
Ausführlicher Katalog + 2 PD-Disketten nur DM 5,-!
4 PD-Katalogdisketten gegen DM 4,- Rückporto!
Programmankauf - Händleranfragen sehr erwünscht

H O T L I N E: (02 28) 66 50 38

Commodore® Ersatzteil Service

✕ Wir liefern
für **Händler** und Privat-
anwender preiswert und prompt

✕ Rufen Sie uns an: (02331-43001)
oder schreiben Sie uns:

CIK-Computertechnik • Ingo Klepsch
Berliner Straße 49b • D-5800 Hagen 7

TELEFAX: 02331-42499

Haben Sie Fragen zum Thema AMIGA und AMIGA DOS?

Dann rufen Sie doch einfach mal an!

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:

Markus Matejka:
05651/809-741

Jürgen Borngießer:
05651/809-742

Vera Brinkmann:
05651/809-743

Claus Daschner:
05651/809-744

Thomas Baum:
05651/809-740

Wir freuen uns auf Ihren Anruf!

Ihr AMIGA-DOS-TEAM



Alle Programmaktionen leben von der Kommunikation mit dem Benutzer. Immer, wenn es darum geht, Entscheidungen zu treffen oder einfach nur Informationen zu übermitteln, kommen uns Dialogboxen (Requester) zu Hilfe. Jeder Amiga-Benutzer kennt seit seinen ersten Computer-Tagen die System-Requester der Workbench, die beispielsweise zum Einlegen einer bestimmten Diskette auffordern. Deutlich abgesetzt von der übrigen Arbeitsumgebung zeigen sie an, wenn etwas Bestimmtes zu tun ist und daß es gleich getan werden muß, bevor die eigentliche Arbeit weitergehen kann. Die explizite Eingabe eines Kommandos über die Tastatur wäre in diesem Fall zwar genauso gut möglich, erscheint aber bei Computern mit den Leistungsmerkmalen eines Amiga nicht mehr zeitgemäß. Mit der Maus können alle Entscheidungen wesentlich schneller, übersichtlicher und bequemer übermittelt werden.

Die Dialogbox

Diesen Bedienungskomfort bieten Intuition-Requester. Sie erscheinen auch immer dann, wenn der Benutzer Gefahr läuft, unbeabsichtigt einen nicht wiedergutzumachenden Befehl auszulösen, und fragen nach, ob es auch wirklich "ernst gemeint" war; "gefährliche" Befehle können so noch rechtzeitig verhindert werden. Zählen Sie doch mal nach, wie viele Requester dieser Art mit den Optionen »OK« und »CANCEL« in einem komplexen Programm auftreten können! Jetzt stellen Sie sich mal vor, wieviel Schaden auf einmal ein Benutzer anrichten könnte, wenn die Requester plötzlich nicht mehr erscheinen würden, sondern alle Befehle sofort zur Ausführung kämen. Sicherlich würde er bald entnervt den Spaß an Programm und Computer verlieren.

Sie sollten daher auch in Ihren eigenen Programmen an kritischen Stellen Requester zur Sicherheitsabfrage vorsehen, um auf diese Weise Benutzerfrust von vornherein zu vermeiden. Aber bitte immer nach dem Motto "So wenig wie möglich, so viel wie nötig", denn eine ständige und übermäßige Requester-Nachfrage bei allen möglichen Be-

Franz-Josef Reichert

Vielseitige Requester für alle Fälle

Grundlagen der Library-Programmierung

Nach unserem File-Requester in der letzten Ausgabe geht's heute mit der Standarddialogbox und der Farbeinstellungspalette als Bereicherung für Ihre selbstgeschriebenen Programme weiter.

fehlen kann die Arbeitsgeschwindigkeit und damit auch die Arbeitsfreude erheblich bremsen. Denken Sie immer daran, daß der Benutzer vorrangig seine Arbeit in der kürzest möglichen Zeit erledigen will.

Unser Library-Requester ist ein gelungener Kompromiß zwischen geforderter Vielseitigkeit und dem Ziel, den Aufruf so einfach wie möglich zu halten. Sie können ihn in jedem beliebigen Window erscheinen lassen und (fast) beliebig viele Zeilen Text in einer beliebigen Schriftart (Font) übergeben. Die Requester-Dimensionen passen sich automatisch an Fenstergröße, Schriftart und Zeilenlänge an. Ihr Text wird in Schattenschrift auf einer frei wählbaren Untergrundfarbe dargestellt und automatisch zentriert. Zur Entscheidungsabfrage haben Sie die Möglichkeit, bis zu drei Gadgets mit beliebigem Text zu erzeugen. Wenn Sie nur Text für ein Gadget übergeben, erscheint dieser in der rechten unteren Ecke. Dies ist vor allem bei reinen Informations-Requestern nützlich, die nur eine Mitteilung für den Benutzer enthalten und ihm keine Entscheidung abverlangen. Wenn Sie zwei Gadgets definieren, erscheint der erste in der linken unteren Ecke und der zweite in der rechten unteren Ecke. Damit können beispielsweise Entscheidungs-Requester für »JA« oder »NEIN«, »OK« oder »CANCEL« definiert werden. Für komplexere Abfragen können Sie auch drei Gadgets definieren, wobei das dritte in der Mitte erscheint, beispielsweise für die Alternativen »JA«, »NEIN« und »ABBRUCH« auf die Frage hin, ob vor Verlassen des Programms noch Projekte gesichert werden sollen.

Halten Sie sich bei der Gadget-Definition aber bitte stets an die einschlägigen Programmierrichtlinien, wonach zwischen einer "guten" und einer "bösen" Seite unterschieden wird (siehe [1], Seite 211 ff.). Auf der rechten Seite sollten demnach nur Gadgets erscheinen, deren Betätigung keine unerwarteten Programmaktionen auslösen, das heißt, sie führen den noch unerfahrenen Benutzer stets wieder an den Ausgangspunkt zurück, ohne Schaden anzurichten. Die linke Seite dagegen bleibt der Auslösung der "gefährlichen" Befehle vorbehalten. Wer hier ein Gadget aktiviert, der sollte genau wissen, was er tut. Bei Beachtung dieser Vorgaben kann der Erstbenutzer sich nach der Methode "Versuch und Irrtum" mit dem Programm vertraut machen und eigentlich nie einen schwerwiegenden Fehler begehen. Auch wenn er die Requester-Meldungen noch nicht vollständig versteht, so ist seine Annahme immer richtig, mit Betätigung des rechten Gadgets keinen Schaden anrichten zu können. Mit den vielfältigen Möglichkeiten von »BuildBoolRequester()« sollten Sie ab heute für die gängigsten Requester-Probleme gewappnet sein. Aufrufen wird unsere Funktion mit folgenden Parametern:

```
result = BuildBoolRequester(backfill, textpen, shadow, lines, es, ta, window, text, gadgets)
```

Parameter:

```
ULONG backfill, textpen, shadow, lines;
struct TextAttr *ta;
struct Window *window;
UBYTE *text[], *gadgets[];
```

Rückgabewert:
ULONG result;

Die Bedeutung der Parameter im einzelnen:

backfill - Die Füllfarbe (Grundfarbe) des Requesters.

textpen - Textfarbe für die Schrift des Requesters.

shadow - Textfarbe für Schattenschrift. Wenn keine Schattenschrift gewünscht wird, kann dieser Parameter den gleichen Wert wie »backfill« erhalten.

lines - Anzahl der Textlinien im Stringpointer-Array

ta - Zeiger auf eine Struktur »TextAttr« zur Definition des Schriftsatzes (font). Wird dieser Wert auf null gesetzt, so wird die voreingestellte Schriftart des Window-Screens akzeptiert.

window - Ein Zeiger auf die Intuition-Window-Struktur des Fensters, in dem der Requester erscheinen soll.

text - Ein Zeiger auf ein Stringpointer-Array mit beliebig vielen Einträgen. Die Einträge zeigen auf die erste bis letzte Zeile des Requester-Texts.

gadget - Ein Zeiger auf ein Stringpointer-Array mit drei Einträgen. Die Einträge zeigen auf den Text für die drei möglichen Gadgets in der Reihenfolge von links nach rechts. Soll das entsprechende Gadget unterdrückt werden, ist "NULL" einzutragen.

Die Bedeutung des Rückgabewertes:

result = -1

- Beim Aufruf trat ein Fehler auf. Dies kann seinen Grund in mangelndem Speicherplatz oder falschen Aufrufparametern haben.

result = 0

- Das linke Gadget wurde betätigt.

result = 1

- Das mittlere Gadget wurde betätigt.

result = 2

- Das rechte Gadget wurde betätigt.

Die komplette Funktion »BuildFilerequester()« finden Sie in Listing 2. Zum Testen haben wir ein kleines Beispielprogramm »Minex.c« abgedruckt (Listing 4), das ein Window auf der Workbench eröffnet und mit einem Mausklick die Dialogbox erscheinen läßt. Die Entscheidungen werden im CLI-Window protokolliert. Compilieren und Linken können es die C-Programmierer unter Ihnen mit den Aufrufen

```
lc -v -w -O buildboolrequester
lc -v -w -cf -O -lnt+global.o
+buildboolrequester.o minex
```


zum ausführbaren Programm. Erinnern Sie sich noch an »global.c« (Listing 3) aus dem vorigen Teil unseres Artikels? Das Compilat »global.o« muß hier beim Linken mit angegeben werden, da wir Funktionen daraus in unserem Programm benötigen. Falls Sie die Include-Datei »requester.h« vermissen sollten, auch sie wurde bereits im ersten Teil als Listing 4 abgedruckt.

Zwei Funktionen unserer zukünftigen Library haben wir damit abgehandelt. Wenden wir uns nun der dritten und letzten Routine zu. Gerade beim Grafik-Computer Amiga kommt der Farbauswahl große Bedeutung zu.

Die Farbpalette

Nicht nur bei Mal- und Zeichenprogrammen spielt die Farbpalette eine tragende Rolle, obwohl hier die Nützlichkeit am ehesten deutlich wird. Wer sich intensiv mit Anwenderprogrammen befaßt, die einen genaueren und längeren Blick auf den Bildschirm erfordern, der wird hier und da die Möglichkeit vermissen haben, die Farben der Arbeitsumgebung zu verändern. Denn manchmal ist die Farbwahl dermaßen grell ausgefallen, daß sich bei längerer Arbeit Ermüdungserscheinungen einstellen können. Es bleibt als letzter Ausweg, an den Helligkeits- und Kontrastregeln des Monitors "herumzumursen", wobei eine befriedigende Einstellung auch hier nur in den seltensten Fällen gefunden werden kann. Deshalb sollten Sie in Ihren Programmen immer die Möglichkeit vorsehen, dem Benutzer die Farbauswahl zu über-

lassen. Für alle, die sich in diesem Zusammenhang bereits geärgert haben, haben wir noch eine kleine Überraschung parat, doch dazu später mehr...

Bei der Gestaltung unserer Farbpalette wurde besonderer Wert auf Vielseitigkeit gelegt. Sie sollte sowohl bei allen 32 zur Verfügung stehenden Farben als auch bei nur zweien ein ansprechendes Bild und handliche Bedienungselemente zeigen. Außerdem muß sie für alle Screen-Auflösungen und -Betriebsarten brauchbar sein, ohne daß dabei die Bedienungselemente verzerrt werden, und im Screen frei beweglich bleiben, ohne wichtige Teile der Arbeitsumgebung zu verdecken. Um diese Ziele zu erreichen, wurde sie als Window konzipiert, kann daher beliebig herumgeschoben und über die Tiefen-Gadgets in Vorder- oder Hintergrund geschaltet werden. Sie paßt sich den Screen-Dimensionen und -Farbtiefen an und ist somit für alle Betriebsarten, mit Ausnahme des HAM-Modus, zu gebrauchen. Die Farben werden im unteren Auswahlfeld durch einfaches Anklicken selektiert, wonach sie umrandet erscheinen. Mit drei Schieberglern können jetzt die Rot-, Grün- und Blauanteile in bekannter Manier festgelegt werden. Dabei haben Sie in einem gesonderten Feld stets die Kontrolle über die genaue Zusammensetzung in hexadezimaler Schreibweise. Mit den Bedienungsgadgets »OK«, »UNDO« und »QUIT« können Sie die Farbeinstellung sichern, die letzte Veränderung wieder rückgängig machen oder aber die Palette so verlassen, als wäre nichts geschehen.

Alles, was Sie in Zukunft zur Verwirklichung dieses Bedienungskomforts für Ihre eigenen Programme brauchen, sind die Funktion »BuildColorPalette()« und drei Aufrufparameter. Was könnte einfacher sein?

```
colormap = BuildColorPalette(
    cmap, screen, title)
```

Parameter:

```
UWORD *cmap;
struct Screen *screen;
UBYTE *title;
```

Rückgabewert:

```
UWORD *colormap;
```

Die Bedeutung der Parameter im einzelnen:

cmap – Ein Zeiger auf ein reserviertes UWORD-Array mit 32 Einträgen. Hier werden von der Routine die Farbwerte gespeichert beziehungsweise zurückgegeben.

screen – Ein Zeiger auf die Intuition-Screen-Struktur, in der das Fenster geöffnet werden soll.

title – Dieser String wird in die Titelzeile des Fensters geschrieben.

Die Bedeutung des Rückgabewertes:

colormap = -1

– Beim Aufruf trat ein Fehler auf. Dies kann seinen Grund in mangelndem Speicherplatz oder falschen Aufrufparametern haben.

colormap = 0

– Der Benutzer hat das Paletten-Window durch Betätigung des Gadgets »Quit« beendet.

colormap > 0

– Der Benutzer hat »OK« betätigt. »colormap« zeigt auf ein UWORD-Array mit 32 Einträgen nach einem Muster, aus dem die neuen Farbwerte übernommen werden können: Wie dieses Muster aussieht, können Sie in der Abbildung erkennen.

Zur Demonstration der Farbpaletten-Routine gibt es ein praktisches Beispielprogramm »Palette.c« (Listing 3) zum Abtippen, das Ihnen die Farbeinstellung überall dort ermöglicht, wo sie bisher nicht vorgesehen war. Sie haben jederzeit die Möglichkeit, in jedem Screen die Farben nach Ihren Vorstellungen zu verändern und diese Änderung auch abzuspeichern. Das Programm läßt sich beim Start in den Speicher, verharnt dort im Hintergrund und wird erst dann aktiv, wenn es durch eine Hotkey-Sequenz, der gleichzeitigen Betätigung von [AMIGA-LEFT] und [F5], aus seinem Ruhezustand aufgeweckt wird. Mit dem Bedienungsgadget »OK« wird die aktuelle Farbeinstellung gespeichert, »UNDO« macht sie rückgängig, »QUIT« verläßt das Programm, ohne die vorgenommenen Veränderungen zu übernehmen. Wenn Sie es also geschafft haben, derart gräßliche Farben zu erzeugen, daß Sie wirklich gar nicht mehr hinsehen können, betätigen Sie einfach »QUIT« und alles ist wieder wie vorher. Danach tritt »Palette« wieder in den Hintergrund. Endgültig beenden können Sie »Palette« durch Betätigung von [AMIGA-LEFT] + [SHIFT] + [F5].

Es schadet übrigens nichts, wenn das Programm versehentlich mehrfach aufgerufen wird. Jedesmal, wenn »Palette« gestartet wird, prüft es selbständig nach, ob bereits ein Palette-Task aktiv ist. In diesem Fall veranlaßt es die Beendigung des anderen Tasks und installiert sich selbst, wodurch immer nur ein Palette-Task aktiv sein kann, nämlich der zuletzt ge-

Ist Ihr Programm der HIT?

Der DMV-Verlag sucht ständig nach neuer, interessanter Software zur Aufnahme in unser Software-Sortiment.

Dabei ist es einerlei, ob Sie nun ein Anwendungs- oder ein Spielprogramm geschrieben haben.

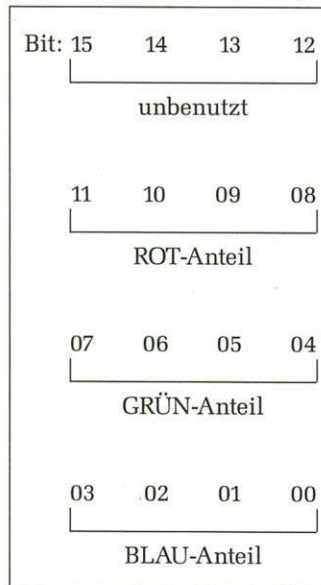
Der DMV-Verlag bietet Ihnen sein Software-Know-how an!

startete. Sie können »Palette« sowohl von der Workbench als auch vom CLI aus starten. In beiden Fällen haben Sie außerdem die Möglichkeit, es mit einem Parameter aufzurufen. Hier kann der Name eines Files angegeben werden, in dem dann eine gewählte Farbeinstellung als ausführbares Programm gespeichert wird. Geben Sie also im CLI »1> Palette testfile« ein, so wird die aktuelle Farbeinstellung beim Betätigen des »OK«-Gadgets jedesmal in »testfile« abgelegt. Zur Erinnerung erscheint im Palette-Window der Name »testfile« in der Titelzeile.

Das Programm »testfile« können Sie jetzt aus dem CLI starten oder auch von der Workbench aus, wenn Sie ihm ein Tool-Icon verschaffen. Hierzu kann »Palette«, wenn es von der Workbench aus gestartet werden soll, der Aufrufparameter in Form eines anderen Tool-Icons übergeben werden. Wählen Sie das Palette-Icon an, drücken Sie [Shift] und halten dies gedrückt, während Sie das gewünschte Tool-Icon mit Doppelklick anwählen. Rufen Sie nun die Farbpalette mit der Hotkey-Sequenz auf, und stellen Sie die Farben nach Ihrem Geschmack ein. Betätigen Sie danach das Gadget »OK«. Zur Reproduktion der Farbeinstellung starten Sie einfach Ihr Arbeitsprogramm, schalten Sie zur Workbench zurück ([AMIGA-LEFT] und

[N] beziehungsweise [M]), und klicken Sie zweimal auf das Tool-Icon. Der zuletzt geöffnete Screen wird nun die gewünschte Farbeinstellung aufweisen. Leider halten sich nicht alle für den Amiga erhältlichen Programme an die Vorgaben, die für den reibungslosen Ablauf des Multitaskings eingehalten werden müssen. Bei sehr vielen Programmen, besonders häufig bei Spielen, wird zugunsten bestimmter Effekte, aufgrund von Kopierschutzmaßnahmen oder einfach nur wegen schlampiger Programmierung das Multitasking unmöglich gemacht. Hier ist es dann leider auch nicht möglich, »Palette« zu benutzen. Bei allen anderen Programmen sollte jetzt aber der Ärger über Farben und deren Einstellung ein Ende haben. Nicht zuletzt ist die Farbgebung auch Geschmackssache, und über Geschmack läßt sich bekanntlich streiten.

Im Programmlisting haben wir schon einen Vorgriff gemacht. Sie können es sowohl als Stand-alone-Modul wie auch für die Verwendung zusammen mit unserer »shared-library« compilieren, die wir in der nächsten Folge aus unseren drei Routinen zusammenbauen werden. Für heute wollen wir die erste Möglichkeit realisieren und mit dem Aufruf »lc -v -w -O buildcolorpalette« zunächst die Funktion »BuildColorPalette()« (Li-



sting 1) compilieren und mit
 lc -v -w -cf -O -t -
 Lnt+global.o+buildcolorpalette.o palette
 zum lauffähigen Programm »Palette« gelangen.
 Damit ist unsere Library fast komplett! Mit diesen drei Funktionen haben Sie die Möglichkeit, Ihre Programme mit einem ansprechenden und durchdachten Benutzer-Interface auszustatten. Durch die konsequente Verwendung einheitlicher Bedienungselemente werden dem Benutzer alle Programme vertrauter, wird die Einarbeitungszeit verkürzt und die Lernbereitschaft gesteigert. Dies trägt er-

heblich zur vielgepriesenen (und leider genauso oft auch verfehlten) »Benutzerfreundlichkeit« bei. Wer Software für den Amiga entwickeln möchte, sollte dieses Thema nicht stiefmütterlich behandeln und sich ernsthaft Gedanken darüber machen. Es sollte nicht dem Zufall überlassen werden, ob bestimmte Bedienungselemente, die gleichartige Operationen auslösen, sich auch bei völlig unterschiedlichen Programmen stets an der gleichen Stelle befinden und auch gleich aussehen. Wichtige Bedienungselemente stehen Ihnen ja jetzt zur Verfügung.

Wann fangen Sie an zu programmieren?
 Etwas gedulden müssen sich bis zur nächsten Folge noch diejenigen, die nicht in C programmieren. Dann wollen wir, wie versprochen, unsere Funktionen für alle Programmiersprachen über eine echte Amiga-Library verfügbar machen. Ganz nebenbei können Sie noch eine Menge über den allgemeinen Aufbau und die Technik der Library-Programmierung lernen.

Hier noch ein Hinweis zur Literatur:

[1] Commodore-Amiga, Inc.: Amiga ROM Kernel Reference Manual: Libraries and Devices, Revised and Updated; 1. Auflage, Addison-Wesley 1989, ISBN 0-201-18187-8.

(jb)

Listings

```

1: /*=====
2: *==
3: *==          BuildBoolRequester.c
4: *==
5: *==          by Franz-Josef Reichert
6: *==
7: *==          Lattice C-Compiler V 5.04
8: *==
9: *=====*/
10:
11: #include <clib/macros.h>
12: #include <exec/types.h>
13: #include <exec/memory.h>
14: #include <intuition/intuition.h>
15: #include <proto/exec.h>
16: #include <proto/intuition.h>
17: #include "requester.h"
18:
19: #define GADGSPACE 6
20:
21: static struct TextLine {
22:   struct IntuiText tl_front,tl_back;
23: };
24:
25: static struct DATA {
26:   struct Requester req;
27:   struct Gadget gadg[3];
28:   struct Border gadgborder,reqborder;
29:   struct Remember *rkey;
30:   struct IntuiText gadgtext[3],gadgbgtext[3];
31:   WORD rvec[16],gvec[16];
32: };
33:
34: extern VOID CenterIText
35: (struct IntuiText*,WORD,WORD,WORD,WORD);
36: extern VOID InitBorder(WORD*,WORD,WORD,WORD,WORD);
  
```

Listing 1

```

37:
38: extern LONG _saves BuildBoolRequester(LONG backfill,
39: LONG textpen, LONG shade, LONG lines, struct TextAttr *ta,
40: struct Window *w, UBYTE *t[], UBYTE *g[]) {
41:   LONG result = -1;
42:   ULONG class, OldIDCMPFlags;
43:   struct MsgPort *OldPort;
44:   struct IntuiMessage *msg;
45:   struct DATA *gd;
46:   WORD height,width,i;
47:   struct TextLine *tl;
48:   struct IntuiText *it,*last = NULL;
49:   if(t != NULL && g != NULL && w != NULL) {
50:     if(ta == NULL) ta = w->WScreen->Font;
51:     OldIDCMPFlags = w->IDCMPFlags;
52:     OldPort = w->UserPort;
53:     width = w->WScreen->Width >> 1;
54:     if(gd = (struct DATA*)AllocMem
55: (sizeof(struct DATA),MEMF_PUBLIC|MEMF_CLEAR)) {
56:       for(i = 0; i < lines; i++) {
57:         if(tl = (struct TextLine*)AllocRemember(&gd->rkey,
58: sizeof(struct TextLine),MEMF_PUBLIC|MEMF_CLEAR)) {
59:           tl->tl_front.FrontPen = textpen;
60:           tl->tl_back.FrontPen = shade;
61:           tl->tl_front.ITextFont =
62: tl->tl_back.ITextFont = ta;
63:           tl->tl_back.NextText = &tl->tl_front;
64:           tl->tl_front.NextText = last;
65:           tl->tl_front.IText = tl->tl_back.IText = t[i];
66:           width = MAX(4 * GADGSPACE + IntuiTextLength
67: (&tl->tl_front),width);
68:           tl->tl_front.TopEdge = tl->tl_back.TopEdge =
69: (WORD)(GADGSPACE + i*(GADGSPACE + ta->ta_ysize));
70:           tl->tl_back.LeftEdge = 2;
71:           tl->tl_back.TopEdge += 1;
72:           last = &tl->tl_back;
73:         } else break;
74:       }
75:       for(it = last; it; it = it->NextText) {
  
```

Listing 1


```

76:   CenterIText(it,it->LeftEdge,it->TopEdge,
77:   width,(WORD)(GADGSPACE + ta->ta_ysize));
78: }
79: height = (lines + 1) * (ta->ta_ysize +
80: GADGSPACE) + 4 * GADGSPACE;
81: width = MIN(w->width,width);
82: height = MIN(w->height,height);
83: for(i = 0; i < 3; i++) {
84:   gd->gadgtext[i].FrontPen = textpen;
85:   gd->gadgbgtext[i].FrontPen = shade;
86:   gd->gadgtext[i].ITextFont =
87:   gd->gadgbgtext[i].ITextFont = ta;
88:   gd->gadgtext[i].IText =
89:   gd->gadgbgtext[i].IText = g[i];
90:   gd->gadgbgtext[i].NextText = &gd->gadgtext[i];
91:   gd->gadg[i].Width = (width - 8 * GADGSPACE) / 3;
92:   gd->gadg[i].LeftEdge = 2 * GADGSPACE + i *
93:   (gd->gadg[i].Width + 2 * GADGSPACE);
94:   gd->gadg[i].Height = ta->ta_ysize + 2 * GADGSPACE;
95:   gd->gadg[i].TopEdge = height -
96:   gd->gadg[i].Height - GADGSPACE;
97:   gd->gadg[i].GadgetType = BOOLGADGET|REQGADGET;
98:   gd->gadg[i].Activation = ENDGADGET|RELVERIFY;
99:   gd->gadg[i].GadgetRender = (APTR)&gd->gadgborder;
100:  gd->gadg[i].GadgetText = &gd->gadgbgtext[i];
101:  gd->gadg[i].NextGadget = (i) ?
102:  (&gd->gadg[i - 1]) : (NULL);
103:  gd->gadg[i].GadgetID = i;
104:  CenterIText(&gd->gadgtext[i],2,1,
105:  gd->gadg[i].Width,gd->gadg[i].Height);
106:  CenterIText(&gd->gadgbgtext[i],0,2,
107:  gd->gadg[i].Width,gd->gadg[i].Height);
108: }
109: InitBorder(gd->rvec,0,0,
110: (WORD)(width - 2),(WORD)(height - 1));
111: InitBorder(gd->gvec,-2,-1,
112: gd->gadg[0].Width,gd->gadg[0].Height);
113: gd->reqborder.FrontPen = 1;
114: gd->gadgborder.FrontPen = 3;
115: gd->reqborder.Count = gd->gadgborder.Count = 8;
116: gd->reqborder.XY = gd->rvec;
117: gd->gadgborder.XY = gd->gvec;
118: gd->req.LeftEdge = MAX(0,w->width - width >> 1);
119: gd->req.TopEdge = MAX(0,w->height - height >> 1);
120: gd->req.Width = width;
121: gd->req.Height = height;
122: gd->req.RegGadget = &gd->gadg[2];
123: gd->req.RegBorder = &gd->reqborder;
124: gd->req.RegText = last;
125: gd->req.BackFill = backfill;
126: if(g[1] == NULL) {
127:   gd->gadg[2].NextGadget = &gd->gadg[0];
128:   if(g[0] == NULL) gd->gadg[2].NextGadget = NULL;
129: }
130: w->UserPort = NULL;
131: ModifyIDCMP(w,GADGETUP);
132: if(Request(&gd->req,w)) {
133:   FOREVER {
134:     WaitPort(w->UserPort);
135:     msg = (struct IntuiMessage*)GetMsg(w->UserPort);
136:     class = msg->Class;
137:     result = ((struct Gadget*)
138:     msg->IAddress->GadgetID);
139:     ReplyMsg((struct Message*)msg);
140:     if(class == GADGETUP) break;
141:   }
142: }
143: FreeMem((UBYTE*)gd,sizeof(struct DATA));
144: FreeRemember(&gd->rkey,TRUE);
145: ModifyIDCMP(w,NULL);
146: w->UserPort = OldPort;
147: ModifyIDCMP(w,OldIDCMPFlags);
148: }
149: }
150: return(result);
151: }

```

Listing 1

```

1: /*=====
2: **
3: **      BuildColorpalette.c
4: **
5: **      by Franz-Josef Reichert
6: **
7: **      Lattice C-Compiler V 5.04
8: **
9: **=====*/
10:
11: #include <clib/macros.h>
12: #include <exec/types.h>
13: #include <exec/memory.h>
14: #include <graphics/gfx.h>
15: #include <graphics/gfxbase.h>
16: #include <intuition/intuitionbase.h>
17: #include <proto/exec.h>
18: #include <proto/graphics.h>
19: #include <proto/intuition.h>
20: #include "requester.h"
21:
22: #define FWIDTH      2
23: #define FHEIGHT     22
24: #define MAXCOLORS   32
25: #define WINDOWSPARE 8
26: #define GADGTOP      4
27: #define GADGLEFT     6
28: #define GADGSPACE    4
29: #define GADGGAP      8
30: #define PGADGFLAGS   GADGHCOMP|GADGIMAGE
31: #define PGADGACT     GADGIMMEDIATE|RELVERIFY|FOLLOWMOUSE
32: #define RGBMAXVAL    16L
33:
34: static struct DATA {
35:   struct Image rI,gI,bI,sI;
36:   struct PropInfo piR,piG,piB;
37:   struct Window *pw;
38:   struct NewWindow pnw;
39:   struct ViewPort *vp;
40:   struct Gadget *cgad[MAXCOLORS],pgR,pgG,pgB;
41:   struct Gadget bgOK,bgUNDO,bgQUIT;
42:   struct Image *ci[MAXCOLORS];
43:   struct Remember *rkey;
44:   struct Border gadB,cgadB,siB,selB;
45:   struct IntuiText OKtxt,UNDOtxt,QUITtxt,hextxt;
46:   UWORD *colormap,buf[3],undobuf[3];
47:   WORD gpoints[16],ipoints[16],spoints[16];
48:   UBYTE maxcolors,color,hexbuf[4];
49: };
50:
51: extern VOID InitBorder(WORD*,WORD,WORD,WORD,WORD);
52: extern VOID CenterIText
53: (struct IntuiText*,WORD,WORD,WORD,WORD);
54: extern struct TextAttr taPlain;
55: static VOID _regargs Cleanup(UBYTE,struct DATA*);
56: static VOID _regargs SelectGadget
57: (UBYTE,UBYTE,struct DATA*);
58: static VOID _regargs BackupColors(struct DATA*);
59: static VOID _regargs GetColor
60: (UBYTE,UWORD*,struct DATA*);
61: static VOID _regargs PrintHexVal(struct DATA*);
62: static VOID _regargs SetColor(UBYTE,struct DATA*);
63: static VOID _regargs SetGadget
64: (struct Gadget*,UWORD,struct DATA*);
65: static VOID _regargs UpdateGadgets
66: (UBYTE,UWORD*,struct DATA*);
67: static struct DATA _regargs *InitGlobal(VOID);
68: static UBYTE *const hexTab = "0123456789ABCDEF";
69: static UBYTE *const gdtxt[] = { "OK","UNDO","QUIT" };
70:
71: static struct DATA _regargs *InitGlobal() {
72:   struct DATA *gd;
73:   if(gd = (struct DATA*)AllocMem
74:   (sizeof(struct DATA),MEMF_PUBLIC|MEMF_CLEAR)) {
75:     gd->gadB.Count = gd->cgadB.Count = 8;
76:     gd->siB.Count = gd->selB.Count = 8;
77:     gd->gadB.FrontPen = gd->cgadB.FrontPen =

```

Listing 2



1180 Wien, Schulgasse 63
Tel: (0 222) 408 52 56
Telefax: (0 222) 408 99 78
Postversand - Teilzahlung - Leasing

Amiga 500 Speichererweiterung 512 KB + Uhr	öS 1.490,- (DM 213,-)
Amiga 5.25 Zoll Laufwerk, Bus/Ein- u. Ausschalter, 40/80 Tr.	öS 1.990,- (DM 285,-)
Amiga 2000, 2 MB Memory	öS 4.990,- (DM 713,-)
Amiga 2000 XT-Karte inkl. Laufwerk	öS 5.990,- (DM 856,-)
Amiga 2000 XT-Turbo mit 8 MHz	öS 1.590,- (DM 227,-)
80 MB SCSI Harddisk / 24 ms, für A 500 komplett	öS 15.990,- (DM 2285,-)
Amiga Ersatzmaus	öS 590,- (DM 85,-)
Control-Center 500 (Top-Styling!)	öS 1.490,- (DM 213,-)
Digi View 4.0 Gold	öS 2.690,- (DM 385,-)
Genlock Pal Ver 1.3	öS 4.990,- (DM 713,-)

Eurosystems (Midi-Manager, Pro Sampler, Syncro Express, Handyscanner) lagernd
GVP (45 MB 28 ms, 40 MB 19 ms, 80 MB 19 ms, 68030 Karte,...) lagernd



H.Rodat J.Haas M.Kiel

Der Computeriaden Ihr AMIGA Spezialist in Ostwestfalen!

Festplattensysteme mit dem neuen TRUMPCARD - Controller
(getestet in AMIGA 01/02 90, mit Harddiskunterstützung für A-MAX)
TRUMPCARD A2000 SCSI-Controller einzeln **DM 498,-**
(Superschnell bis 500 KB/sec, Autoboot etc.)
als Filecard mit SEAGATE ST 157N (46 MB) **DM 1328,-**
oder Vollgas mit QUANTUM P40 S (40 MB) **DM 1598,-**
NEU für AMIGA 500 im Gehäuse, mit Steckplatz für Speichererweiterung max 4 MB
TRUMPCARD 500 46 MB komplett **DM 1398,-**
dto. natürlich auch mit QUANTUM P40S 40 MB **DM 1648,-**
Knaller: Hurricane für A500 mit 68020 / 16 MHz **DM 1098,-**
Hurricane H2800 68030/28 MHz f. A2000 **DM 2398,-**
Speichererweiterung A500 intern mit 2 MB best **DM 649,-**

Amiga 3000 auf Anfrage
Weitere aktuelle Angebote finden Sie in unserer Preisliste, die wir Ihnen gerne zusenden.
CompiMate Computer, Sudbrackstr. 31, 4800 Bielefeld 1, Tel. 05 21-13 36 21 / FAX 12 43 33


```

78:  gd->siB.FrontPen = 1;
79:  gd->gadB.XY = gd->siB.XY = gd->ipoints;
80:  gd->cgadB.XY = gd->gpoints;
81:  gd->selB.XY = gd->spoints;
82:  gd->siB.NextBorder = &gd->cgadB;
83:  gd->pgG.NextGadget = &gd->pgR;
84:  gd->pgB.NextGadget = &gd->pgG;
85:  gd->pgR.LeftEdge = gd->pgG.LeftEdge =
86:  gd->pgB.LeftEdge = GADGLEFT - 2;
87:  gd->pgR.TopEdge = GADGTOP;
88:  gd->pgR.Flags = gd->pgG.Flags =
89:  gd->pgB.Flags = PGADGFLAGS;
90:  gd->pgR.Activation = gd->pgG.Activation =
91:  gd->pgB.Activation = PGADGACT;
92:  gd->pgR.GadgetType = gd->pgG.GadgetType =
93:  gd->pgB.GadgetType = PROPGADGET;
94:  gd->pgR.GadgetRender = (APTR)&gd->rI;
95:  gd->pgG.GadgetRender = (APTR)&gd->gI;
96:  gd->pgB.GadgetRender = (APTR)&gd->bI;
97:  gd->pgR.SpecialInfo = (APTR)&gd->piR;
98:  gd->pgG.SpecialInfo = (APTR)&gd->piG;
99:  gd->pgB.SpecialInfo = (APTR)&gd->piB;
100: gd->pgR.GadgetID = gd->pgG.GadgetID =
101: gd->pgB.GadgetID = MAXCOLORS;
102: gd->OKtxt.FrontPen = gd->UNDOTxt.FrontPen =
103: gd->QUITxt.FrontPen = 1;
104: gd->OKtxt.ITextFont = gd->UNDOTxt.ITextFont = gd->
105: QUITxt.ITextFont = gd->hextxt.ITextFont = &taPlain;
106: gd->OKtxt.IText = gdtxt[0];
107: gd->UNDOTxt.IText = gdtxt[1];
108: gd->QUITxt.IText = gdtxt[2];
109: gd->hextxt.IText = gd->hexbuf;
110: gd->hextxt.DrawMode = JAM2;
111: gd->bgOK.NextGadget = &gd->pgB;
112: gd->bgUNDO.NextGadget = &gd->bgOK;
113: gd->bgQUIT.NextGadget = &gd->bgUNDO;
114: gd->bgOK.Activation = gd->bgUNDO.Activation =
115: gd->bgQUIT.Activation = RELVERIFY;
116: gd->bgOK.GadgetType = gd->bgUNDO.GadgetType =
117: gd->bgQUIT.GadgetType = BOOLGADGET;
118: gd->bgOK.GadgetRender = gd->bgUNDO.GadgetRender =
119: gd->bgQUIT.GadgetRender = (APTR)&gd->gadB;
120: gd->bgOK.GadgetText = &gd->OKtxt;
121: gd->bgUNDO.GadgetText = &gd->UNDOTxt;
122: gd->bgQUIT.GadgetText = &gd->QUITxt;
123: gd->bgOK.GadgetID = MAXCOLORS + 1;
124: gd->bgUNDO.GadgetID = MAXCOLORS + 2;
125: gd->bgQUIT.GadgetID = MAXCOLORS + 3;
126: gd->pnw.BlockPen = 1;
127: gd->pnw.IDCMPFlags = GADGETUP|GADGETDOWN
128: |ACTIVIEWINDOW|REFRESHWINDOW|MOUSEMOVE;
129: gd->pnw.Flags = ACTIVATE|GIMMEZEROZERO
130: |SIMPLE_REFRESH|WINDOWDRAG|WINDOWDEPTH;
131: gd->pnw.Type = CUSTOMSCREEN;
132: }
133: return(gd);
134: }
135:
136: static VOID __regargs cleanup
137: (UBYTE arg, struct DATA *gd) {
138:   switch(arg) {
139:     case 0: CloseWindow(gd->pw);
140:     case 1: FreeRemember(&gd->rkey, TRUE);
141:             FreeMem((UBYTE*)gd, sizeof(struct DATA));
142:   }
143: }
144:
145: static VOID __regargs SelectGadget
146: (UBYTE color, UBYTE pen, struct DATA *gd) {
147:   gd->selB.FrontPen = pen;
148:   DrawBorder(gd->pw->RPort, &gd->selB,
149:   (LONG)gd->cgad[color]->LeftEdge + 1,
150:   (LONG)gd->cgad[color]->TopEdge);
151:   return;
152: }
153:
154: static VOID __regargs BackupColors(struct DATA *gd) {
155:   UWORD i;
156:   for(i = 0; i < MAXCOLORS; i++)
157:     gd->colormap[i] = ((i < gd->maxcolors) ?
158:     ((UWORD)GetRGB4(gd->vp->ColorMap, (LONG)i)) : (0));
159:   return;
160: }
161:
162: static VOID __regargs GetColor
163: (UBYTE color, UWORD buffer[3], struct DATA *gd) {
164:   UWORD rgb, i;
165:   rgb = (UWORD)GetRGB4(gd->vp->ColorMap, (LONG)color);
166:   for(i = 0; i < 3; i++)
167:     buffer[i] = (rgb >> (8 - (i << 2))) & 0x000F;
168:   return;
169: }
170:
171: static VOID __regargs PrintHexVal(struct DATA *gd) {
172:   UWORD i, cbuf[3];
173:   GetColor(gd->color, cbuf, gd);
174:   for(i = 0; i < 3; i++)
175:     gd->hextxt.IText[i] = *(hextab + cbuf[i]);
176:   gd->hextxt.BackPen = gd->color;
177:   gd->hextxt.FrontPen = gd->maxcolors - 1 - gd->color;
178:   CenterIText(&gd->hextxt, gd->si.LeftEdge, (WORD)(gd->
179:   si.TopEdge + 1), gd->si.Width, gd->si.Height);
180:   PrintIText(gd->pw->RPort, &gd->hextxt, 0L, 0L);
181:   return;
182: }
183:
184: static VOID __regargs SetColor
185: (UBYTE num, struct DATA *gd) {
186:   SetRGB4(gd->vp, (LONG)num,
187:   (LONG)(gd->piR.HorizPot * RGBMAXVAL) >> 16,

```

Listing 2

```

188: (LONG)(gd->piG.HorizPot * RGBMAXVAL) >> 16,
189: (LONG)(gd->piB.HorizPot * RGBMAXVAL) >> 16);
190: PrintHexVal(gd);
191: return;
192: }
193:
194: static VOID __regargs SetGadget
195: (struct Gadget *gadg, UWORD val, struct DATA *gd) {
196:   NewModifyProp(gadg, gd->pw, NULL, FREEHORIZ|AUTOKNOB,
197:   (LONG)((ULONG)val * 0xFFFFL / (RGBMAXVAL - 1)),
198:   (LONG)((struct PropInfo*)gadg->SpecialInfo)->
199:   VertPot, 0xFFFFL / RGBMAXVAL,
200:   (LONG)((struct PropInfo*)gadg->SpecialInfo)->
201:   VertBody, 1L);
202:   return;
203: }
204:
205: static VOID __regargs UpdateGadgets
206: (UBYTE color, UWORD buffer[3], struct DATA *gd) {
207:   GetColor(gd->color, buffer, gd);
208:   SetGadget(&gd->pgR, buffer[0], gd);
209:   SetGadget(&gd->pgG, buffer[1], gd);
210:   SetGadget(&gd->pgB, buffer[2], gd);
211:   gd->si.PlaneOnOff = color;
212:   DrawImage(gd->pw->RPort, &gd->si, 0L, 0L);
213:   DrawBorder(gd->pw->RPort, &gd->siB, 0L, 0L);
214:   SelectGadget(color, 1, gd);
215:   PrintHexVal(gd);
216:   return;
217: }
218:
219:
220: extern UWORD* saveds BuildColorPalette
221: (UWORD *cmap, struct Screen *s, UBYTE *title) {
222:   struct IntuiMessage *msg;
223:   struct DATA *gd;
224:   ULONG class;
225:   UWORD gadgnum, i;
226:   if(gd = InitGlobal()) {
227:     gd->colormap = cmap;
228:     gd->pw.Title = title;
229:     gd->vp = &s->ViewPort;
230:     gd->si.Depth = s->RastPort.BitMap->Depth;
231:     gd->maxcolors =
232:     MIN(1 << s->RastPort.BitMap->Depth, MAXCOLORS);
233:     gd->pgR.Width = gd->pgG.Width =
234:     gd->pgB.Width = s->Width / FWIDTH + 4;
235:     gd->bgOK.Width = gd->bgUNDO.Width =
236:     gd->bgQUIT.Width = gd->si.Width =
237:     (s->Width / FWIDTH - 3 * GADGGAP) / 4;
238:     gd->si.LeftEdge = GADGLEFT;
239:     gd->bgOK.LeftEdge = GADGLEFT +
240:     (gd->si.Width + GADGGAP);
241:     gd->bgUNDO.LeftEdge = GADGLEFT +
242:     ((gd->si.Width + GADGGAP) << 1);
243:     gd->bgQUIT.LeftEdge = GADGLEFT +
244:     3 * (gd->si.Width + GADGGAP);
245:     gd->bgOK.Height = gd->bgUNDO.Height =
246:     gd->bgQUIT.Height = gd->si.Height =
247:     gd->pgR.Height = gd->pgG.Height =
248:     gd->pgB.Height = s->Height / FHEIGHT;
249:     gd->pgG.TopEdge = (gd->pgG.Height +
250:     GADGSPACE) + GADGTOP;
251:     gd->pgB.TopEdge = ((gd->pgB.Height +
252:     GADGSPACE) << 1) + GADGTOP;
253:     gd->bgOK.TopEdge = gd->bgUNDO.TopEdge =
254:     gd->bgQUIT.TopEdge = gd->si.TopEdge =
255:     3 * (gd->pgB.Height + GADGSPACE) + GADGTOP;
256:     gd->pnw.Width = s->Width / FWIDTH +
257:     (GADGLEFT << 1) + WINDOWSPARE;
258:     gd->pnw.Height = 6 * (gd->pgR.Height +
259:     GADGSPACE) + 3 * GADGTOP;
260:     gd->pnw.TopEdge = (s->Height - gd->pnw.Height) >> 1;
261:     gd->pnw.LeftEdge = (s->Width - gd->pnw.Width) >> 1;
262:     gd->pnw.Screen = s;
263:     CenterIText(&gd->OKtxt, 0, 1,
264:     gd->bgOK.Width, gd->bgOK.Height);
265:     CenterIText(&gd->UNDOTxt, 0, 1,
266:     gd->bgUNDO.Width, gd->bgUNDO.Height);
267:     CenterIText(&gd->QUITxt, 0, 1,
268:     gd->bgQUIT.Width, gd->bgQUIT.Height);
269:     gd->siB.LeftEdge = gd->cgadB.LeftEdge = GADGLEFT;
270:     gd->siB.TopEdge = GADGTOP +
271:     3 * (gd->pgB.Height + GADGSPACE);
272:     gd->cgadB.TopEdge = GADGTOP +
273:     ((gd->pgB.Height + GADGSPACE) << 2);
274:     for(i = 0; i < gd->maxcolors; i++) {
275:       if((gd->cgad[i] = (struct Gadget*)AllocRemember
276:       (&gd->rkey, sizeof(struct Gadget), MEMF PUBLIC|
277:       MEMF CLEAR)) == NULL || (gd->ci[i] = (struct Image*)
278:       AllocRemember(&gd->rkey, sizeof(struct Image),
279:       MEMF PUBLIC|MEMF CLEAR)) == NULL) break;
280:       gd->cgad[i]->NextGadget = (i > 0) ?
281:       (gd->cgad[i - 1]) : (&gd->bgQUIT);
282:       gd->cgad[i]->TopEdge = (i < (gd->maxcolors >> 1)) ?
283:       (GADGTOP + ((gd->pgB.Height + GADGSPACE) << 2)) :
284:       (GADGTOP + ((gd->pgB.Height + GADGSPACE) << 2) +
285:       gd->pgB.Height - 1);
286:       gd->cgad[i]->Width =
287:       (s->Width / FWIDTH / gd->maxcolors) << 1;
288:       gd->cgad[i]->LeftEdge =
289:       (i < (gd->maxcolors >> 1)) ?
290:       (GADGLEFT + i * gd->cgad[i]->Width) : (GADGLEFT +
291:       (i - (gd->maxcolors >> 1)) * gd->cgad[i]->Width);
292:       gd->cgad[i]->Height = gd->pgB.Height;
293:       gd->cgad[i]->Flags = GADGNONE|GADGIMAGE;
294:       gd->cgad[i]->Activation = GADGIMMEDIATE;
295:       gd->cgad[i]->GadgetType = BOOLGADGET;
296:       gd->cgad[i]->GadgetRender = (APTR)&gd->ci[i];
297:       gd->cgad[i]->GadgetID = (UWORD)i;

```

Listing 2


```

298: gd->ci[i]->LeftEdge = gd->ci[i]->TopEdge = 1;
299: gd->ci[i]->Width = gd->cgad[i]->Width - 2;
300: gd->ci[i]->Height = gd->cgad[i]->Height - 2;
301: gd->ci[i]->Depth = s->RastPort.BitMap->Depth;
302: gd->ci[i]->PlaneOnOff = i;
303: gd->pnw.FirstGadget = gd->cgad[i];
304: }
305: InitBorder(gd->gpoints,-3,-1,(WORD)(s->Width /
306: FWIDTH + 1),(WORD)((gd->pgB.Height < 1) - 1));
307: InitBorder(gd->ipoints,-2,-1,
308: gd->si.Width,gd->si.Height);
309: InitBorder(gd->spoints,-2,0,
310: (WORD)(gd->cgad[0]->Width - 2),
311: (WORD)(gd->cgad[0]->Height - 1));
312: if(gd->pw = (struct Window*)OpenWindow(&gd->pnw)) {
313:   gd->color = 0;
314:   BackupColors(gd);
315:   UpdateGadgets(gd->color,gd->undobuf,gd);
316:   FOREVER {
317:     WaitPort(gd->pw->UserPort);
318:     while (msg = (struct IntuiMessage*)
319:       GetMsg(gd->pw->UserPort)) {
320:       class = msg->Class;
321:       if(class & (GADGETUP|GADGETDOWN)) {
322:         gadgnum = ((struct Gadget*)
323:           (msg->IAddress))->GadgetID;
324:       }
325:       ReplyMsg((struct Message*)msg);
326:       switch(class) {
327:         case GADGETUP:
328:           if(gadgnum > MAXCOLORS) {
329:             if(gadgnum == MAXCOLORS + 1) {
330:               BackupColors(gd);
331:               cleanup(0,gd);
332:               return(cmapi);
333:             }
334:             if(gadgnum == MAXCOLORS + 2) {
335:               GetColor(gd->color,gd->buf,gd);
336:               SetRGB4(gd->vp,(LONG)gd->color,
337:                 (LONG)gd->undobuf[0],(LONG)gd->undobuf[1],
338:                 (LONG)gd->undobuf[2]);
339:               UpdateGadgets(gd->color,gd->undobuf,gd);
340:               for(i = 0; i < 3; i++)
341:                 gd->undobuf[i] = gd->buf[i];
342:             }
343:             if(gadgnum == MAXCOLORS + 3) {

```

Listing 2

```

344:   LoadRGB4(gd->vp,gd->colormap,(LONG)gd->maxcolo
rs);
345:   cleanup(0,gd);
346:   return(NULL);
347: }
348: break;
349: }
350: case GADGETDOWN:
351:   if(gadgnum < MAXCOLORS) {
352:     SelectGadget(gd->color,0,gd);
353:     gd->color = gadgnum;
354:     UpdateGadgets(gd->color,gd->undobuf,gd);
355:     break;
356:   }
357: case MOUSEMOVE:
358:   SetColor(gd->color,gd);
359:   break;
360: case ACTIVEWINDOW:
361:   UpdateGadgets(gd->color,gd->buf,gd);
362:   break;
363: case REFRESHWINDOW:
364:   BeginRefresh(gd->pw);
365:   UpdateGadgets(gd->color,gd->buf,gd);
366:   EndRefresh(gd->pw,TRUE);
367:   break;
368: }
369: }
370: } else cleanup(1,gd);
371: }
372: }
373: return((UWORD*)-1);
374: }

```

Listing 2

```

1: /*=====
2: **
3: **          Palette.c
4: **
5: **          by Franz-Josef Reichert
6: **
7: **          Lattice C-Compiler V 5.04
8: **
9: /*=====
10:
11: /* lc -v -w -cf -O -t -Lnt palette; (shared-library) */
12: /* lc -v -w -cf -O -t -Lnt+global.o+

```

Listing 3



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 Verity.
Alle Disks sind:
— 100 % Virus- und Error frei
— etikettiert.

Leerdisketten 3,5" 2DD von
Sentinel ab 1,25 DM
Sony ab 1,70 DM

3 ausführliche Katalogdisketten
mit Kurzbeschreibung aller
Programme gegen 10,— DM
(V-Scheck/Briefmarken) anfordern!
gratis zu unseren Katalogen:
Virusskiller, CLJ-Wizard + Turbo Backup

Das große Amiga-PD-Handbuch
Band I-IV + alle 42 Disks
+ 3 Katalogdisketten
(Einzelpreise erfragen) **299,—**

Pakete für Einsteiger und Anwender
(jeweils 10 Disketten)

Einsteiger 1,2; Spiele 1,2,3;
Sound; Grafik; Modula II; Glanzlichter
jedes Einzelpaket 35,— DM
3 Pakete nach Wahl nur 99,— DM

Floppy 3 1/2" int.
Floppy 3 1/2" ext. } abschaltbar
Floppy 5 1/4" ext. } mit allen
Extras

+ 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 #

Computer & Zubehör

Der Umbausatz

In der Grundausstattung bietet er Ihrem A 500 mit seinem Laufwerk einer internen Speichererweiterung und einem zweiten 3,5" Slimlinelaufwerk Platz. Eine abgesetzte Tastatur erleichtert Ihnen Ihre Arbeit. Das Gehäuse ist so stabil, daß Sie ohneweiteres den Monitor darauf abstellen können. Eine Erweiterungsbox wird Ihnen ab ca. September noch mehr Platz und Power verschaffen. Bei weiteren Fragen rufen Sie uns doch einfach an, oder fordern den kostenlosen Katalog an, die Grundausstattung des

MW 500 System

kostet mit Kabelsatz und Tastaturgehäuse

nur **DM 349,—**

Speichererweiterung 512 kB	169,—	DM
3,5" ext. Laufwerk* durchgef. Bus, abschaltbar	189,—	DM
5,25" ext. Laufwerk*	249,—	DM
Kickstart ROM 1.3	59,—	DM
Kickst. Umschaltplatine	59,—	DM
Festplatten* für A500/A2000	ab 890,—	DM

*Betrieb im Bereich der deutschen Bundespost verboten.

(Stichwort AD 9/90)

Miky Wenngatz

Jägerweg 31, D-8031 Gilching

Tel: 08105/24540 Fax: 24530


```

13: /* buildcolorpalette.o palette; (stand-alone) */
14:
15: /* #define SHARED_LIBRARY */
16:
17: #include <exec/types.h>
18: #include <exec/interrupts.h>
19: #include <exec/io.h>
20: #include <exec/ports.h>
21: #include <devices/input.h>
22: #include <devices/inputevent.h>
23: #include <graphics/gfxbase.h>
24: #include <intuition/intuitionbase.h>
25: #include <libraries/dos.h>
26: #include <workbench/startup.h>
27: #include <proto/exec.h>
28: #include <proto/dos.h>
29: #include <proto/intuition.h>
30: #ifndef SHARED_LIBRARY
31: #define NO_PRAGMAS
32: #endif
33: #include "requester.h"
34:
35: #define F5_KEYDOWN 0x54
36: #define SIG_POPUP 0x01
37: #define SIG_END 0x02
38: #define SIG_BUSY 0x04
39: #define SIG_CLSCR 0x08
40: #define COLORCOUNT 109
41: #define COLORMAP 110
42: #define CODESIZE 308
43: #define MAXCOLORS 32
44:
45: #define WHITESPACE(c) ((c) == ' ' || (c) == '\t')
46: #define QUOTE(c) ((c) == '"' || (c) == '\'')
47: #define LINEEND(c) ((c) == '\0' || (c) == '\n')
48:
49: #define IDS_IS_BUSY(id) { Disable(); (id)->id_Signal |
50:   = SIG_BUSY; Enable(); }
51: #define IDS_NOT_BUSY(id) { Disable(); (id)->id_Signal &
52:   = ~SIG_BUSY; Enable(); }
53:
54: extern struct WBStartup *WBenchMsg;
55: static VOID RemHandler(UWORD);
56: static LONG AddHandler(struct IntData *);
57: static struct InputEvent *asm_HandlerInterface
58: (register __a0 struct InputEvent*,
59:  register __a1 struct IntData*);
60: VOID _main(UBYTE*);
61:
62: #ifdef SHARED_LIBRARY
63: struct IntuiText SysReqText[] = {
64:   { AUTOFRONTPEN, AUTOBACKPEN, JAM1, 10, 6, AUTOITEXTFONT,
65:     REQUESTERNAME, &SysReqText[1] },
66:   {
67:     AUTOFRONTPEN, AUTOBACKPEN, JAM1, 10, 18, AUTOITEXTFONT,
68:     "nicht gefunden", NULL
69:   },
70:   {
71:     AUTOFRONTPEN, AUTOBACKPEN, JAM1, AUTOLEFTEDGE,
72:     AUTOTOPEDGE, AUTOITEXTFONT, "OK",
73:   },
74: };
75: #endif
76:
77: #endif
78:
79: static struct MsgPort *ip;
80: static struct IOStdReq *irb;
81: static struct Interrupt hint;
82: static UBYTE portname[] = "palette.port";
83: static UWORD colormap[MAXCOLORS];
84: static struct IntData {
85:   struct MsgPort id_Port; UBYTE id_Signal;
86: } INTERRUPT_DATA = {
87:   { NULL, NULL, NT_MSGPORT, 0, portname }, PA_SIGNAL, },
88:   SIG_BUSY,
89: };
90:
91: extern struct IntuitionBase *IntuitionBase;
92: extern struct GfxBase *GfxBase;
93: struct RequesterBase *RequesterBase;
94:
95: static UWORD code[] = {
96:   0x0000, 0x03F3, 0x0000, 0x0000, 0x0000, 0x0002, 0x0000,
97:   0x0000, 0x0000, 0x0001, 0x0000, 0x0023, 0x0000, 0x0018,
98:   0x0000, 0x03E9, 0x0000, 0x0023, 0x48E7, 0x000E, 0x49F9,
99:   0x0000, 0x0000, 0x93C9, 0x2C78, 0x0004, 0x4EAE, 0xFEDA,
100:  0x2A40, 0x4AAD, 0x00AC, 0x6614, 0x41ED, 0x005C, 0x4EAE,
101:  0xFE80, 0x41ED, 0x005C, 0x4EAE, 0xFE8C, 0x2940, 0x0000,
102:  0x4EBA, 0x001C, 0x4AAC, 0x0000, 0x670C, 0x4EAE, 0xFF7C,
103:  0x226C, 0x0000, 0x4EAE, 0xFE86, 0x4CDF, 0x7000, 0x7000,
104:  0x4E75, 0x2F0E, 0x43EC, 0x004C, 0x7000, 0x2C78, 0x0004,
105:  0x4EAE, 0xFDD8, 0x2040, 0x2940, 0x0004, 0x2068, 0x0022,
106:  0x2068, 0x0000, 0x202C, 0x0008, 0x43EC, 0x000C, 0x2C6C,
107:  0x0004, 0x4EAE, 0xFF40, 0x226C, 0x0004, 0x2C78, 0x0004,
108:  0x4EAE, 0xFE62, 0x2C5F, 0x4E75, 0x0000, 0x03EC, 0x0000,
109:  0x0001, 0x0000, 0x0001, 0x0000, 0x0006, 0x0000, 0x0000,
110:  0x0000, 0x03F2, 0x0000, 0x03EA, 0x0000, 0x0018, 0x0000,
111:  0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0355, 0x0FA0,
112:  0x0777, 0x0AA1, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
113:  0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
114:  0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
115:  0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
116:  0x6962, 0x7261, 0x7279, 0x0000, 0x0000, 0x0000, 0x03F2
117: };
118:
119:
120: BPTR_BackGroundIO = 0;
121: long _stack = 4000;

```

Listing 3

```

122: char * _procname = "palette.task";
123: long _priority = 5;
124:
125: static struct InputEvent *asm_HandlerInterface
126: (register __a0 struct InputEvent *ievent,
127:  register __a1 struct IntData *data) {
128:   if((ievent->ie_Class & IECLASS_RAWKEY)
129:   && (ievent->ie_Qualifier & IEQUALIFIER_LCOMMAND)
130:   && !(ievent->ie_Qualifier & IEQUALIFIER_REPEAT)
131:   && ievent->ie_Code == F5_KEYDOWN
132:   && (data->id_Signal & SIG_BUSY) == FALSE) {
133:     data->id_Signal = SIG_POPUP;
134:     if(ievent->ie_Qualifier &
135:       (IEQUALIFIER_LSHIFT|IEQUALIFIER_RSHIFT))
136:       data->id_Signal = SIG_END;
137:     Signal(data->id_Port.mp_SigTask,
138:     1L << data->id_Port.mp_SigBit);
139:     ievent->ie_Class = IECLASS_NULL;
140:   }
141:   return(ievent);
142: }
143:
144: static VOID RemHandler(UWORD arg) {
145:   switch(arg) {
146:     case 0: irb->io_Command = IND_REMHANDLER;
147:       irb->io_Data = (APTR)&hint;
148:       DoIO((struct IORequest*)irb);
149:     case 1: CloseDevice((struct IORequest*)irb);
150:     case 2: DeleteStdIO(irb);
151:     case 3: DeletePort(ip);
152:     default: break;
153:   }
154: }
155:
156: static LONG AddHandler(struct IntData *data) {
157:   if(ip = CreatePort(NULL, NULL)) {
158:     if(irb = CreateStdIO(ip)) {
159:       if(OpenDevice("input.device", NULL,
160:       (struct IORequest*)irb, NULL) == NULL) {
161:         hint.is_Data = (APTR)&data;
162:         hint.is_Code = (VOID(*)())HandlerInterface;
163:         hint.is_Node.ln_Pri = 51;
164:         irb->io_Command = IND_ADDHANDLER;
165:         irb->io_Data = (APTR)&hint;
166:         if(DoIO((struct IORequest*)irb) == 0)
167:           return(FALSE);
168:         else RemHandler(1);
169:       } else RemHandler(2);
170:     } else RemHandler(3);
171:   } return(TRUE);
172: }
173:
174: VOID _main(UBYTE *line) {
175:   BPTR fh;
176:   UBYTE *arg = NULL;
177:   UWORD *cm, i;
178:   ULONG iLock;
179:   struct IntData *id;
180:   struct WBArg *wbarg;
181:   struct Screen *First;
182:
183:   if(!WBenchMsg) {
184:     while(!WHITESPACE(*line) && !LINEEND(*line)) line++;
185:     while(WHITESPACE(*line) && !LINEEND(*line)) line++;
186:     if(!QUOTE(*line) && !LINEEND(*line)) {
187:       arg = line;
188:       while(!WHITESPACE(*line) && !LINEEND(*line)) line++;
189:     } else {
190:       arg = line + 1;
191:       while(!QUOTE(*++line) && !LINEEND(*line));
192:     }
193:     *line = NULL;
194:   } else {
195:     if(WBenchMsg->sm_NumArgs > 1) {
196:       wbarg = &WBenchMsg->sm_ArgList[1];
197:       arg = (UBYTE*)wbarg->wa_Name;
198:       CurrentDir(wbarg->wa_Lock);
199:     }
200:   }
201:   if(GfxBase = (struct GfxBase*)
202:   OpenLibrary("graphics.library", LIBRARY_VERSION)) {
203:     if(IntuitionBase = (struct IntuitionBase*)
204:     OpenLibrary("intuition.library", LIBRARY_VERSION)) {
205:       #ifdef SHARED_LIBRARY
206:       if(RequesterBase = (struct RequesterBase*)
207:       OpenLibrary(REQUESTERNAME, LIBRARY_VERSION)) {
208:         #endif
209:         Disable();
210:         if(id = (struct IntData*)FindPort(portname)) {
211:           if(id->id_Signal & SIG_BUSY) return;
212:           id->id_Signal = SIG_END;
213:           Signal(id->id_Port.mp_SigTask,
214:           1L << id->id_Port.mp_SigBit);
215:         }
216:         id = &INTERRUPT_DATA;
217:         if(id->id_Port.mp_SigBit = AllocSignal(-1)) {
218:           id->id_Port.mp_SigTask = FindTask(NULL);
219:           AddPort(id->id_Port);
220:           Enable();
221:           if(AddHandler(id) == FALSE) {
222:             IDS_NOT_BUSY(id);
223:             FOREVER {
224:               Wait(1L << id->id_Port.mp_SigBit);
225:               if(id->id_Signal == SIG_POPUP) {
226:                 iLock = LockIBase(0L);
227:                 First = IntuitionBase->FirstScreen;
228:                 UnlockIBase(iLock);
229:                 code[COLORCOUNT] = 1 << First->
230:                 RastPort.BitMap->Depth;
231:                 cm = BuildColorPalette(colormap,
232:                 First, (*arg) ? (arg) : ("P A L E T T E"));

```

Listing 3




```

234:         if(cm > NULL && *arg) {
235:             if(fh = Open(arg,MODE_NEWFILE)) {
236:                 for(i = 0; i < code[COLORCOUNT]; i++)
237:                     code[COLORMAP + i] = cm[i];
238:                 Write(fh,(UBYTE*)code,COESIZE);
239:                 Close(fh);
240:             }
241:         }
242:         IDS_NOT_BUSY(id);
243:         else if(id->id_Signal == SIG_END) break;
244:         RemHandler(0);
245:         RemPort(&id->id_Port);
246:         FreeSignal((LONG)id->id_Port.mp_sigBit);
247:     }
248:     CloseLibrary(&RequesterBase->rb_LibNode);
249: }
250: #ifdef SHARED LIBRARY
251: CloseLibrary(&RequesterBase->rb_LibNode);
252: }
253: else AutoRequest(NULL,&SysReqText[0],NULL,
254: &SysReqText[2],0,0,220,63);
255: #endif
256: CloseLibrary(&IntuitionBase->LibNode);
257: CloseLibrary(&GfxBase->LibNode);
258: }
259: return;
260: }
261: }
262: }

```

Listing 3

```

1: /*=====
2: **
3: **          Minex.c
4: **
5: **          by Franz-Josef Reichert
6: **
7: **          Lattice C-Compiler V 5.04
8: **
9: **=====
10:
11: #include <exec/types.h>
12: #include <intuition/intuition.h>
13: #include <proto/exec.h>
14: #include <proto/intuition.h>
15: #include <stdio.h>
16: #define NO PRAGMAS

```

Listing 4

```

17: #include "rēquester.h"
18:
19: #define BACKFILL 2 /* Hintergrundfarbe */
20: #define TEXTPEN 1 /* Textfarbe */
21: #define SHADOW 0 /* Schattenfarbe */
22:
23: void main(void);
24: extern struct IntuitionBase *IntuitionBase;
25: struct NewWindow nw = {
26:     0,0,640,200,0,1,CLOSEWINDOW|MOUSEBUTTONS,
27:     WINDOWDRAG|WINDOWCLOSE|WINDOWDEPTH|SMART_REFRESH|
28:     NOCAREREFRESH|WINDOWSIZING|GIMMEZEROZERO,NULL,NULL,
29:     "TestWindow",NULL,NULL,400,150,-1,-1,WBENCHSCREEN
30: };
31:
32: UBYTE *RequesterText[] = {
33:     "Requester-Library Version 33.11",
34:     "(C) by F.J.Reichert & AmigaDOS",
35:     "Dies ist eine Demonstration",
36:     "der Standard-Dialogbox!",
37: };
38: UBYTE *GadgetText[] = { "OK", "LoESCHEN", "ABBRUCH" };
39:
40: void main() {
41:     struct IntuiMessage *msg; struct Window *w;
42:     ULONG class = 0; UWORD code,result;
43:     if(IntuitionBase = (struct IntuitionBase*)
44:     OpenLibrary("intuition.library",LIBRARY_VERSION)) {
45:         if(w = (struct Window*)OpenWindow(&nw)) {
46:             while(class != CLOSEWINDOW) {
47:                 WaitPort(w->UserPort);
48:                 while(msg = (struct IntuiMessage*)
49:                 GetMsg(w->UserPort)) {
50:                     class = msg->Class; code = msg->Code;
51:                     ReplyMsg(&msg->ExecMessage);
52:                     if(class == MOUSEBUTTONS && code == SELECTUP) {
53:                         result = BuildBoolRequester(BACKFILL,TEXTPEN,
54:                         SHADOW,sizeof(RequesterText) >> 2,NULL,w,
55:                         RequesterText,GadgetText);
56:                         printf("Ergebnis: \"%s\\n\"",GadgetText[result]);
57:                     }
58:                 }
59:             }
60:             CloseWindow(w);
61:         }
62:         CloseLibrary((struct Library*)IntuitionBase);
63:     }
64: }

```

Listing 4

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 Zwittereigenschaften
M2Amiga-Inline-Code per Programm



DATABOX Amiga 29,- DM*
TOOLBOX EDITION 68000
Sonderpreis für AMIGA-DOS-Leser 9,- DM*
Paketpreis Edition 68000 + DATABOX Amiga 29,- DM*

* Unabhängig von der Anzahl der bestellten Produkte berechnen wir für das Inland 4,- DM bzw. für das Ausland 6,- DM Porto und Verpackung.

DMV-Verlag · Postfach 250 · 3440 Eschwege

Nr. 1/88-89

18,- DM 145 ÖS 18 sfr

DMV Verlag

EDITION



· Amiga · Atari · 68000 · Macintosh ·

AMIGA:

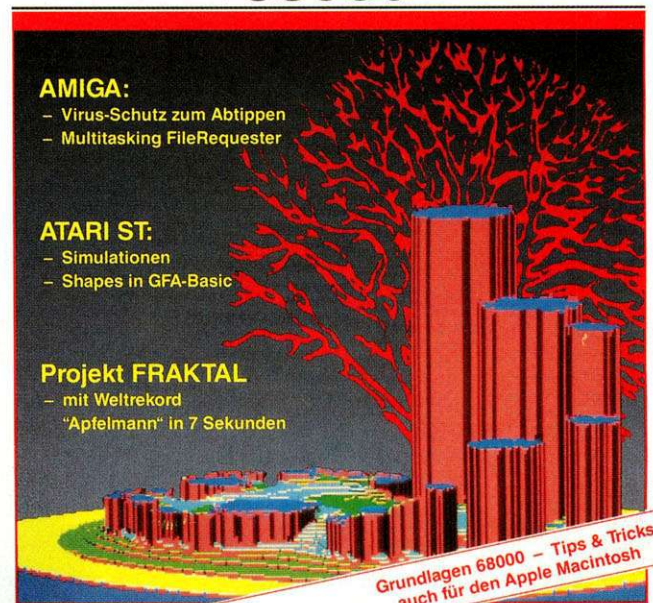
- Virus-Schutz zum Abtippen
- Multitasking FileRequester

ATARI ST:

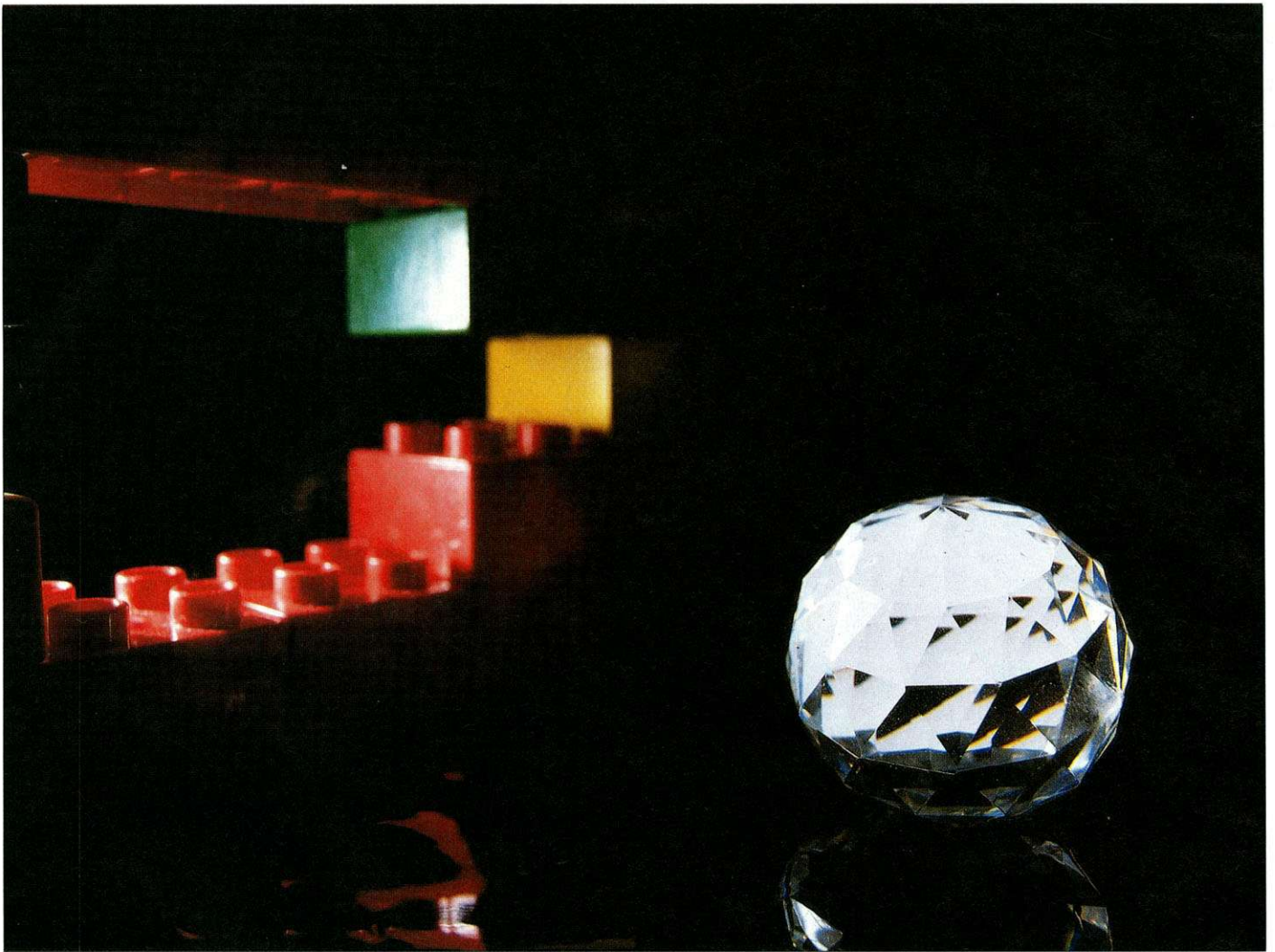
- Simulationen
- Shapes in GFA-Basic

Projekt FRAKTAL

- mit Weltrekord
- "Apfelmännchen" in 7 Sekunden



Grundlagen 68000 – Tips & Tricks
auch für den Apple Macintosh



Bernd Rudolf

Reflections-Werkstatt

Raytracing – worum es sich dabei handelt, dürfte nach dem ersten Teil dieser Werkstatt kein Geheimnis mehr für Sie sein. Diesmal wollen wir damit beginnen, kleine Szenen zu erstellen und berechnen zu lassen.

Bevor wir uns wieder in das Gewimmel der Routinen stürzen, wollen wir eine ganz einfache Szene erstellen, da ein Beispiel vorweg sicherlich einige Fragen klärt, die bisher noch unbeantwortet geblieben sind. Dazu rufen wir »Construct« auf und wählen im Project-Menü »LoadMat«. Es erscheint daraufhin ein File-Requester, der bereits auf den Pfad- und Dateinamen für die Materialdatei voreingestellt ist. Wir müssen also nur noch »OK« anklicken. Jetzt konstruieren wir eine Kugel und einen Quader. Von vorne gesehen, soll die Kugel vor dem

Kursfahrplan



Teil 1: Grundsätzliche Bedienung

Teil 2: Material und Licht

Teil 3: Werkstoffe und Oberfläche

Teil 4: Reflections in der Praxis

Quader liegen und diesen nicht ganz verdecken. Nun muß die Kameraeinstellung vorgenommen werden: Wählen Sie im Plotten-Menü »Kam.beweg.«. Es erscheinen

einige Schieberegler und eine Art Bühne. Den linken vertikalen Schieberegler mit der Unterschrift »VZB« (Vor- und Zurück-Bewegung) müssen Sie nun etwas nach oben

schieben, so daß die »Bühne« auf Sie zukommt, bis ihre Ränder fast hinter der Umrandung verschwinden. Dann wählen Sie »Kugel« und ziehen den horizontalen Schieberegler nach links, bis Sie die Bühne ungefähr aus 45 Grad von der Seite sehen. Klicken Sie »OK« an. So, das war die Kameraeinstellung.

Als nächstes wollen wir die Materialien der beiden Körper festlegen. Deshalb wählen wir im Körper-Menü »Material« aus. Sofort erscheint eine Liste mit den Körpern, die »Construct« bekannt sind. Nehmen wir einmal an, Sie haben die Kugel auch »Kugel« genannt. Wählen Sie also Kugel.

Jetzt erscheint eine Liste mit allen zur Zeit verfügbaren Materialien, aus der Sie durch Auf- und Abbewegen der Maus und Drücken der linken Maustaste eines auswählen können. Die Kugel soll in diesem Fall aus Glas sein. Bevor wir jetzt dieselbe Prozedur für den Quader noch einmal wiederholen, rufen wir die Funk-

tion »LoadMat« erneut auf. Diesmal klicken wir jedoch »Take« an und wählen aus der Übersicht die Datei »Texturen« aus. Das bestätigen wir mit »OK« und ordnen mit der Material-Funktion dem Quader ein Material zu. Wie Sie sicherlich bemerkt haben werden, ist die Auswahl an Materialien um einige rot dargestellte erweitert worden. Hierbei handelt es sich allerdings nicht um Materialien als solche sondern um Texturen, doch dazu später. Wählen Sie »Schachbrett-Muster« und legen Sie fest, daß es sich hier um eine Flächen-Textur handelt. Damit ist auch schon die Festlegung des Materials geschehen. Das einzige, was noch fehlt, ist das Licht. Das erledigt die Funktion »NeuLicht« im Licht-Menü für uns. Sie zeigt als erstes wieder die Liste der Materialien. Auf diese Weise wird die Farbe des Lichtes bestimmt. Beim ersten Aufruf von »NeuLicht« wählen wir »Glas«, beim zweiten »Schachbrett-Muster«. Nachdem man sich für die Farbe entschieden hat, muß man noch die Intensität festlegen. Dies passiert durch Bewegen der Maus nach rechts und links.

Für »Glas« soll die Lichtstärke ungefähr zwei betragen, für »Schachbrett-Muster« drei. »Construct« positioniert die Lampen mitten über den Körpern. Sie sind durch beige Kreise gekennzeichnet. Falls Sie sie im Moment noch nicht sehen sollten, müssen Sie die Zeichnung mit »Bild*2« solange verkleinern, bis Sie die Lampen sehen. Das beige Kreuz markiert übrigens die Kamera. Es ist vorteilhaft, die Lampen in Nähe der Kamera zu positionieren. Dies geschieht mit »LichtPos«. Wenn »Construct« wissen will, welche Lichtquelle Sie positionieren wollen, klicken Sie die gewünschte einfach an. Daraufhin verschwindet sie und erscheint dort wieder, wo Sie als nächstes klicken. Vorher sollten Sie allerdings die gewünschte Blickrichtung (hier von links) eingestellt haben, damit Sie die Lampen auch in die richtige Richtung bewegen. Vergessen Sie jetzt nicht, Ihre mühsam erstellte Szene abzuspeichern, bevor Sie »Construct« verlassen. Sie tun dies mit »SaveSz«. Als Name geben Sie zum Beispiel »Kugel.szn« ein, indem Sie das mit »Datei« bezeichnete Feld anklicken und den Namen

dann in dem Requester eingeben, der daraufhin erscheint. Danach müssen Sie auch noch die Kamera- und Lichteinstellung abspeichern. Dafür ist »SaveKam/Li« im Projekt-Menü da. Den Namen können Sie unverändert übernehmen, da »Construct« die Unterscheidung der Dateien für Sie übernimmt. Verlassen Sie nun »Construct« und kehren Sie zum »Manager« zurück.

Beams - Strahlen

Von dort aus muß nun »Beams« gestartet werden. Die Dateinamen für »Szene«, »Kam/Li« und »Bild« heißen alle »Kugel.szn« beziehungsweise so, wie Sie sie eben abgespeichert haben. In dem Feld hinter »Size« müßte jetzt »test1« stehen. Klicken Sie das Feld einmal an. »test2« ist eine vernünftige Größe, die in der Berechnung nicht zu lange dauert, bei der aber die Szene schon recht gut zu erkennen ist. Jetzt müssen Sie nur noch »Ray_Tie« auf zwei stellen. Mit einer Raytracing-Tiefe von zwei erreicht man, daß durchsichtige Gegenstände auch wirklich durchsichtig erscheinen. Das heißt, daß der Strahl, der vom Hintergrund eines durchsichtigen Körpers ausgeht, diesen auch durchqueren kann. So, das war's auch schon. Sie können »Beams« jetzt starten. Über dem Start-Gadget sehen Sie die Status-Zeile. Hier können Sie verfolgen, womit »Beams« gerade beschäftigt ist. Zuerst ruft »Beams« »Grid« auf, dann werden nacheinander die Zeilen 1 bis 52 bearbeitet. Wenn in der Status-Zeile »ruht« erscheint, kehren Sie in den Manager zurück und rufen »Show« auf.

Als Bildname setzen Sie den Namen ein, den Sie auch in »Beams« als Bildname verwendet haben (hier »Kugel.szn«) und als IFF-Dateinamen am besten denselben Namen mit Suffix ».iff« oder ».lbm« (zum Beispiel Kugel.iff). Mit dem Gadget »ViewMod« können Sie zwischen HAM- und Normal-Darstellung wählen. Wie Sie sich entscheiden, hängt allein davon ab, welche Auflösung das Bild hat (HAM bis maximal 320x256) und wie Sie es weiter verarbeiten wollen, da viele Malprogramme den HAM-Modus nicht unterstützen. Starten Sie jetzt »Show«. Nach kurzer Rechenzeit wird der Bildschirm dunkel und

der Bildaufbau beginnt. Am Ende sollte das Bild so aussehen wie Bild 2. Dann kehren Sie durch Drücken der linken Maustaste wieder zu Show zurück.

Entspricht das Bild Ihren Vorstellungen, so wird es im Directory »Reflections/iff« unter dem Namen abgespeichert, den Sie zuvor angegeben haben, wenn Sie jetzt »JA« anklicken. Sollte Ihnen das Bild so noch nicht gefallen, klicken Sie einfach »NEIN« an und editieren es mit »Construct« erneut. Bevor wir uns wieder »Construct« zuwenden, wollen wir schnell noch einmal auf »Get_IFF« eingehen: Der einzige Parameter, den »Get_IFF« benötigt, ist der Filename eines IFF-Bildes inklusive Pfadname. Wenn Sie den Pfadnamen nicht mit angeben, wird die Datei automatisch im Directory »Reflections/iff« gesucht.

Neue Oberflächen, Materialien und Texturen

Jedes Material, das wir mit »Construct« an einen Körper vergeben können, besteht aus einer Oberfläche, einer Farbe und eventuell einer Textur. Wir wollen jetzt einmal zwei neue Materialien kreieren und sie den Körpern der anfangs erstellten Szene zuordnen. Das eine Material ist »Mauerwerk«, das andere ist »mattes_Glas«. Bevor wir mit »Construct« loslegen können, müssen wir erst einmal D-Paint zu Hilfe nehmen und eine Mauer zeichnen.

Dazu öffnen wir einen Screen in der Größe 640x256 Punkte mit mindestens vier Farben.

Die Hintergrundfarbe setzen wir auf schwarz, die Farben eins und zwei auf rot und weiß. Jetzt zeichnen wir mit rot ein 16x5 Punkte großes ausgefülltes Rechteck, das wir dann als Brush definieren und nochmals neben das gezeichnete Rechteck mit einem Abstand von zwei Punkten setzen. Mitten unter diese zwei Rechtecke setzen wir nochmal eine Reihe aus drei Rechtecken, die voneinander wieder zwei Punkte Abstand, nach oben aber nur einen Punkt Abstand haben. Am besten, Sie führen die Zeichnung mit der Lupe durch.

Wir haben jetzt eine kleine Mauer aus fünf Steinen, die wir erneut als Brush definieren. Diese Definition muß sehr exakt erfolgen. Dazu bewegen wir das Kreuz auf den linken oberen Punkt des linken oberen Backsteins, drücken die linke Maustaste und ziehen die Maus so weit nach rechts unten, bis die untere Backsteinreihe genau über der unteren Linie des Brushes ist und zwischen dem oberen rechten Backstein und der rechten Linie des Brushes genau eine Punktreihe frei bleibt. Ist der Brush exakt so definiert, so können wir das Fill-Gadget mit der rechten Maustaste anklicken und im Fill-Requester »From Brush« anklicken.

In dem Fenster für das Füllmuster müßte jetzt sofort eine schöne gleichmäßige Mauer erscheinen. Falls die Abstände zwischen den Steinen nicht stimmen, haben Sie den Brush falsch ausgeschnitten. Ansonsten bestätigen Sie mit »OK« und zeichnen ein ausgefülltes Quadrat. Siehe da, die Mauer ist perfekt. Wenn Sie

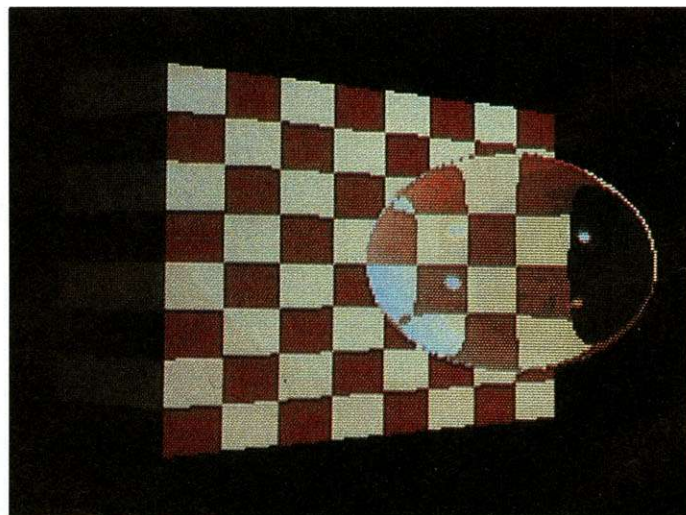


Bild 1. »Beams«, ein Reflections-Programmteil

jetzt im Fill-Requester wieder »Solid« einstellen und ein weißes ausgefülltes Rechteck zeichnen, auf das Sie dann per Brush die rote Mauer setzen, ist die Textur fast fertig. Sie müssen nur noch die Mauer so ausschneiden, daß die eine Backsteinreihe genau in der Mitte durchgeschnitten wird und die andere komplett innerhalb des Brushes liegt. Das ist nicht weiter schwer, wenn man berücksichtigt, daß die Punkte unter der Linie des Brushes noch zu diesem dazugehören.

Wenn Sie alle Punkte nacheinander befolgt haben, müßten Sie jetzt einen perfekten Mauerbrush haben, den Sie mit »Save Brush« auf die Reflections-Diskette im Directory »Reflections/text« speichern. So, das war's.

Eine neue Textur,

Aus dem eben abgespeicherten IFF-File müssen wir jetzt mit »Construct« eine Textur herstellen. Hierzu dient die Funktion »NeuTex« im Material-Menü. Klicken Sie IFF-File an, und geben Sie den Namen ein, unter dem Sie gerade den Brush im Text-Directory gespeichert haben. Unter »Name« geben Sie den Namen an, den die Textur haben soll: beispielsweise »Backstein«.

Die vier Gadgets in der oberen Bildschirmhälfte haben folgende Bedeutung: »Etik« legt fest, daß es sich bei diesem Material um eine »Etikett-Textur« handelt. Darunter versteht man beispielsweise das Etikett einer Flasche. Die Flasche besteht nicht nur aus dem Etikett, sondern auch aus

Glas. Da man einem Körper aber nicht ein Material und eine Textur zuordnen kann, wird mit »Etik« eine Textur in ein Material eingebettet. »Genl« entspricht in seiner Funktion einem »normalen« Genlock: Überall, wo im IFF-Bild die Hintergrundfarbe zu sehen ist, erscheint beim Raytracing die Farbe und Oberfläche des Grundmaterials. »Mat« ist eine Erweiterung der Genlock-Textur. Hier kann man mit Hilfe des Feldes »Mat_Tab« jeder Farbe des IFF-Bildes ein eigenes Material zuordnen. Auf diese Weise könnte man den weißen Fugen der Mauer das Material »Spiegel« zuordnen. Die Effekte sind mitunter recht verblüffend, besonders wenn man verschiedene Farben der Textur auf »nichts« setzt und sie somit unsichtbar macht.

Die »Pkt«-Option ist vor allem dann zu empfehlen, wenn als Texturmuster sehr kleine Brushes verwendet werden. Auf großen Flächen läßt »Reflections« dann die Farben ineinander verlaufen. Die vier Einstellungen können beliebig kombiniert werden. Für unsere Mauer wählen wir jedoch keine der Eigenschaften aus, womit wir auch schon mit der Textur-Definition fertig sind.

Als nächstes müssen wir die Textur in ein Material einbinden, denn Texturen allein lassen sich noch nicht an Körper vergeben. Zuerst sollten wir jedoch mit »Load Mat« alle bereits vorhandenen Materialien laden. Dann wählen wir »NeuMat« an. Als erstes legen wir den Namen des Materials fest. Wie wäre es mit »Mauer«? Für die Oberfläche setzen

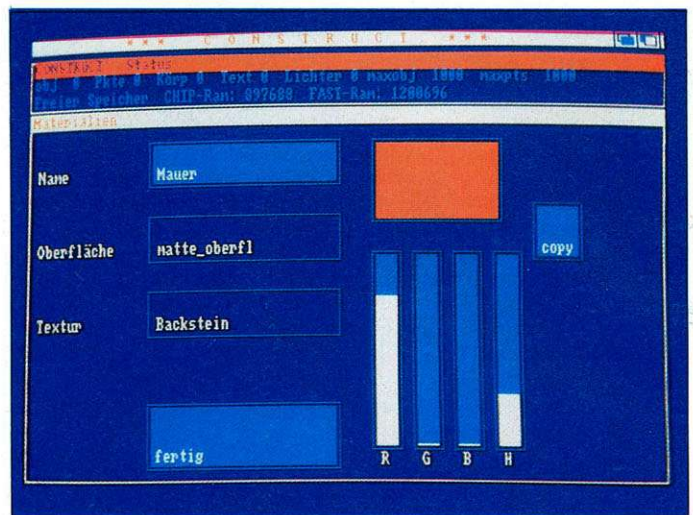


Bild 2: Der Menüpunkt »Neumat« aus »Construct«

wir »matte_oberfl« ein. Die Textur, die wir eben erstellt haben, hieß »Backstein«. Mit den vier Schieberegler kann man die Farbe einstellen. Wir wählen ein schönes Rot, was eigentlich überflüssig ist, da es sowieso von der Textur überschrieben wird. Nur für das Licht und den Fall, daß keine Textur verwendet wird, muß »Reflections« wissen, welche Farbe das Material haben soll. Wenn wir fertig sind, müssen wir die Einstellungen sichern.

Hinterher sollte man unbedingt im Projekt-Menü »SaveMat« anwählen. Als Dateiname geben Sie am besten »materialien2« an. Wenn Sie anfangs auch schon die Texturen hinzugeladen haben, wird jetzt alles zusammen abgespeichert, so daß Sie beim nächsten Mal nur noch eine einzige Datei laden müssen. Das neu erstellte Material kann man jetzt ebenso wie jedes andere Material an Körper vergeben.

eine neue Oberfläche...

Den Quader aus der Beispielszene wandeln Sie jetzt einfach mal in eine Backsteinwand um. Was noch fehlt, ist ein neues Material für die Kugel. Sie sollte aus mattem Glas sein. Das erste, was wir brauchen, ist eine neue Oberfläche, die wir mit »NeuOb« (Bild 3) erstellen. Zuerst klicken wir »Copy« an und kopieren »matte_oberfl«. Den Namen ändern wir sofort auf »mattes_Glas_oberfl«, damit wir nicht plötzlich zwei

Oberflächen mit gleichem Namen haben.

Mit den fünf Balken der Balkenanzeige lassen sich die Koeffizienten einstellen. Sie legen neben dem Graphen hauptsächlich das Erscheinungsbild fest. »df« ist der diffuse Farbanteil, »sp« der spiegelnde und »br« der brechende. Auf diese Weise können wir festlegen, welche Farbe das Licht hat, nachdem es an der Oberfläche reflektiert oder gebrochen wurde. »l« bestimmt das Eigenleuchten eines Körpers. Es ist allerdings keine Lichtquelle als solches, das heißt, ein Körper mit Eigenleuchten wirft keine Schatten. »bi« ist der Brechungsindex, der festlegt, wie stark ein Strahl beim Durchdringen eines durchsichtigen Körpers abgelenkt (gebrochen) wird.

und zwei neue Materialien

Die Werte lassen sich einstellen, indem Sie mit der Maus über dem entsprechenden Balken klicken. Wie Sie sicherlich bemerkt haben werden, beeinflussen sich die Werte gegenseitig, denn eine spiegelnde Oberfläche kann nur begrenzt diffus oder durchsichtig sein. Die Kurve des Glanzlichtes können Sie sehr einfach mit der Maus verändern. Je steiler die Kurve abfällt, um so glatter ist die Oberfläche. Stellen Sie die Werte entsprechend Bild 3 ein. Die neue Oberfläche müssen wir ebenso wie die Textur noch in ein Material einbinden.

Das funktioniert genauso, wie oben beschrieben, mit »Neu-

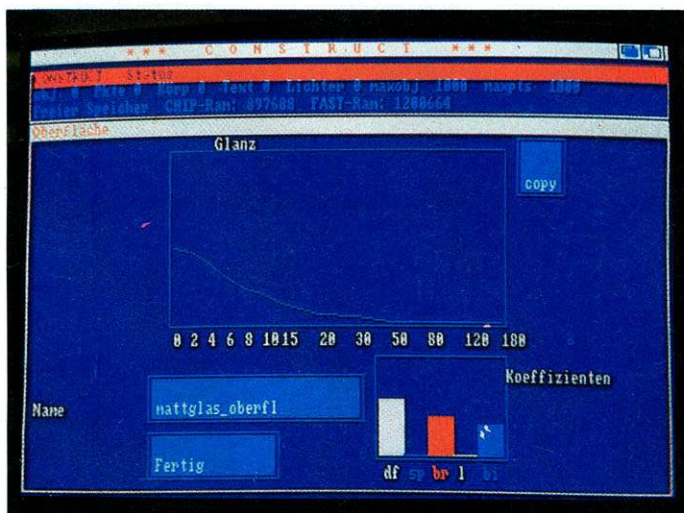


Bild 3: »Neu Oberfläche« Menüpunkt aus »Construct«

Mat«. Als Farbe wählen wir diesmal ein helles Grau und dazu die eben erstellte Oberfläche. Als Textur ist »keine_textur« anzugeben, und das Material nennen wir »mattes_Glas«. Dieses Material ordnen wir der Kugel unserer Beispielszene zu. Fertig berechnet sollte die Szene dann so aussehen wie in Bild 4. Wie nicht anders zu erwarten, kann man mit den Menüpunkten »Matweg«, »Obweg« und »Texweg« Materialien, Oberflächen und Texturen aus dem Speicher entfernen. Um sie auch auf Diskette zu löschen, muß man hinterher »Save Mat« ausführen.

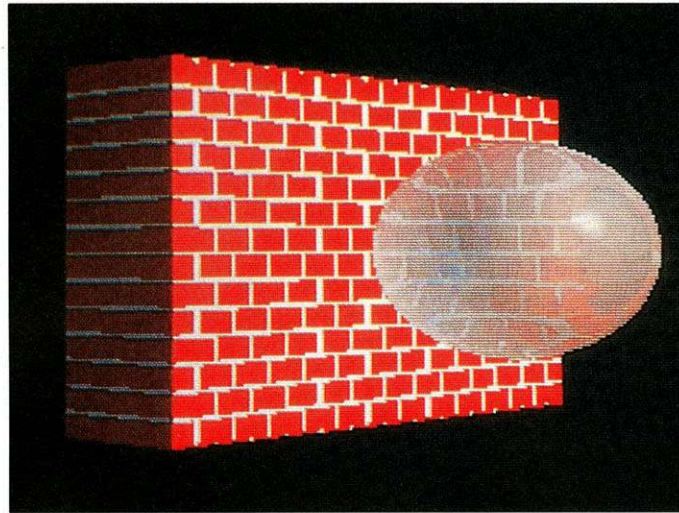


Bild 4: Das fertige Mauerbild

Die Kamera-einstellung

Als nächstes sollten wir uns noch einmal etwas intensiver als am Anfang dieses Kurses mit der Kameraeinstellung befassen. Mit der Funktion »Kamera« läßt sich die Kamera wie auf einer Kugeloberfläche um die Szene herum bewege-

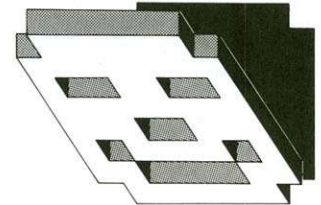
gen. Mit dem horizontalen Regler läßt sich die Szene von links, vorne, rechts oder hinten betrachten, mit dem vertikalen von oben, vorne oder unten. Die zweite Möglichkeit, die Kamera einzustellen, ist durch »Kam.beweg« gegeben: Mit dem linken Regler läßt sich die Kamera beliebig weit vor- und zurückbewegen.

»RLD« bedeutet Rechts-Links-Drehen und »RLB« Rechts-Links-Bewegen. »Tel« ist das Teleobjektiv. Entsprechend ist »HRB« Hoch-Runter-Bewegen und »HRD« Hoch-Runter-Drehen. Mit »Roll« kann man die Kamera auf den Kopf drehen. Wie die Szene im Endeffekt aussehen wird, kann man sich mit »persp« an-

sehen. Achten Sie darauf, daß alles, was Sie in Ihrer Szene sehen möchten, zwischen den zwei gestrichelten Linien liegen muß.

Zum Schluß möchte ich Ihnen eine kleine Aufgabe stellen. Anhand der Lösung können Sie ersehen, ob alle Informationen, die wir bisher durchgearbeitet haben, auch richtig verarbeitet wurden. Konstruieren Sie doch einmal eine Weinflasche mit Etikett, die neben einem Weinglas auf einem Schachbrett steht. Wenn Ihnen die Erstellung dieser Szene keine Probleme bereitet, sind Sie sattelfest für den nächsten Teil der Reflektions-Werkstatt.

(hs)



Achtung, Super-Service!

Als besonderen Service der AMIGA DOS gibt es ja die bekannten privaten Kleinanzeigen, in denen Soft- und Hardware angeboten werden kann oder die einfach der Kontaktaufnahme mit anderen Amiga-Interessierten dient.

Für die Ausgaben 10, 11 und 12 des Jahrgangs 1990 können wir Ihnen nun eine besondere Attraktion bieten:

Die kostenlose private Kleinanzeige!

Und so können Sie dieses tolle Angebot wahrnehmen:

Benutzen Sie einfach die im Heft befindliche Servicekarte und notieren Sie Ihre Kleinanzeige. Schicken Sie diese Postkarte dann an den Verlag, und Ihre Kleinanzeige erscheint in der nächsterreichbaren Ausgabe von AMIGA DOS!

Kleinanzeigen

Biete Software

Harddiskmenü Programmstart per Mausclick 49 DM, Demo 10 DM. IHS Bartsch, Lindenstr. 21, 8900 Augsburg G

PD-Soft für Amiga & PC Info-Disk gratis bei: Brunosoft/Schreiber, 1000 Berlin 51, Sommerstr. 37 G

Neue Grafik PD-Serie eröffnet: Info: Roger Hassler, 05 04 25 23 16 AMIGA-VICE, Grafik-PD-Serie: Grafiken, Anims, Demos, Kurse...

Verk. Aztec C V3.6a Developer, Zeitschriften. Tel. 05 41/1 79 81

AMIGA SuperLiga V1.3! mit Spieltagen, Saisonverlauf, Wappen, Play-Off-Tabelle, Stadiondigisound, Gratisinfo "SL1.3/D" sofort anfordern beim Autor: Rolf Morlock, Bahnhofstr. 42, D-6729 Jockgrim, Tel. 0 72 71/5 13 44 G

BAVARIAN-Public-Domain: das ist 160 Disketten fast nur deutscher PD. Gratisinfo bei Fr. Neuper, 8473 Pfreimd, Postfach 72 G

Biete Hardware

Archimedes A310, 4 MB, 2. Laufwerk, Soft + Lit. Maic, Tel. 0 40/7 37 44 71

KICKSTART AUF EPROMS f. AMIGAS INFO: S. M., RAB12/213, 8000 MÜNCHEN 80

Verschiedenes

SUPERBASE-DML-Kenner gesucht! Wer schreibt mir gegen gute Bezahlung die Lösung zu einem Problem? Tel: 0 57 73/14 36 Günter

Amiga 5,25" intern? Wir haben es: Shugart-Floppy 720/880 KB (Keine HD-Disketten mehr erforderlich!), 40/80 Track, incl. Anleitung DM 149,-; 3,5"-Disks 100 St. DM 99,95; Mentis GmbH, Poststr. 15, 4650 Gelsenkirchen, Tel. 02 09/5 25 72 G

Computergeschäfte mit Public-Domain. Einkaufsquelle für supergünstige Leerdisketten, uvm. Gratisinfo "DMV-AM" von: Rolf Morlock, Bahnhofstr. 42, D-6729 Jockgrim, Tel. 0 72 71/5 13 44 G

Geschäftsverbindungen

Geld verdienen mit Amiga! Wer hat Interesse, an Amiga-Büchern mitzuarbeiten? Wir suchen ständig Experten für Fachbücher zu allen Gebieten. Schriftliche Informationen bei: technicSupport, Bundesallee 36-37, 1000 Berlin 31 G



Robert Münch

Hilfe bei Viren

ZeroVirus

Viren sind sehr unfreundliche Plagegeister, und ihre Bekämpfung ist oft schwierig. Mit »ZeroVirus« aus dem Bereich der Public Domain haben Viren keine Chance. Wir möchten Sie in diesem Workshop mit diesem Programm vertraut machen.

Bevor wir mit der Installation und Funktion des Programms beginnen, möchte ich einen kurzen Abriss über die Möglichkeiten von »ZeroVirus« geben. Das Programm ist ein vollständig integrierter Viren-Checker und -Killer, ein Bootblock-Archivierer und Link-Virentester. Disketten können mit einem von vier verschiedenen Bootblöcken installiert werden. »ZeroVirus« benutzt sogenannte Brainfiles (frei Übersetzt: Gedächtnisdateien), um ein Update auf neue Viren besonders einfach zu gestalten. Die Möglichkeit, durch einen Lernmodus selbst Bootblöcke in das Brainfile zu übernehmen und somit die Diskette dem Programm bekannt zu machen, ist selbstverständlich auch gegeben. Besonders nützlich ist die Möglichkeit, »ZeroVirus« als Task auf der Workbench zu installieren.

Möchten Sie »ZeroVirus« starten, so gestaltet sich dies sehr einfach, entweder klicken Sie zweimal das Icon an oder tippen im CLI »ZeroVirus« ein.

Ran an den Feind

Die einzige Bedingung ist, daß sich das Brainfile (heißt übrigens sinnigerweise »Brainfile.ZeroVirus«) entweder im aktuellen (meistens das gleiche Verzeichnis, in dem sich auch »ZeroVirus« befindet) oder im »S: Directory« befindet. Das Programm liest nun dieses Brainfile und führt danach einen Check der wichtigsten Systemvektoren durch. (Für Experten: Die Vektoren Warm, Cool und Cold Capture, der KickTagPtr, KickMemPtr und KickCheckSum werden überprüft.) Normalerweise sollten diese Vektoren den Wert \$000000 haben.

Ist das nicht der Fall, kann es sich um einen Virus im System handeln. »ZeroVirus« zeigt den Inhalt der Vektoren an und bietet nun die Möglichkeit, diese Vektoren auf \$000000 zu setzen und somit den Virus zu löschen. Wenn Sie noch einen Virenkiller im System installiert haben, empfehle ich Ihnen, sich für einen zu entscheiden, da sonst der eine von dem anderen glaubt er sei ein Virus (»ZeroVirus« verträgt sich unter Umständen nicht mit anderen Virenkillern). Die reset-feste RAM-Disk der Kickstart 1.3 (RAD) wird aber als solche erkannt, so daß es in dieser Beziehung zu keinen Problemen kommt. Haben Sie Pech, und »ZeroVirus« zeigt die Existenz eines Virus an, so ist höchste Vorsicht geboten. Es gilt zunächst, den Virus zu bannen. Danach sollten Sie sich etwas

Zeit nehmen, denn jetzt kommt die große Arbeit auf Sie zu. Sie sollten jede Ihrer Disketten überprüfen und danach mit einem Schreibschutz versehen. »ZeroVirus« bittet Sie, eine von beiden Maustasten zu drücken, um in das Hauptmenü zu gelangen. Sie sehen nun verschiedene Optionen, die ganz einfach über Gadgets angesprochen werden. Eine kurze Beschreibung:

»Check Disk for Viruses« – Hiermit werden Disks auf Viren überprüft und gelöscht.

»Backup Bootblocks to Disk files« – Mit dieser Funktion können Bootblöcke auf Diskette gespeichert werden.

»Restore Bootblocks from Disk file« – Dies ist das Gegenstück zum vorher genannten. Die archivierten Bootblöcke werden zurückgeschrieben.

»Catalogue Backed-Up Bootblocks« – Durch Anwahl dieses Punktes erstellt »ZeroVirus« ein Inhaltsverzeichnis der abgespeicherten Bootblöcke.

»Adjust Colors« – Ermöglicht das Verändern der Farben von »ZeroVirus«.

»About ZeroVirus« – Gibt Informationen über das Programm.

»Iconify« – Reduziert »ZeroVirus« auf die Größe einer Windowzeile als Zeit- und Speicheranzeige.

»Exit from ZeroVirus« – Beendet das Programm und kehrt zum CLI oder der Workbench zurück.

Ins Detail

Nach diesem »Crash-Kurs« wollen wir einmal näher auf die einzelnen Funktionen von »ZeroVirus« eingehen. Eine erfolgreiche Großwildsafari ist ja auch nur durch eine gute Jagdausbildung zu erreichen (und wir jagen jetzt ein paar Viren). Ich werde dies chronologisch in der gleichen Reihenfolge wie oben durchführen.

Mit der Funktion »Check Disks for Viruses« werden die Bootblöcke der Disketten geprüft sowie die Startup-Sequenz, die Directories und Files auf einen Link-Virus gecheckt. In der oberen linken Ecke des Screens sind vier Disketten-Gadgets plazierte (nicht angeschlossene Diskettenlaufwerke sind in Geisterschrift gehalten). Diese vier Gadgets werden uns übrigens

noch oft in den anderen Funktionen von »ZeroVirus« begegnen. Wollen Sie nun eine Diskette überprüfen, müssen Sie nur Ihr »Nagetier« so über den Tisch locken, daß das »Computerdouble« über dem entsprechenden Laufwerkssymbol plazierte wird. Jetzt noch eben der Maus einen Klaps geben, und die Diskette wird überprüft (alle Tierschützer mögen mir für die mausfeindliche Beschreibung vergeben).

Kommt es dabei irgendwie zu einem Fehler, erscheint entweder ein Requester, oder eine Meldung wird am unteren Bildschirm angezeigt. Verläuft alles ohne Probleme, wird die Diskette geprüft und das Ergebnis der Überprüfung angezeigt. Rechts vom jeweilig aktuellen Laufwerk-Gadget wird eine Spezifikation des Bootblocks ausgegeben. Beinhaltet der Bootblock einen »normalen Bootcode«, so erscheint dort die Ausgabe »Normal DOS Bootblock«, dies teilt uns mit, daß alles in Ordnung ist.

Beinhaltet der Bootblock einen anderen als den »Standard Bootcode«, wird die Meldung »Non-Standard Bootblock« ausgegeben. Diese Meldung erscheint aber nur, wenn der Bootblock dem Brainfile von »ZeroVirus« noch nicht bekannt ist. Ansonsten wird nämlich der Name des Bootblocks ausgegeben. Handelt es sich um einen Virus, und »ZeroVirus« kennt diesen Virus, wird der Name des erkannten Virus in **rot** ausgegeben, beispielsweise »SCA virus recognized« oder »CCS-BOOT bootblock recognized«.

Am unteren Bildschirmrand erscheint daraufhin eine Meldung, was weiter zu tun ist. Sie kann wie folgt aussehen: »This disc is okay. Insert another disk to keep checking.« (Übersetzung: »Alles O.K. Andere Diskette zum Testen einlegen.«) oder »This disc contains a virus! INSTALL it immediately!« (Übersetzung: »Diskette beinhaltet Virus! Sofort installieren!«) Diskettenwechsel werden übrigens automatisch erkannt, so daß eine neu eingelegte Diskette sofort überprüft wird, ohne daß Sie das Laufwerk-Gadget noch einmal anklicken müssen.

Mit der ersten Option des Menüs »Learn« (Lernen) können Sie den aktuellen Bootblock dem Brainfile einverleiben

und somit für »ZeroVirus« bekannt machen. Wenn Sie »Learn« anwählen, erscheint ein Requester, in dem Sie einen maximal 60 Zeichen langen Kommentar (zum Beispiel: Name des Bootblocks und so weiter) eintragen. Die Zeichenkette sollte nicht mit einem Punkt oder dem Wort »recognized« enden, da »ZeroVirus« dies automatisch an den Text anhängt. Danach können Sie noch eine weitere Zeichenkette (wieder 60 Zeichen) eingeben, nämlich eine Beschreibung des Bootblocks. Eine Erweiterung der Zeichenkette findet diesmal nicht statt. Handelt es sich um einen Virus, sollte man einfach nur [RETURN] drücken. »ZeroVirus« fügt dann selbständig die Meldung »**This Disk contains a virus! INSTALL it immediately!**« in roter Schrift ein. Danach wird der Bootblock in das Brainfile geschrieben und ist damit dem Programm ab sofort bekannt. Übrigens, die Diskette, auf der sich das Brainfile befindet, sollte natürlich nicht schreibgeschützt sein, sonst kann »ZeroVirus« das Brainfile nicht aktualisieren.

Der Programmierer hat sich für die Verwaltung des Brainfiles etwas spezielles einfallen lassen. Jedesmal, wenn Sie einen neuen Bootblock mit der Funktion »Learn« archiviert haben, wird die User-Versionsnummer um eins erhöht. Die offizielle Versionsnummer des Brainfiles bleibt davon jedoch unberührt, sie wird nur bei der Veröffentlichung von neuen Brainfiles durch den Programmierer erhöht. Durch dieses System kann jeder Benutzer sofort feststellen, ob das gefundene Brainfile (beispielsweise in anderen PD-Serien, auf irgendwelchen Utility-Disketten und so weiter) eine höhere oder niedrigere Versionsnummer hat, und dieses gegebenenfalls austauschen oder erweitern. Dazu später jedoch mehr.

Als nächstes wäre »Install Disc« zu erläutern, damit wir mit sauberen Bytes den Viren endlich den Nährboden entziehen. Bei dieser Funktion haben Sie die Möglichkeit, einen von vier verschiedenen Bootblöcken auszuwählen, mit dem Sie die Diskette installieren wollen. Vorsicht ist bei dieser Funktion geboten, da der alte Bootblock natürlich verlorengeht. Ich empfehle Ihnen daher, immer erst ein Backup des Bootblocks zu er-

stellen und die Diskette dann zu installieren. Wenn Sie nun den rechten Mausknopf drücken, erscheint ein Requester. Hier können Sie nun den von Ihnen gewünschten Bootblock auswählen. Es stehen Ihnen verschiedene zur Verfügung:

- »Standard«: Dies ist der Standard AmigaDOS 1.3 Bootblock.

- »NoFastMem«: Dieser Bootblock schaltet alle autokonfigurierenden Speichererweiterungen ab.

- »BigScreenTest«: Bei einem PAL-Amiga kann es nach einem Reset dazu kommen, daß der Screen nur die Ausmaße eines NTSC-Screens besitzt (56 Zeilen weniger in der Vertikalen). Mit diesem Bootblock ist die Möglichkeit gegeben, bei einem solchen Fehler noch einmal zu booten.

- »AutoAddRam«: Falls Sie eine nicht autokonfigurierende Speichererweiterung besitzen, dann können Sie mit diesem Bootblock dem System einen Teilbereich des RAMs hinzufügen. In der darauffolgenden Abfrage können Sie die Start- und Endadresse (in hexadezimaler Schreibweise) eingeben. Erfolgt keine Eingabe, werden die Werte \$f80000 bis \$fbfff benutzt.

Mit der Funktion »View« können Sie sich den Bootblock anschauen, bevor Sie irgend etwas unternehmen wollen (dringendst zu empfehlen). Da der Bootblock 1024 Byte groß ist, sehen Sie auf der linken Seite die ersten 512 Bytes und auf der rechten Seite die letzten 512 Bytes. »ZeroVirus« zeigt alle Zeichen, die nicht mit dem Zeichen des Original-Bootblocks an der gleichen Stelle übereinstimmen, rot an und alle anderen weiß.

»Return to Main Menu« springt einfach zurück in das Hauptmenü.

»File Virus Check« ermöglicht das Überprüfen von Directories und Files auch Link-Viren/File-Viren. File-Viren können nicht in das Brainfile übernommen werden. Bei Auftreten eines neuen File-Virus wird vom Programmator eine neue Version des Brainfiles oder von »ZeroVirus« veröffentlicht. Welche File-Viren »ZeroVirus« erkennt, erfahren Sie beim Aufruf dieses Menüpunktes. Danach erscheint ein File-Requester, in dem Sie nun entweder ein Directory oder ein einzelnes File über-

prüfen können. Der Name des Directorys und des Files muß mit Pfad versehen in die Zeile »Drawer« und der Filename in Zeile »Filename« eingegeben werden. Wollen Sie den Check starten, müssen Sie nur noch auf »OK« klicken. Findet »ZeroVirus« nun einen File-Virus, so werden Sie gefragt, ob Sie diesen entfernen wollen. Wird der »BGS9-File-Virus« entdeckt, haben Sie noch die Möglichkeit, das Originalfile wieder herzustellen. Funktionierte dies nicht, so werden Sie darauf aufmerksam gemacht. Zum Schluß werden dann noch einige statistische Daten ausgegeben.

Besondere Tasten

Die »Toolkit Keys« können Sie vom Check-Disk-Menü aus benutzen. Diese drei Tasten stellen Funktionen zur Verfügung, die nicht über Menüs aufgerufen werden können.

[F1] »Uninstall« - Hiermit wird die aktuelle Diskette so behandelt, daß sie nicht mehr bootfähig ist.

[F2] »Fix Checksum« - Die Checksumme des Bootblocks wird restauriert und die Diskette somit wieder bootfähig.

[F3] »Copy Bootblock« - Der Bootblock der aktuellen Diskette kann damit auf eine andere kopiert werden. Sie werden dabei aufgefordert, das Ziellaufwerk anzuklicken.

Als nächstes nehmen wir uns die Funktion »Backup Bootblocks to Disk File« vor. Sinn dieser Funktion ist es, von Programmen, die eigene Bootblöcke benutzen, eine Kopie des Bootblocks auf Diskette abzulegen. Wird dieser nun durch einen Virus zerstört, schreibt man einfach den abgespeicherten Bootblock auf die kaputte Diskette zurück, und alles bleibt beim alten. Der Screen-Aufbau ähnelt sehr dem der Funktion »Check Disc for Viruses«. Lediglich die Menüpunkte »Learn« und »File Virus Check« wurden durch »Backup Bootblock« ersetzt. Für die anderen Funktionen gilt das oben erläuterte.

Wird »Backup Bootblock« nun aufgerufen, wird der Bootblock des aktuellen Diskettenlaufwerks eingelesen, und ein Filerequester erscheint. Sie müssen hier den Pfad angeben, wo der Bootblock sein Plätzchen auf der Diskette bekommen soll. »Zero-

Virus« bietet Ihnen auch hier wieder einen guten Service und fügt automatisch den Diskettennamen als Filenamen ein. Der Überblick, von welcher Diskette der Bootblock nun kommt, bleibt damit im großen und ganzen erhalten. Sie können den Filenamen natürlich auch ändern. Ich würde Ihnen empfehlen, sich eine eigene Diskette anzulegen oder für Festplattenbesitzer ein eigenes Directory. So, damit es nun weiter geht, klicken Sie einmal auf »OK«, und siehe da, ein neuer Requester erscheint. Wer nun Lust hat, kann hier einen 40 Zeichen langen Kommentar zum Bootblock eingeben, dieser wird dann mit dem Bootblock abgespeichert (soweit kein Fehler gemacht wurde). Damit wir unsere Bootblöcke auch von der Diskette wieder herunterbekommen, gibt es die Funktion »Restore Bootblocks from Disk Files«. Nach der Auswahl erscheint der uns bekannte File-Requester wieder und fragt nach dem Pfad und dem Filenamen. Wenn kein Fehler auftritt, wird der ausgewählte Bootblock geladen und auf die aktuelle Diskette gespeichert.

»View Saved Bootblock« richtet die gleiche Arbeit wie »View Bootblock« (siehe oben) mit einem kleinen Unterschied. Es kann ein Bootblock angesehen werden, der bereits abgespeichert ist. Bei der Auswahl erscheint wieder besagter File-Requester und fragt nach dem Pfad zu Ihrem Bootblock.

Übersicht wahren

Mit »Catalogue Backed-Up Bootblocks« können Sie ein Inhaltsverzeichnis der Bootblöcke eines bestimmten Directories erstellen lassen. Durch die Anwahl von »Catalogue to File« oder »Catalogue to Printer« wird das Inhaltsverzeichnis entweder an die Diskette oder aber an den Drucker gesendet.

»Include Comments« und »Include Dates« ermöglichen, daß auch Kommentare und Datum in dem Inhaltsverzeichnis erscheinen. Einmal »Generate Catalogue« angeklickt und der File-Requester stellt wieder die Frage (»Wänä, di Frogä«). Also Pfad eingeben, mit »OK« bestätigen, und der Spaß beginnt. Haben Sie sich als Ausgabemedium ein File ausgesucht, werden Sie jetzt nach dem

Filenamen gefragt. Wichtig ist zu wissen, daß nur Bootblöcke, die auch mit »ZeroVirus« gespeichert wurden, berücksichtigt werden.

Für Leute, denen der Sinn nach viel Farbe steht und die die vorhandenen nicht mögen, ist die nächste Funktion von großer Bedeutung. »Adjust Colours« konfrontiert Sie mit einem Farb-Requester. Durch Anklicken eines der acht Farb-Gadgets im oberen linken Viertel des Bildschirms wählen Sie die Farbe aus, die Sie verändern möchten. Die ersten drei Schieberegler bestimmen den Anteil der RGB-Farben. Die letzten drei Regler sind für Helligkeit, Farbsättigung und Brillanz verantwortlich. In dem breiten Balken können Sie die Veränderungen beobachten und auch die Anteile von Rot, Grün und Blau ablesen. Die Funktion »Copy« erklärt sich, glaube ich, von selbst, mit »Spread« werden Farbverläufe ermöglicht. Dazu klicken Sie einfach auf die erste Farbe, dann auf »Spread« und zu guter Letzt auf die Farbe, die die dunkelste in Ihrem Farbverlauf sein soll. »Reset« setzt alles wieder in den Urzustand zurück. Falls Sie sich »verlaufen« haben, genügt ein »Druck« auf »Cancel«. Mit »OK« werden Änderungen übernommen. »Save« akzeptiert die Farben und speichert die Einstellung in das Brainfile von »ZeroVirus« (ZeroVirus.Palette). Dieses File muß natürlich auch wieder im Verzeichnis »S:« oder im aktuellem Directory stehen.

Das klassische »About« fehlt natürlich nicht in diesem Programm und gibt (wie sollte es auch anders sein) Informationen über Autor und Programmpreis. Des weiteren erfahren Sie hier, welche Brainfile-Version Sie benutzen.

Die Option »Iconify« ist eigentlich das »Beste« an »ZeroVirus«. Man kann damit das Programm auf die Minimalmaße eines Windows beschränken. Wenn Sie nun eine Ähnlichkeit zu »VirusX« feststellen, ist dies durchaus berechtigt. Jedoch verfügt »ZeroVirus« zusätzlich über eine Uhr und zeigt den noch verfügbaren Chip- und Fast-RAM-Vorrat an. Wird nun eine neue Diskette in irgendein Laufwerk eingelegt, wird diese sofort überprüft, und falls sich ein »Non-Standard Bootblock« auf ihr befindet, erscheint ein Requester, der es

Ihnen ermöglicht, ins Hauptprogramm zurückzukehren. Wollen Sie einfach so »ZeroVirus« von der Workbench aufrufen, dann klicken Sie einfach in das Close-Gadget. Falls Sie »ZeroVirus« nun in Ihr System installieren wollen, gibt es die Möglichkeit, das Programm schon im Iconify-Modus aus der Startup-Sequence aufzurufen. Sie können am besten als ersten Eintrag in Ihrer Startup-Sequence »ZeroVirus -i« eingeben; dadurch wird das Programm als erstes gestartet. »Exit from ZeroVirus« beendet das Programm.

Kurz und knapp – File-Requester

Wer sich mit dem File-Requester noch nicht angefreundet hat, für den folgt ein kurzer Crash-Kurs. Durch die Datei-abfrage können Sie sehr bequem und einfach jedes Directory und File auf Ihrem Amiga anwählen. Die einzelnen Felder im schnellen Überblick:

- »Directory«: Hier können Sie Dateien (Files), Verzeichnisse (Directories) oder Laufwerke (Volume) auswählen. Die Auswahl erfolgt entweder durch Blättern (Scrollen) durch die Liste und Anklicken des gewünschten Files, Directories und so weiter. Sie können aber den Namen auch direkt in die Drawer-Zeile eintippen.

- »Parent«: Hiermit gelangen Sie in die nächsthöhere Verzeichnissstufe.

- »Okay«: Verläßt den Requester und akzeptiert Ihre Auswahl.

- »Cancel«: Verläßt den Requester, akzeptiert Ihre Auswahl jedoch nicht.

- »Rename«: Die aktuelle Datei kann umbenannt werden.

- »Delete«: Die aktuelle Datei wird gelöscht (VORSICHT!).

- »Make Dir«: Erstellt ein Unterverzeichnis im aktuellen Verzeichnis.

Um schnell durch die Listen zu blättern, haben Sie einen Regler und für Einzelschritt die Auf- und Ab-Gadgets. Die Verzeichnisse werden bei einem Diskettenwechsel automatisch aktualisiert.

Zusätzlicher Helfer

Damit könnte unser kurzer Rundgang durch das Programm »ZeroVirus« beendet

werden, aber ein Zusatz-Utility wartet noch auf uns. Es handelt sich um »UDBF«. Dieses Programm gestaltet das Updaten von Brainfiles wesentlich einfacher, als dies von Hand geschieht.

Durch die Option »Learn« werden Sie Ihre eigenen Bootblöcke dem Brainfile von »ZeroVirus« zuführen. Werden nun neue Brainfiles von dem Autor herausgegeben, so enthalten diese natürlich nicht Ihre eigenen Bootblöcke. Zwei Möglichkeiten stehen Ihnen offen. Sie lesen alle Bootblöcke noch einmal ein, oder Sie benutzen »UDBF« (UpDate BrainFile). Dieses Programm sorgt übrigens dafür, daß kein Bootblock in dem neuen Brainfile wiederholt wird, wie dies beispielsweise bei dem DOS-Befehl »Join« der Fall wäre. »UDBF« ist sehr einfach zu bedienen.

Nach dem Start erscheinen drei Gadgets mit den Namen: »First Brainfile«, »Second Brainfile« und »Output Brainfile«. Dort geben Sie nun mit vollständiger Pfadangabe die einzelnen Filenamen ein und klicken dann auf »Update Brainfile«. Sie setzen zunächst hier unter dem Menüpunkt »FBF« Ihre alte Datei ein. Die neue geben Sie als »SBF« an und erhalten als Ergebnis dann »S:ZV.BF«.

Ich rate Ihnen, Ihr Update immer im Directory »S:« abzulegen, da Sie dann »ZeroVirus« aus jedem Unterverzeichnis starten können, ohne daß es zu Komplikationen kommt. Hat die Ursprungsdatei den gleichen Namen wie das Update, gibt es mit »UDPF« keine Probleme. Diese Klippe meistert das Programm eigenständig.

Wir hoffen, Ihnen mit diesem Workshop den Virenkiller »ZeroVirus« nähergebracht zu haben, und wünschen Ihnen noch viel Erfolg bei der Virenjagd.

(vb)

Autor von ZeroVirus:

Jonathan Potter
3 William Street
Clarence Park S.A. 5034
Australia

Sie finden dieses Programm auf folgenden Disketten:

- Fish 242 (Version 1.3)
- KickPD 218 (Version 1.3)

AMIGA-DOS-Pixel-Panorama

Wer Amiga sagt, der meint auch Grafik damit. Kein Wunder, denn diese Maschine birgt Möglichkeiten, durch die so gut wie jedermann mit ein wenig Geduld feine Bilder erstellen kann.

Wer nicht zu den begabten Menschen gehört, die mit Stift und Farbe oder mit Maus und Monitor Bildwerk erstellen können, sich aber trotzdem gerne damit beschäftigen möchte, wird bald auf Raytracing und 3D-Konstruktions-

Impression aus den Frühtagen der Welt wurde mit Deluxe Paint III in Szene gesetzt. Sollten Sie durch das Betrachten der Bilder Lust bekommen haben, sich selbst einmal an bunter Grafik zu versuchen, lassen Sie es uns wissen. Nach wie vor steht das Pixel-



Raytrace-Spiegelei von M. Krywdzinski

Panorama Ihren Einsendungen offen. Also, wenn Sie gute Amiga-Bilder zeichnen, eine Diskette formatieren, Bilder draufkopieren, vermerken, welches Grafikformat und Programm verwendet wurde, und an folgende Adresse schicken:

DMV-Verlag,
Redaktion AMIGA DOS
Kennwort Pixel
Postfach 250
3440 Eschwege

Für Bilder, die wir im Rahmen des Pixel-Panoramas veröffentlichen, winkt ein Dankeschön in Form eines Software-Paketes im Wert von zirka 100 DM.

(hs)



“Der Sonne entgegen“ von Joachim Heck

programme stoßen und entdecken, daß sich auch damit ungeahnte Kreativität entfalten läßt. Daß dabei nicht immer nur geometrisch strengen Formen geformt werden muß, das zeigt das Bild “Spiegelei“ von Martin Krywdzinski. Das Bild wurde mit Reflections erstellt.

Völlig anderes hatte Joachim Heck mit seinem Bild “Der Sonne entgegen“ im Sinn. Ganz ungegenständlich spielen hier Farbe, Form und Reflektion zusammen. Hier wurde mit Digi und Photon Paint gearbeitet.

Der grauen Vorzeit hat sich Frank Plohmman verschrieben und präsentiert in seinem Bild “Primitive Times“ Schachtelhalme, Rieseninsekten und Dinosaurier. Diese



Vorzeit von Frank Plohmman

Datenfernübertragung, kurz DFÜ genannt, erfreut sich wachsender Beliebtheit. Leider gibt es nur wenige gute deutschsprachige Programme. Diese Lücke wird nun von »AmigaCall« geschlossen. Das Programm wird als Bookware angeboten, das heißt Buch und Programm auf Diskette in einer Packung.

AmigaCall

Das Buch zum Programm »AmigaCall« setzt sich aus fünf Kapiteln und einem Anhang zusammen. Im ersten Kapitel wird kurz, aber prägnant erklärt, wie man eine Sicherheitskopie anfertigt, denn »AmigaCall« ist nicht kopiergeschützt, was als begrüßenswert einzustufen ist.

Kapitel zwei befaßt sich mit den auf dem Bildschirm sichtbaren Bedienungselementen. Wichtige Einstellungen sind auf der Menüleiste ständig sichtbar und können auch während einer Datenübertragung geändert werden. Dazu klickt man sie einfach mit dem Mauszeiger an. Auf diese Art und Weise kann die Baudrate (300 bis 19200) verändert, das Logbuch geschlossen oder geöffnet und die Uhr zurückgestellt werden.

Etwas Besonderes hält »AmigaCall« bei den Pull-down-Menüs parat. Anstatt mit dem Mauszeiger bis hoch zur Menüleiste fahren zu müssen, genügt hier ein kurzer Druck auf den rechten Mausknopf, um sofort ein übersichtlich gestaltetes Bildschirmmenü erscheinen zu lassen. Alle Parameter und Einstellungen können nun über weitere Untermenüs ausgewählt werden.

So gibt zum Beispiel ein integrierter Gebührenzähler, dessen Werte individuell eingestellt werden können, ständig die Anzahl der bereits vertelefonierten Einheiten an.

Eine weitere lobenswerte Eigenschaft ist das »Textklicking«. Angenommen, das Auswahlmeneü einer Mailbox verlangt die Eingabe einer Zahl, um eine bestimmte Funktion auszuführen. Anstatt auf der Tastatur herumzuhacken, wird ganz einfach die auf dem Bildschirm erscheinende Zahl angeklickt. Durch »Textklicking« wird also jedes auf dem Bildschirm erscheinende Zeichen zurückgesendet. Häufig benutzte Phrasen, die länger als ein Zeichen sind (zum Beispiel Passwörter) können komforta-

bel auf Funktionstasten gelegt werden.

Im dritten Kapitel wird nun näher auf die einzelnen Parameter eingegangen, ohne die eine Datenübertragung mit Modem oder Akkustikkoppler nicht möglich sind. Dies sind zum Beispiel Baudrate, Parität, Stopbits, CR und Selbstwahl-Parameter für das Modem.

läßt »AmigaCall« keinen Anwender im Stich.

Im fünften Kapitel plaudern die Autoren dann ein wenig aus dem Nähkästchen. Deshalb heißt dieses Kapitel auch »Insider-Informationen«. Hier bekommt der Leser endlich erklärt, wie er zum Beispiel kostenlos telefonieren kann – vorausgesetzt, die Deutsche



Auch diese Parameter werden über ein Bildschirmmenü verändert.

Mit den diversen Übertragungsprotokollen beschäftigt sich das vierte Kapitel. Hier wird erklärt, wie man ASCII-Texte sendet oder Binärdaten (zum Beispiel lauffähige Programme) empfängt (Download) oder sendet (Upload). Weiterhin stehen Funktionen wie Paket-Modus, Ändern von Empfangs- und Sendezeichen sowie ein Telefonbuch, von dem aus mit dem Modem ein Teilnehmer ausgewählt werden kann, zur Verfügung. Ganz bequeme DFÜler schreiben sich ein Script-File, das den Wahlvorgang zu einer Mailbox, das Heraussuchen von Daten und ähnliches zur billigsten Telefonierzeit erledigt. Auch hier

Bundespost ist damit einverstanden...

»AmigaCall« konnte im Test alle gestellten Erwartungen erfüllen.

Hier liegt ein Programm vor, das neben leichter Bedienbarkeit auch ein hohes Maß an Sicherheit vorweist. Das dazugehörige Buch ist locker geschrieben und leicht verständlich. »AmigaCall« stellt somit ein ideales Programm für Einsteiger dar.

(tb)

Name: AmigaCall
Autor: Atlantis Production
Verlag: Markt & Technik
ISBN: 89090-716-4
Preis: 99,- DM

Animationen sind wohl eins der Spezialgebiete auf dem Amiga, mit der sogar Tante Hildchen beim Anstandsbesuch zu einem erstaunten »Oh!« bewegt werden kann. Mit der Bewegung ganz anderer Objekte, nämlich graphischer, befaßt sich »Das Große Animationsbuch«. Es wird fast ausschließlich auf Programme eingegangen, die mit vektorisierten, dreidimensionalen Objekten arbeiten.

Das große Animationsbuch

Bei dem ersten Programm handelt es sich um »VideoScape 3D«. In fünf Kapiteln mit über 100 Seiten wird dem interessierten Leser, beginnend mit dem Prinzip der Vektorgrafik bis zum Erstellen einer kompletten dreidimensionalen Animation, der Einstieg in die Funktionsweise von »VideoScape 3D« erläutert. Dies geschieht in einem ungezwungenen, aber keineswegs lächerlichen Schreibstil, der das Lesen zur Unterhaltung macht. In ähnlicher Weise werden die anderen drei beschriebenen Programme abgehandelt: »Turbo Silver«, »DBW Render« (zwei Raytracer) und »VideoEffects 3D«, ein IFF-Animationsprogramm. Bemerkenswert ist, daß hier auf ein Public-Domain-Programm eingegangen wird (»DBW Render«, zu finden auf Chiron 16). Dies ist, nebenbei bemerkt, wieder einmal ein Beweis dafür, daß PD-Programme kommerziellen Programmen durchaus das Wasser reichen können.

Alles in allem ist »Das Große Animationsbuch« eine sehr gute Einsteigerhilfe, die außerdem zu einem Preis angeboten wird, der selbst bei den »teuren« Animationsprogrammen noch vertretbar ist.

(tb)

Name: Das große Animationsbuch
Autoren: Polk, Schulz
Verlag: DATA-Becker
ISBN: 3-89011-332-X
Preis: 39,- DM

Wer hat gesagt, daß Computerbücher immer bierernst sein müssen? Man kann seinem Hobby oder

seinem Beruf immer das gewisse Quentchen Humor abverlangen, das den Arbeitstag erträglich macht. "Computer total verrückt" kümmert sich überhaupt nicht um Ports, Devices, Drucker oder Assembler, sondern einzig um die Frage, wie man mit der elektronischen Nervensäge fertig wird, ohne selbst den Verstand zu verlieren. Das Buch zeigt die komischsten Situationen, die man sich als Computerbesitzer denken kann, als witzig aufgemachte Cartoon-Zeichnungen.

An irgendeiner Stelle hat jeder, der das Buch liest, das Gefühl "das kenn ich doch", genannt "Déjà vu". Ob nun der Programmierer schonend darauf hingewiesen werden muß, daß er schon seit Stunden Daten in ein defektes Speichermedium eingibt, oder der Oberst samt Adjutant vor einer ferngelenkten und (natürlich) prozessorgesteuerten Rakete davonlaufen, der Navigator einer Flugzeugmannschaft während des Landens statt der Navigationsdaten lieber die Börsenkurse abliest oder sich die Raumschiffe eines Star-Wars-Spiels wutentbrannt auf den Spieler stürzen, so daß dieser auf die Straße flüchten muß, die Situationen reizen gerade zum Anstellen von Vergleichen mit dem eigenen Alltag.

Computer total verrückt

Und wer jetzt immer noch Schwierigkeiten hat, seine Mundwinkel zu verziehen, kann das Buch ja seinem Partner geben, damit die oder der jetzt endlich kapiert, mit wem man es zu tun hat. "Computer total verrückt" sollte ruhig neben den "seriösen" Werken der Computerliteratur stehen.

(jb)

Name: Computer total verrückt

Autor: Daniel Le Noury

Verlag: Sybex

ISBN: 3-88745-042-6

Preis: 12,80 DM

Wenn man Amiga-Profi werden will, kommt man um den Kauf diverser Literatur nicht herum. Und diese "diverse Literatur"

führt oft zu den bekannten Suchszenen à la "Wo stand denn noch die Library xyz?". Das heißt, irgendwann wird der leidgeprüfte Leser nicht mehr aus noch ein wissen. Hilfe verspricht da das neue "Amiga intern" von DATA Becker. Als Standardwerk rund um den Amiga will sich das Buch, an dem sechs Autoren mitgewirkt haben, verstehen. Insgesamt vier Hauptthemen bilden dabei das Gerüst für die einzelnen Kapitel: Hardware, Systemsoftware, IFF-Format und Grundstrukturen.

Amiga intern

Im Bereich Hardware werden die Unterschiede und Besonderheiten der verschiedenen Amigas erläutert, sowie die Customchips unter die Lupe genommen. Die Pinbelegung, Registertabellen und Blockschaltbilder gehören ebenso dazu wie die entsprechenden Speicherstellen, an denen die Register der Chips eingeblendet werden. Ebenso findet der Leser alle Informationen über die Ports des Amiga und die Hardware-Programmierung.

Thema zwei, die System-Software, zeigt den Umgang mit Devices und Libraries sowie die Speicherverwaltung. Interrupt-Programmierung und eine Unterweisung in der Programmierung von reset-festen Programmen finden hier ebenfalls Platz wie auch Informationen über den Aufbau der Boot-Prozedur.

Ein weiteres Thema bildet das IFF-Format mit seinen Unterformaten für Text, Grafik und Musik. Der Leser bekommt hier die Unterschiede zwischen den Unterformaten IFF/LBM-, FTXT-, ANIM-, und 8SVX-Format vor Augen geführt. Das letzte Thema "Grundstrukturen" beschäftigt sich mit programmtechnischen Problemen und deren Lösung. So wird der Leser an die Erstellung von eigenen Druckertreibern oder Font-Dateien herangeführt.

Dies ist in Grundzügen der Inhalt von "Amiga intern"; in Grundzügen wohlgerneht, denn alle Informationen aus dem Buch hier darzulegen, ist aus Platzgründen nicht möglich. Das Buch richtet sich an alle Amiga-Besitzer, darunter auch die Nicht-Profis. Die allerdings werden am Anfang Schwierigkeiten haben, sich mit dem Buch anzufreunden.

Und zwar nicht, weil es schlecht zu lesen wäre, sondern weil hier wieder Kenntnisse in den Sprachen C oder Assembler vorausgesetzt werden und dies sogar schon im Einleitungsteil. Auch sind Fehler bei der Benennung der im Amiga enthaltenen Bausteine zu finden, so wird aus dem Portbaustein 8520 schon mal ein 8250. Dies wäre nicht so schlimm, wenn es den 8250 nicht gäbe, aber das Gegenteil ist der Fall, und ein unter Zuhilfenahme des Buches eingesetzter falscher Baustein kann noch teurer werden.

Ansonsten macht das Buch einen guten Eindruck; wer öfters Fragen hat und ein Nachschlagewerk benötigt, ist mit dem "Amiga intern" gut bedient. Einzig der Preis bietet noch Anlaß zur Kritik; 98,- DM sind für ein Buch dieser Art, das zudem ohne Disketten ausgeliefert wird, zu hoch. Man mag zwar sagen, daß man hier für knapp hundert Mark in kompakter Form all das bekommt, was man sonst einzeln kaufen müßte, aber wer kauft sich schon zu seinem Lieblingsthema nur ein einziges Buch?

(jb)

Name: Amiga intern

Autoren: Bleek, Dittrich, Gelfand, Jennrich, Schemmel, Schulz

Verlag: DATA Becker

ISBN: 3-89011-398-2

Preis: 98,- DM

Als Programmierer des Amiga kommt man öfters in die Nöte, gewisse System-Datenstrukturen genau dann nicht im Kopf zu haben, wenn man sie am nötigsten braucht. Nun ist man ja nicht auf den Kopf gefallen und hat sich schon seine Arbeitsbibliothek angelegt – nur wo suchen und schnell finden?

Amiga Datenstrukturen-Lexikon

Hier kommen nun die reinen Nachschlagwerke zum Zuge, die wenig oder keinen Text zu bieten haben, sondern reine Informationen tabellenartig darlegen. Das "Amiga Datenstrukturen-Lexikon" ist genau so ein Buch, das nach Mei-

nung der Autoren bei der Programmierarbeit gleich neben dem Amiga auf dem Tisch liegen sollte. Was findet man nun im Buch? Nun zuerst einmal einleitende Worte, zu denen auch Tips zum Gebrauch des Buches bei den einzelnen Programmiersprachen gehören. Das ist ja das Schöne, daß alle Datenstrukturen von jeder Programmiersprache genutzt werden können – bei der einen so, bei der anderen so. Diese erklärenden Kapitel, die von Programmierbeispielen unterstützt werden, sind zwar schon von anderen Büchern her bekannt, trotzdem ganz nützlich, wenn man sich einige entfallene Details noch einmal merken will. Und dann geht es auch schon los; wichtige Flags und die Datenstrukturen der »Exec-Library«, »Intuition-Library«, »Graphics-Library« und »DOS-Library« werden aufgezeigt und erklärt. Auch die restlichen Libraries des Amiga-Systems finden hier Platz wie auch die Devices. Zeichnungen lockern nicht nur das Buch auf, sondern helfen beim Verstehen der Thematik. Im Anhang findet sich ein Glossar, in dem die wichtigsten Begriffe rund um die Datenstrukturen noch einmal kurz und knapp erläutert werden, ein Literatur- und Abbildungsverzeichnis, sowie eine Quick-Reference schließen sich an. Für diejenigen, die schlußendlich alles im Kopf haben, gibt es noch einmal ein alphabetisches Verzeichnis der Strukturen.

Das "Amiga Datenstrukturen-Lexikon" ist kein Buch für Einsteiger in die Programmierung, es kann dort höchstens als Ergänzung zum eigentlichen Lehrmaterial dienen. Für Programmierer allerdings kann es durchaus als empfehlenswert gelten, wenn nicht allzusehr Wert auf ausführliche Exkurse in die Amiga-Programmierung gelegt wird. Hinsichtlich der Vorgabe, geballte Information an den Leser zu bringen, haben die Autoren auf jeden Fall ihr Klassenziel erreicht.

(jb)

Name: Amiga Datenstrukturen-Lexikon

Autor: Paul Lukowicz, Olaf Pfeiffer

Verlag: Markt & Technik

ISBN: 3-89090-250-2

Preis: 69,- DM

Ahnungslos schob ich die Diskette Nr. 41 von Nordsoft in das Laufwerk und wartete wie gewöhnlich den Bootvorgang ab. Nachdem ich den Eindruck hatte, der Bootvorgang sei beendet, wendete ich mich mit dem Drehstuhl meinem Amiga-Bildschirm zu und war über alle Maßen erstaunt. Es war nicht ein Guru von dem mein Erstaunen zeugte, sondern die Präsentation des Bildschirms, auf dem sich sage und schreibe vierzig Utilities anklickbereit durch die Maus tummelten.

Bei solch einer Vielzahl von Helferlein ist es natürlich eine Freude, den Viren den Krieg zu erklären. Dazu stehen Ihnen folgende Programme auf dieser Disk zur Verfügung: »IRQ-Killer 1.0«, »Virus-Protector«, »Virus-Killer 2.3« und »VirusX 3.2«. Der »IRQ-Killer« wird durch Befehlseingabe im CLI gestartet. Nach seiner Aktivierung entfernt er den IRQ-Virus, falls er vorhanden ist. Was dem IRQ-Killer recht ist, soll dem »Virus-Protector« längst billig sein. Denn dieses Programm schützt Sie vor dem »BGS9-Virus« (Bundesgrenzschutzvirus). Nach dem Einladen aus dem CLI wird man lediglich dazu aufgefordert, den Maus-Button zu klicken, worauf der »Virus-Protector« ein Antivirusprogramm auf Ihrer Diskette installiert. Beim »Virus-Killer 2.3« öffnet sich ein Fenster. Hier stehen dem Benutzer Optionen zur Verfügung, mit deren Hilfe man Bootblöcke sowohl installieren als auch auslesen kann. Dazu wird der soeben ausgelesene Bootblock auf dem Bildschirm per ASCII aufgelistet.

Beim Arbeiten mit »Virus-Killer« können die Laufwerke DF0 bis DF3 angesprochen werden. Mit »VirusX 3.2« präsentiert sich wieder einmal ein Antivirusprogramm der VirusX-Serie, aus der wir in früheren AMIGA-DOS-Ausgaben schon Programme vorgestellt haben. So weit zu den Viren.

Unter den vielen Programmen befindet sich auch »Quickmonitor«. Auf dem Bildschirm sind die jeweiligen Speicheradressen und deren Speicherinhalte im Hex-Type und im ASCII-Code angegeben. Mit Hilfe der Cursortasten sind Sie in der Lage, den gesamten Bildschirm nach oben und nach unten zu scrollen, womit Sie eine Verschiebung der

Die PD-Werkzeugkiste

Für wenig Geld viel Qualität und Quantität, so präsentieren sich Public-Domain-Disketten. Wir stellen Ihnen wieder zwei Programmsammlungen der Extraklasse vor, die mit einer Vielzahl unentbehrlicher Helfer vollgepackt sind.

Speicheradressen bewirken. Damit Sie nicht den halben Speicher durchforsten müssen, um einen bestimmten Wert ausfindig zu machen, steht Ihnen eine Find-Option zur Verfügung, die den Speicher nach dem Wert durchsucht. Des weiteren steht es Ihnen frei, direkt zu einer Speicheradresse zu springen. Dies bewirken Sie mit der Option »M« (New Memory Address).

Wenden wir uns nun den Crunchern zu. Die Auflistung der folgenden Komprimierungsprogramme wird Ihnen sicherlich sehr schnell verdeutlichen, daß wir Ihnen nicht alle Programme vorstellen können:

»DragpackV1.0«, »Implodecruncher«, »PowerPacker2.2«, »Tetrapack2.2«, »TriangleV1.7«, »TSK-Cruncher«, »HouseCruncherV1.4«, »Mastercruncher3.0«, »SMF-Compactor«, »TNM-Cruncher«, »Tryit«. Zum Programm »Mastercruncher« baut sich ein schlichtes Optionsfeld auf. Auf dem Bildschirm sind nun fünf Gadgets gesetzt, mit denen man die einzelnen Programme einladen, komprimieren und letztlich abspeichern kann. Dazu bedarf es natürlich einiger Voreinstellungen, die Sie durch das Anklicken der Option »Preferences« er-

reichen. In einem ebenso schlichten Kleid präsentiert sich das Packerprogramm »Triangle«. Hier werden Sie lediglich dazu aufgefordert, das zu komprimierende File einzugeben.

Nachdem wir Ihnen zwei Packer vorgestellt haben, kommen wir nun zu Utilities, deren Aufgabe darin besteht, dem Benutzer ein möglichst komfortables DOS zu bieten. Zu diesen Programmen gehört auch »ATool«. Zunächst sollte erwähnt werden, daß es sich bei diesem Programm um Shareware handelt. Das heißt, möchte man dieses Programm benutzen, sollte an den Autor M. Hoffmann ein Obulus von 10 DM entrichtet werden. Nach dem Starten des Programms und dem »Reseten« des Amigas, erscheint vor dem neuen Bootvorgang ein Menü, das sechs Funktionen implementiert. So besteht die Möglichkeit, durch Funktionstaste »Eins« das Fastmemory abzuschalten, was das Einfrieren der eventuell vorhandenen Speichererweiterung bewirkt. Mit der Aktivierung der Bootblockanzeige wird der Bootblock in ASCII-Format angezeigt, so daß man sogar die verborgensten Viren aufstöbern kann. Die dritte Option »Autoboot-Routine« ermöglicht das Booten von al-

len angeschlossenen Laufwerken. Die Standard-Option dient auch der Bekämpfung der Viren. Falls der Bootblock nicht mit den Standard-Routinen bestückt ist, deklariert der Computer den Bootblock als Virus. Die sogenannte Copy-Routine ist nur mit zwei Laufwerken lauffähig. Innerhalb von 90 Sekunden wird nun eine DOS-Diskette kopiert. Mit dem Tool »Click-Dos« lassen sich drei externe Laufwerke, eine Harddisk und eine RAM-Disk ansprechen. Dazu öffnet sich ein Fenster, das zwei File-Requester zur Anzeige des Directory enthält. Die Unter-Directories sind im Gegensatz zu den »einfachen« Programmnamen markiert. Nun können Sie mit Hilfe der Maus die wichtigsten Befehle über Gadgets ausführen. Im Programm sind folgende Befehle enthalten: »MakeDir«, »Rename«, »Copy«, »Move«, »Delete«, »NewCLI«, »Show« und »Type«.

Wie nicht anders zu erwarten ist, fehlt selbst ein Programm zur Belegung der Funktionstasten nicht. Bei »SetKey« erstreckt sich über den gesamten Bildschirm eine Abbildung der Amiga-Tastatur, damit die einzelnen Tasten umbelegt werden können. Per Mausclick wird nun die neu zu belegende Taste gewählt. Danach werden Sie aufgefordert, den neuen Ausdruck für die Taste oder gar die Tastenkombination zu definieren. So können Sie diese durch Anwählen der Funktionstasten mit DOS-Befehlen bestücken. Nach Ihrer Umgestaltung kann die neu definierte Tastatur als »Keymap« abgespeichert werden, so daß die Installation der neuen Tastatur auf jeder Diskette vollzogen werden kann.

Neben diesen kurz vorgestellten Utilities ist diese Diskette, wie bereits erwähnt, vollgepackt mit vielen, nützlichen Helferlein. Diese Sammlung darf bei keinem Amiga-User in seinem PD-Reservoir fehlen.

(Jürgen Seibel/vb)

Name: Special 41
Quelle: Nordsoft

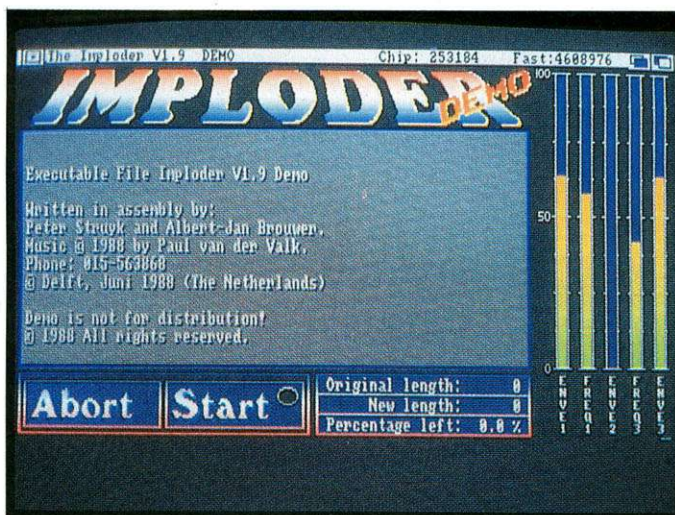


Bild 1. Eins der vielen kleinen Helferlein: der »Implodecruncher«

Mit »Image Ed V2.2« bekommt man ein Utility, mit dem man Images, Sprites, Bobs oder Icons selbst

erstellen und diese dann als Source-Code für die Programmiersprachen BASIC, C, Assembler oder Modula2 auf Diskette speichern und später in eigene Programme einbinden kann. Das Programm wird komplett mit der Maus bedient. Selbst bei der Auswahl eines Files werden Sie von Requestern unterstützt, so daß Sie es nur noch anzuklicken brauchen, um es zu laden.

Das Programm kann mit einer Vielzahl von Möglichkeiten zum Erstellen Ihrer Bilder, die schon an ein gutes Malprogramm erinnern, aufwarten. So sind die Funktionen »Kreis«, »Rechteck«, »Linie«, »Freihand« und »Füllen« ebenso selbstverständlich wie Spiegeln, Kopieren, Vergrößern und Verkleinern von Ausschnitten.

Anwählen können Sie die Funktionen entweder von der Menüleiste, oder Sie klicken einfach am Bildschirmrand an, was natürlich wesentlich einfacher und schneller ist.

Auch das Auswählen der Farben erfolgt durch einen Mausklick am Bildschirmrand. Verändern können Sie die Farben in einem eigens dafür vorgesehenen Window.

Bis hierhin konnte »Image Ed« eigentlich überzeugen. Doch leider hat das Programm auch eine Tücke, die man nicht unerwähnt lassen sollte. Die Zeichenfläche ist sozusagen eine Lupe, das heißt, alles, was Sie im Augenblick malen, ist größer dargestellt. Wenn Sie jetzt die Funktion »Show Image«

anwählen, können Sie Ihr Meisterwerk in einem kleinen Window in Originalgröße bestaunen. Doch aus dem Bestaunen wird schnell ein Erstaunen, denn was auf der Zeichenfläche beispielsweise noch quadratisch oder rund war, ist in Originalgröße rechteckig oder oval. Das XY-Verhältnis der Zeichenfläche stimmt also in keiner Weise mit dem des Windows, in dem die Originalgröße gezeigt wird, überein. Um solchen Überraschungen aus dem Weg zu gehen, sollte man gleich zu Anfang die Funktion »Show Image« aktivieren, um so im kleinen Window sein Werk kontrollieren zu können.

»Filemaster (Version 1.11)« ist ein File-Editor ähnlich »New-Zap«. Mit diesem Programm kann man auf einfachste Weise Bytes eines Files manipulieren.

»Filemaster« ist sehr übersichtlich aufgebaut und komplett mausgesteuert. Die wichtigsten Funktionen sind über Gadgets am Bildschirmrand anwählbar, die nicht so oft benötigten erreicht man über die Menüleiste. Das Auswählen eines Files, das man bearbeiten will, erfolgt mit Hilfe eines File-Requesters. Hier kann man auch entscheiden, ob die Daten direkt auf der Diskette verändert werden sollen oder ob das File vollständig ins RAM geladen wird und dort Ihrem Tatenrang zur Verfügung steht.

Umfassen die Daten eines Files mehr als eine Bildschirmseite, so teilt »Filemaster« diese in mehrere Sektoren, sprich

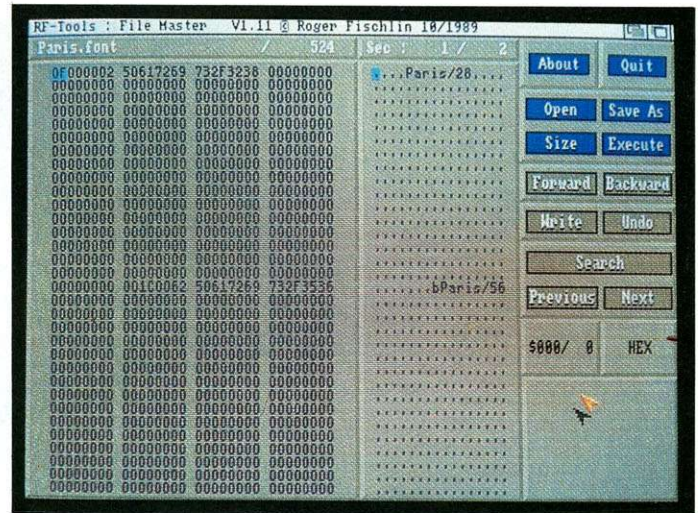


Bild 2. »File Master« ist ein File-Editor, mit dem sich Bytes eines Files manipulieren lassen

Bildschirmseiten, die durch Weiterblättern oder direktes Anwählen erreicht werden. Mit der Suchfunktion können Sie die Daten nach bestimmten Zahlen oder Zeichen durchforsten. Erstreckt sich die zu durchsuchende Datei über mehrere Bildschirmseiten, werden beim Weiterblättern automatisch die gesuchten Zeichen angezeigt. Man braucht also nicht bei jeder neuen Seite wieder den Such-String einzugeben. Fehleingaben können mit »Undo« wieder rückgängig gemacht werden. Die Sharewaregebühr beträgt 10,- DM und ist für dieses Programm keineswegs zu hoch.

Falls Sie es leid sind, Ihre Programme von der Workbench oder vom CLI aus zu starten, dann

ist »Menü« genau das richtige für Sie. »Menü« ist ein Auswahlprogramm, mit dem man Programme oder Batch-Dateien aus der Startup-Sequence heraus anwählen kann. Es kann bis zu elf verschiedene Programme anbieten. Das Erstellen eines solchen Auswahlmenüs stellt kein Problem dar. In einer Textdatei werden einfach die Namen der Programme (inklusive Pfad) angegeben, die später auf dem Auswahlbildschirm erscheinen sollen. Nach dem Booten mit dieser Diskette erscheint dann das Menü.

(Ralf Zuber/vb)

Name: Antares 48
Quelle: Rüdiger Dombrowski

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
Friedrich Neuper
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
Dombrowski, Rüdiger
Postfach 71 04 62
2000 Hamburg 71
Tel.: 040/6428225

Herrmanns & Kommelter
Vom-Bruck-Platz 45
4150 Krefeld 1
Tel.: 02151/399833
Hieske, Dieter
Schillerstr. 36
6700 Ludwigshafen
Tel.: 0621/673105
Kaminski, Walter
Geilenkirchener Str. 4
5110 Alsdorf
Tel.: 02404/22963
Keim, Peter
Vogelsanger Str. 34
5000 Köln 30
Tel.: 0221/520765
Maxon Computer GmbH
Industriestr. 26
6236 Eschborn
Tel.: 06196/481811

Nordsoft
Schwenecker & Behnke
Heidelbergstr. 75
2800 Bremen 21
Tel.: 0421/611430
Pawlowski, Patrick
Software Service
Ellerbruch 19
2177 Wingst
Tel.: 04778/7294
Software Empire
Lange Str. 118
2880 Brake (Utw)
Tel.: 04401/71348
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

M.A.R.-Computershop
A-1100 Wien
Weldengasse 41
Tel.: 0222/621535

Schweiz

Mailsoft Switzerland
Alex Brander
Feldstr. 20
CH-8330 Pfäffikon ZH
Tel.: (0) 1 950 5611

Kommerzprodukte und PD, eigentlich passen diese beiden Dinge nur schlecht zueinander. Trotz der grundverschiedenen Ideen, die hinter diesen beiden Software-Konzepten stecken, existieren Berührungspunkte. Kommerzielle Anbieter machen sich die Eigendynamik der PD-Szene zu nutze.

In der Regel erfreuen sich neue PD-Disketten innerhalb kürzester Zeit weltweiter Verbreitung. Die Geschwindigkeit, mit der hier Programme wandern, läßt sich nur noch mit dem Markt der Raubkopien vergleichen, der zwar ähnlich effektiv verbreitet wie Public-Domain-Strukturen, kommerziellen Anbietern jedoch ein Dorn im Auge ist.

Irgendwann dachte sich ein Hersteller, daß es doch Möglichkeiten geben müßte, PD-Strukturen auch für kommerzielle Anbieter nutzbar zu machen; die Idee der PD-Demodiskette war geboren.

Das Prinzip ist einfach: Der Hersteller, der eines seiner neuen Spiele der Amiga-Anwenderszene vorstellen möchte, stellt eine Demoversion her, die in die PD gegeben wird und sich dort fast automatisch verbreitet.

Wir haben uns ein wenig in diesem Bereich umgesehen und einiges Interessantes gefunden.

»XENON II« läßt das Herz jedes Ballerspielfreundes höher schlagen. Den Spieler erwartet ein vertikal scrollendes Szenario voller phantastischer Formen und Gegner. Es gilt, ein zunächst noch mäßig ausgestattetes Raumschiff durch zerklüftete Felsformationen, korallenartige Baum-



»XENON II«, auch als PD-Demo ein echter Megablast

Kommerzspiele als Public-Domain-Demos

Wer Spiele gerne mag und sich ab und zu auch einmal ein Programm kauft, der wird sich schon oft gewünscht haben, Spiele einmal vor dem Kauf testen zu können. Unglaublich, aber auch hier hilft Public Domain weiter.

strukturen und andere Barrieren zu steuern. Dabei muß der Spieler versuchen, den Legionen anfliegender Aliens und deren abgefeuerten Schüssen auszuweichen.

Diese Demoversion enthält die komplett spielbare erste Hälfte des ersten Levels. Hat man sich hier durchgekämpft, folgt ein Laden mit Extras, die zum größten Teil auch gekauft werden können. Die Demoversion ist absolut empfehlenswert, auch wenn man sich das Spiel nicht unbedingt noch kaufen möchte.

»Lords of the rising Sun« entführt in das feudalistische Japan des 16. Jahrhunderts. Damals rangen viele kleine Feudalherren um die Macht auf der Insel der aufgehenden Sonne. Jeder versuchte, der »Shogun« des Reiches zu werden.

Zu diesem Zweck gilt es nun zu handeln, diplomatische Kontakte zu knüpfen und, wenn gar nichts mehr hilft, ein paar Armeen auszusenden. Alles in allem ein recht komplexes, attraktives Spiel. Die Demoversion ist leider arg

karg und vermag nur sehr oberflächlich einen Eindruck des fertigen Spiels zu vermitteln.

Die letzte Demo, die wir uns genauer ansehen möchten, ist »Ghostbusters II«. Hier erwartet uns wieder ein komplettes, spielbares Level. In New Yorks Untergrund sind die Geister los. Es spukt an allen Ecken und Enden der Kanalisation, und der Spieler muß, in Person eines der Geisterjäger, all die schemenhaften Wesen abwehren, die sich hier herumtreiben. In der Demo stellt sich die Kanalisation als ein langer senkrechter Schacht dar, in den der Geisterjäger abgeseilt wird. Hier lauert an den Wänden die schleimige Geisterbrut darauf, den Geisterjäger so gründlich zu erschrecken, daß er auf ein weiteres Erkunden der Kanäle keine Lust mehr hat. Das Demokonzept kann unumwunden als eine Bereicherung der Public-Domain-Szene gewertet werden. Insbesondere, da es sowohl für Hersteller als auch für Anwender von Vorteil ist. Der Anwender kann probespielen, was im Handel vielfach unmöglich ist, während der Hersteller auf diesem Wege viele Anwender über neue Software informieren kann, ohne eine Flut von Raubkopien zu provozieren. (hs)

Alle hier besprochenen Demos können beim PD-Service Lage, Sonnenborstel 31, in 3071 Steimbke bestellt werden. Grundsätzlich können die Demos nach Abstimmung mit den Herstellern in jede reine PD-Reihe übernommen werden.



»Lords of the Rising Sun«, etwas zu oberflächlich



Ein ganzes spielbares Level – »Ghostbusters II«

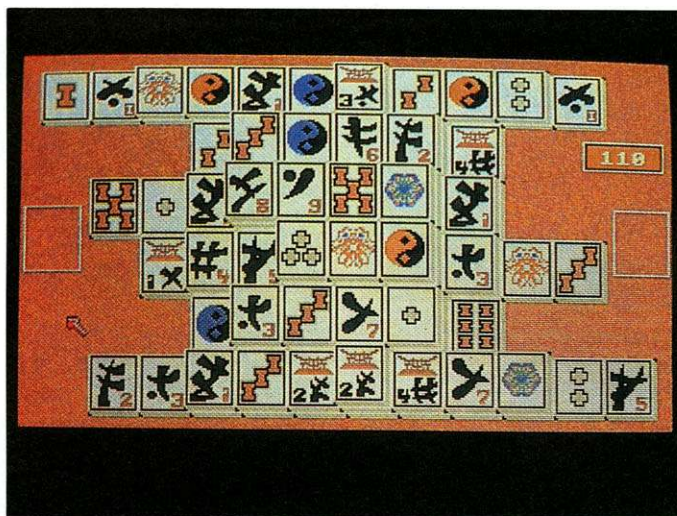
Nein, wahrscheinlich nicht. Aber daß Sie ein Spiel von solchen Qualitäten im kommerziellen Sektor finden, ist genauso unsicher. Trotzdem ist Public Domain für Überraschungen gut, wie man an der vorliegenden Antares Disk # 42 sieht. Hier glänzen viele Spiele durch gute Ideen, gute Aufmachung und attraktiven Preis.

China Challenge

144 Steine, von denen jeder mit einem Symbol gekennzeichnet ist, werden zu einem pyramidenartigen Haufen gestapelt. Die Steine liegen übereinander, so daß der Spieler nur jeweils den obersten Stein aus jedem Stapel sehen kann. Die Symbole, die die Steine tragen, existieren mehrfach, so daß es für jeden Stein drei weitere gleichartige gibt. Die Aufgabe des Spielers ist es, Steinpärchen aus der Pyramide zu entfernen und die Stapel so bis auf den letzten Stein abzutragen. Doch dürfen nur solche Steine entnommen werden, die sich ohne einen anderen Stein zu bewegen, nach links oder rechts aus dem Stapel ziehen lassen. »China Challenge« basiert lose auf dem japanischen Brettspiel Mah Jongg, dem die Spielsteine entliehen wurden. »China Challenge« ist ordentlich programmiert, schnell, grafisch gut gestaltet und sehr bedienerfreundlich.

Accordion

Patience, das Auflegen von Spielkarten nach besonderen Regeln, sind eine weit verbreitete Freizeitbeschäftigung, die auch schon oft zu Computerspielen umgearbeitet wurde. Mit »Accordion« liegt ein gut gemachtes Patience-Spiel für den Amiga vor. Der Spieler muß Karten einzeln aus einem Stapel heraus nehmen und aufgedeckt ablegen. Die auf-



»China Challenge« – ein kniffliger Knobelspaß

Public Domain-Spiele-Show

Antares # 42 – Unterhaltung Deluxe

Viel Kurzweil und Spaß, vielleicht ein wenig Knochelei und Kopfarbeit, gerne viel Action und Abenteuer; sähe so das ideale Spiel aus und kann der Anwender so etwas in der Public Domain finden?

tete Freizeitbeschäftigung, die auch schon oft zu Computerspielen umgearbeitet wurde. Mit »Accordion« liegt ein gut gemachtes Patience-Spiel für den Amiga vor. Der Spieler muß Karten einzeln aus einem Stapel heraus nehmen und aufgedeckt ablegen. Die auf-

gedeckten Karten müssen nun nach Farbe oder Art übereinander gelegt werden. Es gilt, alle Karten auf einen Stapel zu vereinigen. Die vorliegende Version ist leider nur eine Ansichtsversion, die zwar komplett spielbar ist, sich aber auf zwei ver-

schiedene Eröffnungsbilder beschränkt, so daß dem Variationsreichtum des Spiels von vornherein Grenzen gesetzt sind. Trotzdem ist dieses Spiel eine gute Gelegenheit, sich einmal mit Patienten zu beschäftigen, zumal auf der Disk eine gute, leider in Englisch gehaltene Spielbeschreibung enthalten ist.

Boulder Crash

Die Hatz nach Diamanten scheint endlos. Das Kommerzprodukt, das hier Pate stand, kann schon mit Fug und Recht als Klassiker bezeichnet werden.

Der Spieler muß hier einen kleinen Charakter durch diverse Höhlen (Levels) steuern und dabei verstreut umherliegende Diamanten einsammeln. Erst wenn er ein bestimmtes Kontingent Edelsteine besitzt, öffnet sich ein Tor zur nächsten Höhle. Um das Ganze noch ein wenig gefährlicher zu gestalten, liegen in jeder Höhle Felsbrocken herum, die bei der kleinsten Berührung einstürzen. Obendrein wimmelt es in einigen Gewölbchen auch noch von diversen Ungeheuern, die der Spieler nicht berühren darf, da er sonst eines seiner drei Leben verliert. (hs)

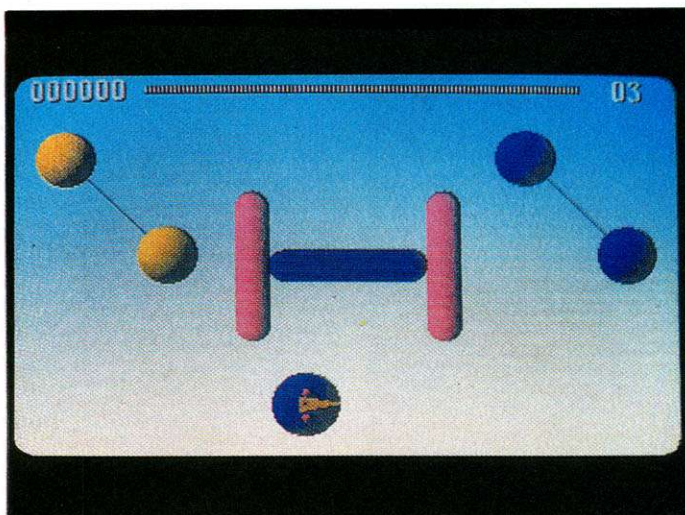
Disk: Antares 42
Enthaltene Titel:
Accordion
Boulder Crash
China Challenge
Imperium Romanum (Strategie)
Revcomp (Reversi)
Slot Cars (siehe PD AMIGA DOS - 8/90)



»Accordion« – Patienten zum Selberlegen



»Boulder Crash« – ein spannender Höhlentrip



Gut gezielt ist halb getroffen...



Wer schießt hier aus dem Hinterhalt?

E-Motion

Kernspaltungen, wirbelnde Elektronen, molekulare Zusammenstöße, explodierende Atome und mitten-drin ich selbst in einer kleinen Kapsel – Einsteins Universum aufzuräumen ist ganz schön anstrengend.

Das Spielprinzip ist einfach, aber effektiv: Man nehme bunte Kugeln und lasse sie miteinander kollidieren. Um die Atombällchen in Bewegung zu setzen, benutzt man ein Miniraumschiff wie ein Billiardqueue, mit dem die Kugeln gegeneinandergeschoben werden. Passieren derartige "Unfälle" mit zwei gleichfarbigen Kugeln, so verschwinden sie. Sobald der Bildschirm leergeräumt ist, kann das nächste Level beginnen. Stoßen allerdings verschiedenfarbige Kugeln aneinander, explodieren sie nicht, sondern produzieren eine zusätzliche, noch sehr kleine Kugel in einer dritten Farbe.

Verspeist man das Nachwuchsatom, erhält man neue Energie. Wartet man zu lange mit der Zerstörung der immer schneller pulsierenden ausgewachsenen Atome, geht durch die dann bald erfolgende Explosion die Energie gen Null, und eins der drei Leben ist schnell dahin.

Erschwert wird das galaktische Billiardspiel durch Röhren und Mauern sowie elastische Bänder, die mehrere Kugeln inklusive des Raumschiffs miteinander verbinden, wodurch sich alle Bewegungen radikal ändern.

»E(instein)-Motion« ist ein Strategiespiel mit einer gehörigen Portion Glück, das einen lange an den Bildschirm fesseln kann. Noch mehr Spaß macht es, wenn man zu zweit auf Atomjagd geht.

(Antje Hink/hs)

AMIGA DOS Blitzlicht

Name: E-Motion

Hersteller: U.S.Gold

Quelle: Fachhandel

Preis: 84,95 DM

Positiv:

- originelles Spielprinzip
- auf Harddisk installierbar
- Zwei-Spieler-Option

Negativ:

- zu hoher Preis



Crack Down

Zwei Freunde müssen die Stadt, das Land, die Welt vor dem bösen Dr. K. retten. Die genetischen Züchtungen des irren Wissenschaftlers sorgen dabei für jede Menge Stress!

Hauptaufgabe des Spielers ist es, in den 16 Levels der Festung alle Zeitbomben an ihre vorgezeichneten Plätze zu legen und die Flucht ins nächste Level anzutreten. Schafft man es nicht, sich mit MG, Bazooka oder Superbombe den Weg innerhalb des Zeitlimits von drei Minuten freizuschießen, ist eines der fünf Leben unwiederbringlich dahin.

Frische Munition liegt in ausreichender Menge in den grafisch sehr abwechslungsreich gestalteten Levels herum. Auch ein Extraleben ist hin und wieder zu ergattern. Sind fünf Leben vertan, fängt zwar die (ohnehin nicht abspeicherbare) Punktzahl wieder bei Null an, das Spiel jedoch geht ohne Unterbrechung weiter.

Flotte Musik untermalt das ebenso flotte Ballerspielchen, an dessen Grafik (das Szenario wird in der Vogelperspektive dargestellt) nicht allzuviel auszusetzen ist. Nur der Übersichtsplan verwirrt mehr als er nützt, doch zum Glück gibt's ja eine Pausentaste...

Bei dieser Automatenumsetzung ist es gelungen, die Schwachstellen des Originals auszubügeln, indem sich die Zwei-Spieler-Option tatsächlich als solche entpuppt und sich die Helden nicht länger

gegenseitig behindern, sondern wirksam ergänzen können. Echtes Teamwork ist möglich, was den Reiz der ansonsten eher durchschnittlichen Spielidee gewaltig erhöht.

(Antje Hink/hs)

AMIGA DOS Blitzlicht

Name: Crack Down

Hersteller: U.S.Gold

Quelle: Fachhandel

Preis: 84,95 DM

Positiv:

- Zwei-Spieler-Option

Negativ:

- abgeschmackte Story
- spielerisch zu wenig Abwechslung





Sierra-Abenteuer aus einer düsteren Zukunft!

Manhunter San Francisco

Neuer Schauplatz, altes Prinzip: Der Kampf gegen die Orbs geht weiter. Dieses Mal darf der Spieler seine Fähigkeiten als Menschenjäger in San Francisco unter Beweis stellen.

Das Ende von »Manhunter New York« ließ schon ahnen, daß der Kampf gegen die Orb-Invasoren und ihren Komplizen Phil noch längst nicht vorbei ist. Sierra hat es endlich geschafft, die Fortsetzung auch auf den Amiga zu portieren. Der Schauplatz ist nun San Francisco, am Spiel selbst hat sich wenig geändert: Unser Manhunter muß (oder besser sollte) Feinde der Orbs aufspüren und ihrer Strafe zuführen, statt dessen schlägt er sich auf die Seite der lokalen Rebellen und kommt einem Komplott der Orbs auf die Spur, den er jedoch rechtzeitig vereiteln kann.

Mit der Logik sollte man es nicht allzu genau nehmen, oftmals tut man etwas, weil es die einzige noch offene Möglichkeit ist – erst später wird dann klar, was eigentlich los war. Das Ableben des Helden wird vom obligatorischen Spritzen von Blut und Eingeweiden begleitet, die Grafiken und Animationen sind stellenweise noch geschmackloser als im ersten Teil. Wer an der ersten Folge bereits Gefallen gefunden hat, der wird auch vor der Fortsetzung nicht sonderlich zurückschrecken. Wer sich dagegen mit der Welt

der Manhunter bekannt machen will, der sollte vielleicht lieber auf den ersten Teil zurückgreifen. Eine weitere Fortsetzung ist durchaus zu befürchten, sie wird wahrscheinlich in London spielen...

(Michael Anton/hs)

AMIGA DOS Blitzlicht

Name: Manhunter San Francisco

Hersteller: Sierra
Vertrieb: Fachhandel
Preis: 119,- DM

Positiv:
– keine Texteingaben nötig

Negativ:
– etwas unlogisch
– recht geschmacklos



Fade Suche nach einer Templerstatue

Astate

Gesucht werden die fünf Teile einer Statue, die mit einer Legende der Tempelritter in Verbindung steht. Der Suchende braucht nicht nur Intelligenz, sondern auch ein möglichst dickes Fell...

Der Spieler steuert die Figur eines Archäologen, der die Landschaft nach Spuren durchsucht, die zu den Teilen der Statue führen. Verschiedene Hilfsmittel sind in der Ausrüstung enthalten: Ein Scanner, mit dem verborgene Gegenstände lokalisiert werden, und Hacke und Bohrer, mit denen diese Gegenstände ausgegraben werden können. Gefundene Objekte können aufgenommen werden und entweder an der passenden Stelle zur Lösung von Rätseln auf dem Weg zu Teilen der Statue eingesetzt werden oder als unnütz in einem Mülleimer verschwinden. Also läuft man durch die Gegend und sucht und sucht und...

Die Landschaften sind in stimmungsvollen Grafiken recht gut gestaltet, der Sound ist dagegen eine Katastrophe. Digitalisierte Sprache kündigt Anfang und Ende des Spiels an, der Rest ist Musik, aber was für eine. Zwar wird jedes Bild mit einer anderen Melodie unterlegt, diese dürften jedoch auch moderne Komponisten zu Schreikrämpfen veranlassen. Einzig die Mausbedienung ist recht gut gelungen – sieht man einmal davon ab, daß man zum Auffinden von Objekten fast pixelgenau treffen muß. Das

sind selbst bei einem Low-Res-Bild noch jede Menge Punkte, so daß Astate eher zu einem Geduldsspiel wird. Und der Spielwitz? Anfangs mag es ja noch reizvoll sein, suchend umherzuirren, mit der Zeit wird das Spiel jedoch so platt wie die Füße, die sich der Archäologe läuft...

(Michael Anton/hs)

AMIGA DOS Blitzlicht

Name: Astate
Hersteller: New Deal
Vertrieb: Fachhandel
Preis: 59,95 DM

Positiv:
– passable Grafiken
– interessante Bedienung

Negativ:
– nervtötende Musik
– extrem langatmig





Ballern mit realen Kriegsgerät

Heavy Metal

Freunde schwerer Geschütze dürfen den Feuerfinger schon mal massieren, hemmungslose Ballerei mit schwerem Kriegsgerät ist angesagt. Skrupel und Moral sollten besser im Schrank belassen werden.

Dem Spieler stehen drei "feurige" Fahrzeuge zur Verfügung, die allesamt reale Kampfmittel simulieren (sollen). Die Aufgabe ist wie so oft, möglichst viel zu treffen, ohne selbst getroffen zu werden. Im ersten Fahrzeug, einem Kampfpanzer, rast man durch die Gegend und erledigt alles, was man ins Visier bekommt. Dies ist noch der anspruchsvollste Teil, da die Bedienung recht komplex ist und dafür eigentlich drei Hände nötig wären. Das zweite Fahrzeug ist eine Art Strandbuggy mit Granatwerfer, der in Fluchtpunktperspektive durch feindliche Stellungen rast und dabei Hindernisse abschießen oder ihnen ausweichen muß. Defensiv ist das letzte Fahrzeug, mit ihm muß man einfliegende Flugzeugformationen und gelegentlich auftauchende Panzer abwehren.

Hat man sich in Trainingsrunden auf allen Fahrzeugen qualifiziert, kann man sich auch als Taktiker betätigen, indem man seine Einheiten denen des Feindes entgegenstellt.

Die Thematik ist sicher keine Sensation, die Gestaltung nur bedingt. Die Grafik ist solider Durchschnitt, nur in der Periskopsicht des Panzers wurde nette Detailarbeit geleistet. Musik und Effekte sind situa-

tionsbezogen mager, die Bedienung manchmal etwas unhandlich. Alles in allem ein Spiel, mit dem höchstens hartgesottene Militaria-Fans glücklich werden dürften.

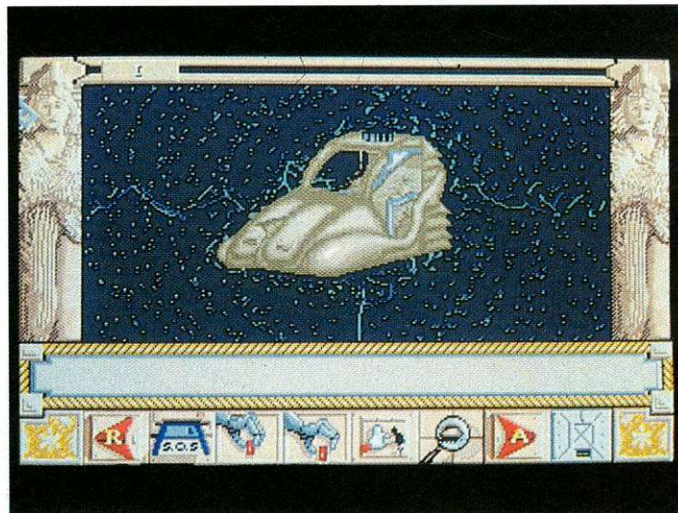
(Michael Anton/hs)

AMIGA DOS Blitzlicht

Name: Heavy Metal
Hersteller: Access
Vertrieb: Fachhandel
Preis: 84,95 DM

Positiv:
- brauchbare Grafik
- Highscores speicherbar

Negativ:
- wenig Abwechslung
- viel zu teuer



Ein besonderes Erlebnis: Ausflüge mit einer defekten Zeitmaschine

Chronoquest II

Wenn einer eine Reise tut, so kann er viel erzählen. Das trifft ganz besonders auf Reisen mit einer defekten Zeitmaschine zu.

Da der Generator der Zeitmaschine Explora bei einem Kurzschluß den gesamten Treibstoff verbraucht hat, muß man sich logischerweise neuen besorgen, will man ins heimatliche Schloß und in das Jahr 1922 zurückkehren. Metalle aller Art sind also zu suchen und in den Generator zu befördern, wobei es nicht unbedingt auf die Masse an Stoff, sondern eher auf den Gehalt an Radioaktivität ankommt. Je nach Schubkraft landet man in einer von 13 Zeitzonen, um dort weiteren Brennstoff aufzutreiben – oder aber in einem aktiven Vulkan, was jeden Kommentar ziemlich überflüssig macht.

Die zu lösenden Rätsel sind in allen Schwierigkeitsgraden vorhanden. Mal stolpert man geradezu über irgendwelche nützlichen Gegenstände, mal muß man sturen Musketieren erst den Fehdehandschuh vor die Füße schleudern, bevor man zu ihrem Chef und damit weiter kommt.

Die Steuerung dieses Grafik-Adventures wird durch 15 Icons vorgenommen. Zehn davon stehen als direkte Actionbefehle zur Verfügung, die restlichen fünf übernehmen Sichern, Laden sowie An- und Abschalten der recht guten Sprachausgabe bei Dialogen.

Kämen dazu nicht auch noch der magere Sound, die langen Ladezeiten und die äußerst umständliche Restore-Routine, so wäre das Programm durchaus im gehobenen Bereich anzusiedeln, in den es rätselmäßig absolut hineingehört. So aber ist es leider nur Mittelklasse und allenfalls Besitzern einer Harddisk zu empfehlen.

(A. Hink/hs)

AMIGA DOS Blitzlicht

Name: ChronoQuest II
Hersteller: Psygnosis
Quelle: Fachhandel
Preis: voraussichtlich 99 DM

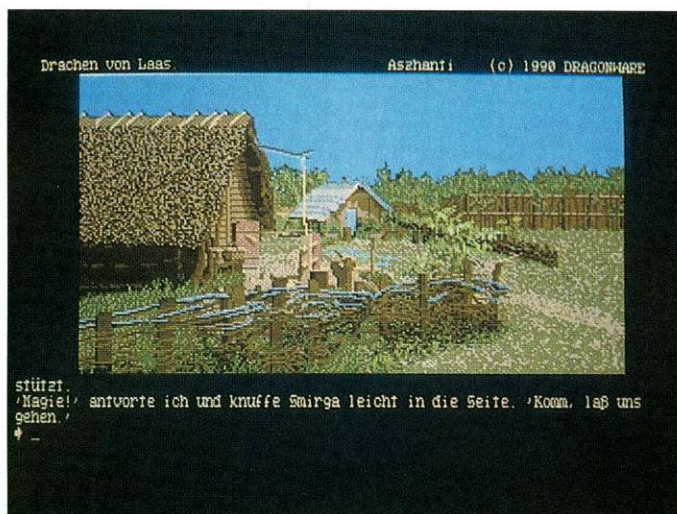
Positiv:
- auf Harddisk installierbar

Negativ:
- umständlich zu bedienen
- schwacher Text



Was hat ein jugendlicher in einem verschlafenen Dorf in einem Fantasyland schon für Perspektiven... Nun, zum einen kann man ein solides Handwerk erlernen und einen Beruf ausüben – und dabei den Rest seines Lebens in der Provinz versauern. Man kann jedoch auch seinen Rucksack packen, in die weite Welt ziehen und versuchen, ein berühmter Abenteurer und Held zu werden. Für die letztere Möglichkeit entscheiden sich auch Smirgal und Aszhanti aus dem verschlafenen Dörfchen Hyllok im Lande Laas. Die beiden sind ein ideales Team, denn während Smirgal die Probleme in bester Conan-Manier angeht, verwendet der Zauberlehrling Aszhanti lieber Magie. Doch Abenteurer brauchen Verpflegung, Kämpfer Waffen und Zauberer Magie, so daß vor dem Start in die weite Welt erstmal etwas Ausrüstung aufgetrieben werden muß. Nach den ersten kleineren Hindernissen kann dann der Ausflug in die weite Welt beginnen. Die ist jedoch nicht so romantisch, wie es auf den ersten Blick scheint. Trolle und Killerbienen sind nur ein Aspekt, der das Leben etwas schwer macht. Immerhin muß man auch Heldentaten vollbringen und für das eigene Wohl sorgen. Einem Bauern bei der Ernte zu helfen, ist vielleicht nicht gerade das, was man sich unter Heldentaten vorstellt, aber immerhin kann man sich dadurch ein Nachtquartier erarbeiten. Ist man erst einmal weiter von Hyllok weg, kommt man schnell mit Leuten ins Gespräch und erhält von ihnen Aufträge, deren Erfüllung Reichtum und Ruhm bringt.

»Die Drachen von Laas« ist ein klassisches Textadventure – und noch mehr. »Klassisch« ist dabei überwiegend als »in der Machart von Magnetic Scrolls« zu verstehen, denn an die Adventures dieser Firma ist es stark angelehnt: Der überwiegende Teil der Handlung wird in Textausgaben vorgestellt, zu einigen Szenen sind jedoch auch illustrierende Grafiken vorgesehen. Die Steuerung der Helden erfolgt vollständig mit Tastatureingaben, die Maus kommt nur selten zum Einsatz. Doch nun zum Mehr. Da wäre zunächst einmal der doppelte Held. Zwar sind beide Figuren sozusagen »unzertrennlich«, man kann jedoch zwischen ihnen



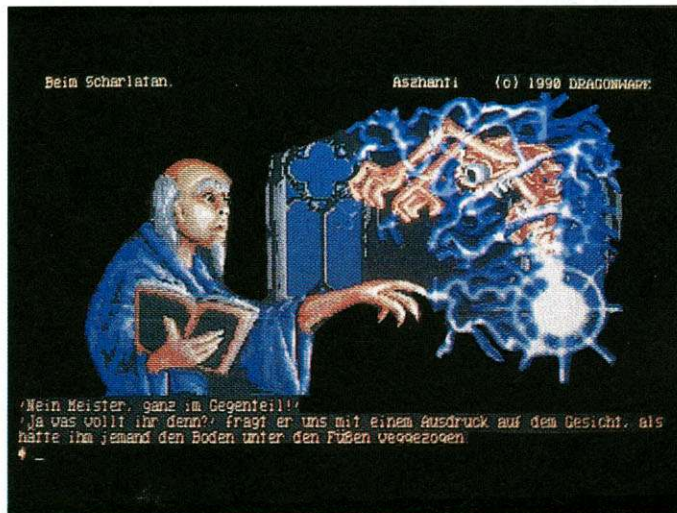
Auf Abenteuersuche im Lande Laas

Die Drachen von Laas

Gute Textadventures aus deutschen Landen sind selten. Dies liegt sicherlich zu einem großen Teil an der komplizierten Grammatik der deutschen Sprache. Mit den »Drachen von Laas« stellt sich ein neues Spiel des Genres vor, das obendrein noch komplett in Deutsch gehalten ist.

hin- und herschalten, was der eine mit Gewalt nicht lösen kann, schafft der andere vielleicht mit Köpfchen. Auch sind einige Elemente des Rollenspiels eingeflossen, so daß die Charaktere mit wachsender Erfahrung auch schwierigere Aufgaben lösen oder mehr Prügel von stärkeren Gegnern verkraften können. Für Abwechslung ist also gesorgt.

Ein generelles Problem bei Textadventures ist die Verarbeitung der Eingaben. Die Steuerung der Helden bedarf einiger Tipparbeit, da ihnen jede Aktion im Klartext mitgeteilt werden muß. Kommt es zu Mißverständnissen zwischen Spieler und Computer, müssen die Eingaben variiert werden, was oft nervig sein kann. Besondere Sorgfalt muß daher auf die Eingabe der



Der Spieler gelangt in eine Welt voller Magie

Kommandos und ihre Auswertung gelegt werden. Mit einem Vokabular von rund 2000 Wörtern ist der Parser recht leistungsfähig, er versteht auch komplexere deutsche Sätze. Leider muß trotz allem öfters noch nach dem richtigen Wort gesucht werden, da an manchen Stellen keine Synonyme vorgesehen sind. (Einkäufe werden in Laas durch einfaches Zahlen getätigt, also »Zahle Waffen« statt »Kaufe Waffen« – das ist ein Hinweis!!!) Die Eingabe ist ebenfalls recht komfortabel, so können häufig benötigte Wörter abgekürzt oder Befehle auf die Funktionstasten gelegt werden. Leider wurde dabei jedoch ein ganz elementarer Punkt vergessen: Die letzte Eingabe kann zwar wiederholt, aber leider nicht richtig editiert werden...

Alles in allem ist »Die Drachen von Laas« ein Spiel, welches allen Rätselfreunden empfohlen werden kann. Zum einen jenen, die den Hauch von Magnetic Scrolls oder Infocom einmal auf deutsch erleben möchten, zum anderen insbesondere auch jenen, die durch die überwiegend englischen Vertreter dieses Genres bislang vor ebendiesem zurückschreckten.

(Michael Anton/hs)

AMIGA DOS Blitzlicht

Name: Die Drachen von Laas
Hersteller: Dragonware
Vertrieb: Fachhandel
Preis: stand bei Redaktionsschluß noch nicht fest

Positiv:

- brauchbarer Parser
- gute Zwischengrafiken
- freundlicher Kopierschutz
- Festplatteninstallation möglich

Negativ:

- kleine Schwächen in der Bedienung
- Thema nicht sonderlich neu



AMIGA DOS -Spieletips

Dungeon-Master-Fans aufgepaßt: Nachdem wir die Levels eins bis vier dieses spannenden Rollenspiels in der letzten Ausgabe als Karten vorgelegt haben, folgen nun auf diesen Seiten die Levels fünf bis acht. Die Symbole auf den Karten bedeuten folgendes:

- X - Geheimtür
- + - wertvoller Gegenstand
- K - Schlüssel
- - Bodenschalter
- ⚡ - Kraftfeld
- | - Tür
- Y - Goldmünze einwerfen
- ≡ - Treppen

Space Quest III - der zweite Teil

Die Steuerung des Schiffs ist relativ einfach, denn sie erfolgt über einen Computer (look computer), der die meisten Aufgaben übernimmt. Etwas Starthilfe braucht er allerdings: Mit eingeschaltetem Radar und aktiviertem Frontshield ist der Durchbruch in die Freiheit, der durch einen Schuß aus der Bordwaffe eingeleitet wird, kein Problem mehr.

Auf zu neuen Abenteuern

Wie Sie sicherlich feststellen werden, ist es gar nicht so leicht, auf Ortega zu überleben. Da sind wohl erst einige Ausrüstungsgegenstände vonnöten. Da sieben Bucks (Weltraumwährung) nicht die Welt sind und Roger sich im Monolith Burger entsprechend wenig leisten kann, ist Phleebhut unser nächstes Ziel. Um Arnold, den Annihilator, der sich das Terminieren zur Aufgabe gemacht hat, brauchen wir uns vorläufig noch nicht zu kümmern. Westnordwestlich von unserem Landeplatz befindet sich das World o' Wonder, ein kleiner Shop, in dem wir das Glowing Gem gegen Bares und etliche nützliche Gegenstände (orat, hat, underwear) eintauschen können. Der Terminator stellt uns beim Verlassen des Geschäftes. Jetzt wird auch klar, was er von uns will: Das Geld für die bestellte und erhaltene, aber nicht bezahlte Pfeife aus dem zweiten Teil soll er eintreiben.

Seitdem die Abenteuerspiele gelernt haben, ihre Inhalte nicht mehr nur über englische Texte auszugeben, werden sie auch hier in Deutschland immer beliebter. Sierras diverse Quest-Serien blicken auf eine große begeisterte Fan-Gemeinde. Kein Wunder also, daß wir uns auch im Rahmen unserer Helpline diesen Spielen intensiver widmen.

Mangels Barschaft räumt er uns eine kleine Chance zur Flucht ein, die zur Terminierung des Terminators genutzt werden sollte. Für diese Aufgabe gibt es prinzipiell zwei Möglichkeiten: Die leichtere ist es, ihn in den Aussichtsturm vom Mog's Head zu locken und ihn mit den dortigen Hilfsmitteln fertigzumachen (use hook).

Einige Punkte mehr kann man allerdings kassieren, wenn man ihn zur Höhle mit den

Pods lockt und diese die Arbeit machen läßt, was aber nicht immer klappt. Wie dem auch sei, Arnolds Gürtel ist für unsere Mission von großem Wert. Beim zweiten Weg kann der Orat Stick von großem Nutzen sein.

Fantastisches Fast Food

Da Arbeit hungrig macht, steht nun ein Besuch im Mo-

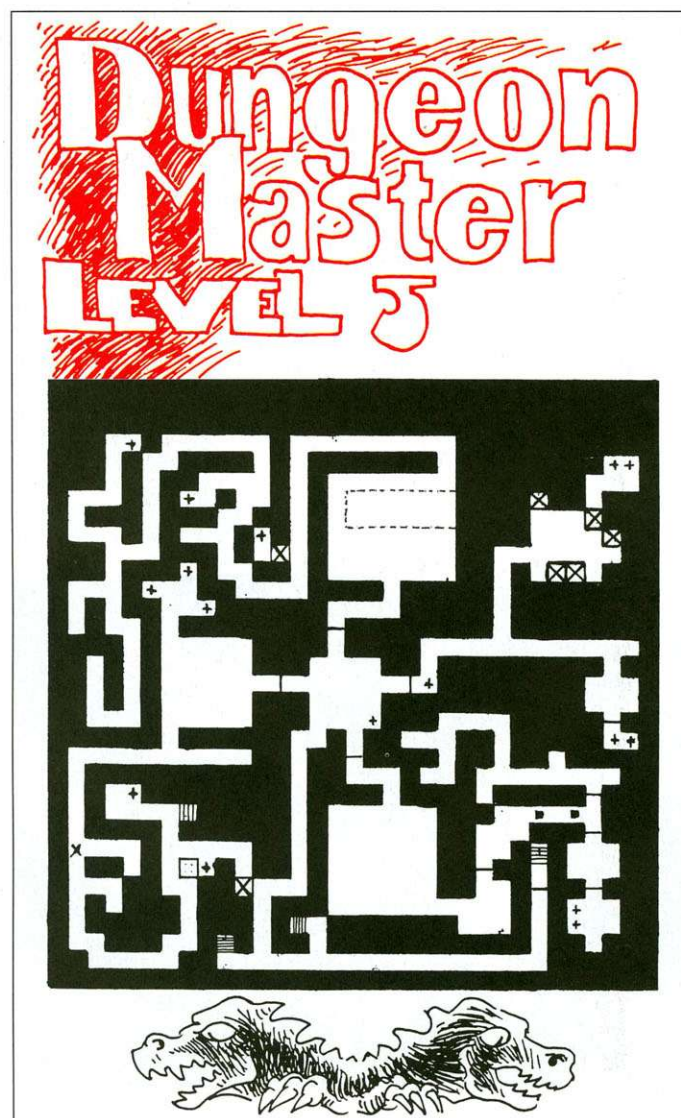
nolith Burger an. Trotz der vielen Sterne im Hintergrund sind die Speisen nur sehr schwer genießbar. Das Ausprobieren bereitet aber trotzdem Spaß. Auf jeden Fall sollte ein Monolith Fun Meal geordert werden, welches einen Decoder-Ring beherbergt. Spielt man jetzt eine erfolgreiche Runde am Videoautomaten, dazu müssen fünf Extraleben eingeflogen werden, erscheint eine codierte Nachricht auf dem Schirm, die mit Hilfe des Rings und sehr guter Augen entziffert werden kann. Um Ihnen die Leseanstrengung zu ersparen, hier der genaue Wortlaut: "Help us. We are being held captive by ScumSoft on the small moon of Pestulon. An impenetrable force field surrounds the moon. It must first be deactivated. Its origin is unknown to us. ScumSoft security is armed with jello guns. We're counting on you whoever you are. Two guys in trouble."

Ortega, ich komme...

Dank der feuerfesten Wäsche von Phleebhut können wir uns frei bewegen. Der Weg wird schnell offensichtlich, die beiden zwielichtigen Typen, die darauf liegen, entfernen sich nach einiger Zeit von selbst. Durch das zurückgelassene Teleskop erhaschen wir einen ersten Blick auf Pestulon, sowie den felderzeugenden Generator. Der Generator liegt auf dem weiteren Weg und kann mittels der auf dem Weg liegenden Sprengladung zerstört werden. Zu diesem Zweck muß die entsprechende Kuppel erklommen und der Detonator an der richtigen Stelle fallen gelassen werden. Der Erfolg ist beachtlich, unser Rückweg wird aber leider auch blockiert. Vielleicht kann man mit der Stange? Jawohl, ohne Probleme, solange man keinen weiteren Detonator im Gepäck hat. Wie nun auch im Schiff auf dem Scanner ersichtlich, ist der Weg nach Pestulon jetzt frei, also auf in die letzte Runde.

Pestulon, Mond der Piraten

Nach der Landung führen alle Wege zur Basis von ScumSoft,



aber nur einer an den Wachen vorbei: der Invisibility Belt erlaubt ein sicheres Betreten der Basis, aber nur auf dem direkten Weg. Mit einem beherzten Knopfdruck gelangen wir in das Zentrum des bösen Imperiums. Nach einigen Runden im Laufrad des Korridors wird uns klar, daß wir so nicht weiterkommen. Probieren wir doch einmal die Türen. Die besonders gesicherten Türen sind nicht zu öffnen, hinter den anderen erwarten uns Überraschungen. Unangenehme auf der Außenseite des Kreises, angenehme auf der Innenseite.

Dort findet sich auch ein Umkleideraum, der nach einigem Suchen eine passable Verkleidung an den Tag bringt (search room, wear coveralls). Als Saubermann getarnt, steht unserem Erfolg nichts mehr im Weg. Im hiesigen Großraumbüro darf man nicht auffallen und vor allem keinen vollen Papierkorb übersehen, solange wir diese leeren (use vaporizer), geht alles klar. Bei der Säuberungsaktion fällt besonders ein Portrait des Firmengründers auf, das am nächsten Farbkopierer vervielfältigt werden kann. Im hinteren Teil des Labyrinths, dem Büro des Chefs, finden wir seine Keycard, die besonders wertvoll für uns ist.

Zwei nette Kerlchen

Durch den Besitz der Keycard und der Fotokopie eröffnen sich uns völlig neue Wege. Die gesicherten Türen lassen sich nämlich jetzt öffnen: use keycard, hold photo. Dahinter befinden sich die beiden Gefangenen, die per Knopfdruck und Einsatz des Vaporizers befreit werden können. Unser Tun ist aber leider entdeckt worden, und es kommt zum Schlussskampf, der per Roboter ausgetragen wird. Ist er besiegt, steht unserer Flucht außer ein paar Wachen nichts mehr im Wege. Fünf von ihnen müssen erledigt werden, der Rest gibt dann auf. Von jetzt an läuft die Schlußsequenz, und das Warten auf »Space Quest IV« beginnt.

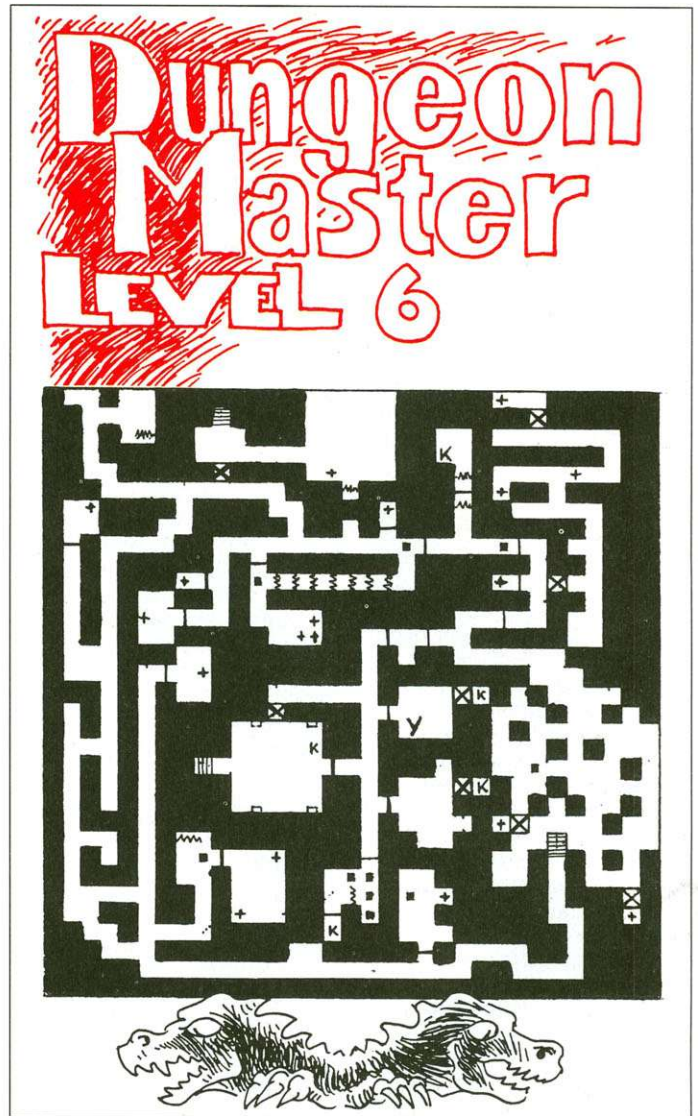
Wie versprochen machen wir in dieser Ausgabe mit der Lösung zu Infocom's »Journey« weiter. Dieses Mal nähert sich die Reise ihrem Ende, und die Helden müssen sich auf eine Konfrontation mit dem Dread Lord vorbereiten. Denn dieser fiese Zeitgenosse besitzt den letzten der gesuchten Steine.

Wir verließen den Zauberer Astrix und fanden uns auf einer Ebene wieder, die nach Zan führte. Als wir einen Schatten bemerkten, der von einem großen, schwarzen Vogel stammte, versteckten wir uns. Praxis war der Meinung, daß wir die Reise nur im Nebel fortsetzen könnten, damit uns der Vogel nicht erspähen kann. Er mischte also Luft und Wasseressenz zusammen, und wir konnten tatsächlich ungehindert weiterziehen. Als ein Unwetter aufzog, versteckten wir uns in einer Höhle, um Schutz zu suchen.

Am Ziel der Reise

In dieser Höhle trafen wir auf Praxis' alten Freund Umber. Der warnte uns vor den "mudwags", die aber nur am Tage auftauchen würden. Da wir aber nicht bis zum Abend warten konnten, mußten wir uns etwas anderes einfallen lassen. Als Umber eingeschlafen war, überwandten wir uns und durchsuchten ihn. Praxis fand in Umbers Sack einen schwarzen Stein. Nachdem er ihn zerbröselte und mit Feueressenz gemischt hatte, wandte er die Mischung auf seinen Stab an. Fortan umgab uns Dunkelheit, und der weitere Weg stand uns offen.

Als wir gegen Abend Zan erreichten, führte uns unser erster Weg in den Curio Shop. Hurth sah sich zunächst einmal im Laden um, und als Praxis vom Besitzer angesprochen wurde, verwickelte er diesen in ein Gespräch über seltene Steine. Der Besitzer zeigte Praxis voll Stolz einen Stein, den ich auf Praxis' Bitten hin gegen das Fernglas eintauschte. Praxis zerbröselte auch diesen Stein und legte die erhaltene Essenz zu seinen anderen Utensilien. Da wir von der anstrengenden Reise müde geworden waren, suchten wir die Taverne auf. Hurth fielen sofort Männer auf, die an einem anderen Tisch saßen und die uns offensichtlich erkannten. Praxis machte mich mit unserem letzten roten Gestein unsichtbar, und so konnte ich mich zum Nachbarisch hinschleichen und die Männer belauschen. Ich erfuhr, daß die Männer uns erwartet hatten und uns eine Falle stellen wollten. Aus diesem Grunde hatten sie mit allen Kapitänen außer einem, der betrunken war, gesprochen. Den Namen



des Käptens merkte ich mir natürlich. Nach einem ausgiebigen Essen verließen wir die Taverne und gingen ins örtliche Inn, um uns ein Quartier für die Nacht zu besorgen. Wir trugen uns ordnungsgemäß ein, akzeptierten den Vorschlag des Wirtes und verließen das Inn wieder.

Es wurde Zeit, sich um ein Boot für die Überfahrt zur Mysty Isle zu kümmern. Der Hafenmeister fragte uns am Kai, ob wir denn an einen bestimmten Kapitän denken würden. Wir nannten ihm natürlich den Namen des betrunkenen Skippers. Auf diese Weise erfuhren wir, daß sein Schiff die Zephir war. Nachdem wir uns alle anderen Schiffe angesehen hatten, entschieden wir uns natürlich für die Zephir.

Nachdem wir den Kapitän über unser Reiseziel aufgeklärt hatten, teilte dieser uns mit, daß ein Sturm aufzöge

und der Abreisetermin der nächste Morgen sei. Da es Zeit zum Schlafen war, gingen wir wieder in die Stadt.

Mißtrauisch und vorsichtig, wie wir nun einmal waren, campierten wir draußen, statt ins Wirtshaus zu gehen. Wie richtig diese Vorsicht war, erfuhren wir am nächsten Morgen, als uns der Sheriff auf dem Weg zum Kai aufhielt und uns wegen Brandstiftung im Wirtshaus verhaftete.

Da ein Ausbruch unmöglich war, mußten wir auf eine List sinnen. Praxis mischte seine graue mit Feueressenz und ließ darauf die Erde beben, um den Sheriff anzulocken. Als er kam, wendete Praxis die Reagenz auf den Sheriff an, der daraufhin in Trance verfiel und uns freiließ. Der Weg zur Zephir war frei. Wir sahen uns auf dem Schiff um und konnten schon bald die Zwillinginseln Estril und Lethor ausmachen. Auf der

Fahrt entspannten wir uns, wurden aber von dem Sturm überrascht. Ein Mast brach, und der Käpten rettete Bergs Leben, ging dabei aber über Bord.

Als der Sturm abflaute, fanden wir uns im Dunstkreis der Mysty Isle wieder. Um uns wenigstens einigermaßen orientieren zu können, kletterte ich auf den Mast. Da ich nicht hoch genug kam, hob Praxis mich mit dem Elevation Spell an, und so konnte ich die Richtung, in der die Insel lag, ausmachen. Praxis zauberte Wind, und wir näherten uns der Insel.

Auf zum großen Finale

Der Dämon kam auf einem großen Vogel herbeigeflogen und landete direkt auf dem Schiff. Praxis wurde bei dem Versuch zu zaubern von einem Feuerball niedergestreckt, so daß jetzt alles von mir abhing. Ich schnappte mir Praxis' Beutel und machte mich daran, einen Blitz zu zaubern. Aber wie? Ich sah nur acht unterschiedlich feine Pulver vor mir, die in verschiedenen Farben schimmerten. Für einen effektiven Blitz benötigte ich eine Mischung aus Feuer und Wasseressenz, der ich eine Prise Erdesenz hinzugab. Aber was war was? Ich erinnerte mich an Praxis frühere Zauberkunststücke und die jeweilige Farbe und Körnung des Pulvers, das an seinen Händen zurückblieb. Aus diesen Mischfarben konnte ich nach einiger Überlegung auf die Farben der Essenzen schließen. Nachdem ich die Mischung gemixt hatte, wandte ich sie auf den Dread Lord an. Der wurde getroffen und hinterließ den Anvillstein, den er um den Hals trug. Nach unserer Rückkehr zu Astrix fand unsere lange Reise endlich ein glückliches Ende.

Future Wars, der Zeitreise zweiter Teil

...Als erstes möchte ich etwas hinzufügen, was ich in meinen ersten Aufzeichnungen vor Aufregung vergessen habe: Im Kloster, genauer gesagt im Sockel von Lo'Ann's Gefängnis, befand sich eine Gaskapsel, die ich natürlich an mich nahm.

Jetzt aber weiter mit meiner Geschichte. Wie bereits er-

wähnt, die Gegend war ziemlich trostlos. Dennoch fand ich in der unteren rechten Ecke des Geländes nach längerem Suchen einen ausgebrannten Schneidbrenner. Im nächsten Szenario stellte sich das kleine weiße Fleckchen auf dem Dreckhaufen nach mehrmaligem Untersuchen als ein Kasten mit Sicherungen heraus.

Unter dem Haufen, der sich rechts unterhalb der Fundstelle befand, war ein Eingang zur örtlichen Kanalisation verborgen, der nur darauf gewartet hatte, geöffnet zu werden. In der Kanalisation selbst konnte ich immer nur in eine Richtung gehen, was die Orientierung natürlich ungemein vereinfachte. Die Linien am Rand stellten Rohrleitungen dar, und an einer Stelle fand ich einen kleinen Hahn (Tap, sehr klein!), an dem ich den Brenner auffüllen konnte. Ich ging also weiter, und schon bald konnte ich Hilferufe vernehmen. Die Quelle der Schreie war eine hübsche junge Frau, die von einem mutierten Oktopus bedroht wurde. Ich näherte mich der Bestie vorsichtig und brannte ihr mit dem Brenner (sehr sinnig) eins über. Die Frau konnte sich vor Dank kaum halten, gab mir auf meine Fragen Antwort und materialisierte zu allem Überfluß auch noch eine Leiter, mit der ich diesen sehr schlecht riechenden Ort verlassen konnte.

Heraus kam ich vor einer U-Bahn-Station, die den Angriff der Crughons mehr recht als schlecht überlebt zu haben schien. Die Tür öffnete sich jedenfalls erst, als ich das Objektiv der Videokamera mit der Lanze vom Dreck befreite. Das erste, worauf mein Blick nach Betreten der Station fiel, war ein Zeitungsautomat. Er schien genauso zu funktionieren wie die in meiner Zeit auch, also Geld rein und Zeitung raus. Aber woher sollte ich bloß das Geld nehmen? Durch Zufall fand ich eine Münze, die sich im Auswurf verklemmt hatte. Nach zweimaligem Einwerfen der Münze und einigem guten Zureden (der Automat dürfte seine Beulen immer noch haben) bekam ich meine Zeitung. Zwischenzeitlich waren einige U-Bahnen an mir vorbeigefahren, aber sie schienen regelmäßig zu fahren, und so stieg ich in die nächste ein.

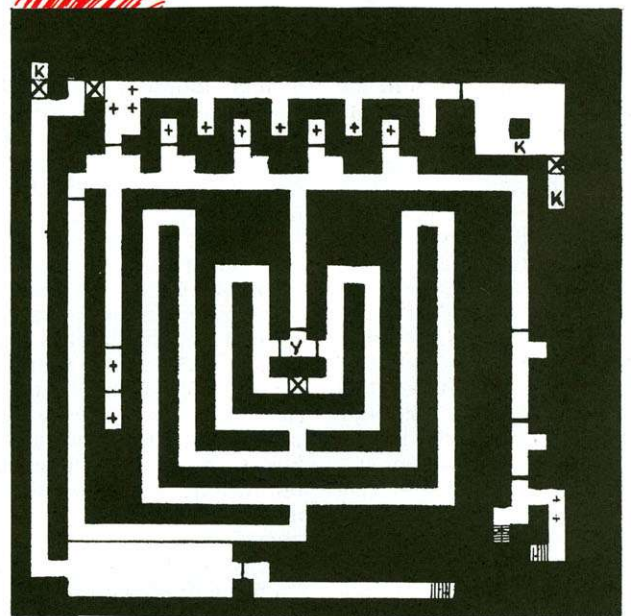
Die Bahn beförderte mich auf direktem Wege zum Flugha-

fen. Hier traf ich auch schon auf das nächste Problem: Der blöde Kontrolleur wollte mich nicht ins Flugzeug lassen, und die freundliche Dame am Schalter hatte offensichtlich Wichtigeres zu tun, als mir weiterzuhelfen. Also stieg ich die Treppen hinunter und besichtigte die Toiletten. Nachdem Klopfen an die Türen nur unwesentliche, aber lustige Erfolge erzielte, wandte ich mich dem Sicherungskasten zu, der an der Wand hing. Ich tauschte die defekte Sicherung aus, ohne zunächst einen Effekt zu bemerken. Doch als ich die Treppen wieder erklimmte, stellte ich fest, daß der Kontrolleur jetzt von dem nun laufenden Fernsehprogramm gebannt war. Ich schlich mich also vorbei. (Hier sollte vorher gespeichert werden, da der Kontrolleur uns bemerkt, wenn wir zu dicht an ihm vorbeilaufen, und dann haben wir keine

Chance mehr.) Der Weg ins Flugzeug stand mir also offen. Doch schon bald erwartete mich die nächste Überraschung: Unser Flugzeug wurde von Crughons überfallen und entführt. Mein Mitreisender verfügte über einen Raketenrucksack und konnte rechtzeitig fliehen. Und ich?

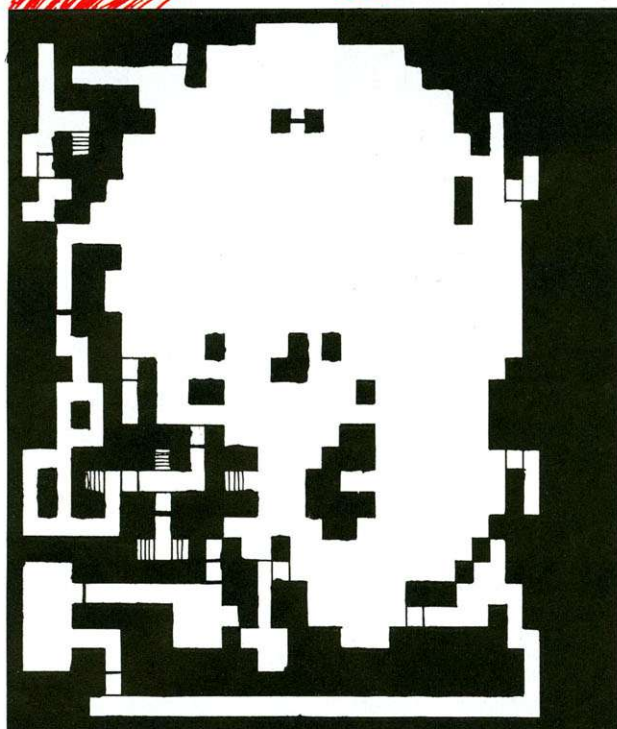
Ich wurde betäubt und wachte in einer sehr kleinen Zelle, in der sich außer einem Bett und einem Luftschacht nichts befand, wieder auf. Die Tür ließ sich durch nichts dazu bewegen, den Weg freizugeben, also mußte ich mir anders helfen. Der Luftschacht. Er war zwar vergittert, aber das ließ sich mit dem kleinen Schlüssel beheben. Leider war er zum Durchkriechen zu klein. Aber die Gaspatrone fand durchaus darin Platz. Damit das Gas nicht in meine eigene Kabine zurückströmte, verstopfte ich die Öffnung mit einer Zeitung. Nach einiger Zeit

Dungeon Master LEVEL 7



Dungeon Master

LEVEL 0



schien das Gas seine Wirkung getan zu haben, auf jeden Fall öffnete sich die Tür, und ich war fast frei. Meine Freudengefühle steigerten sich ins Unermeßliche, als auch noch Menschen das Crughonenschiff übernahmen. Die Freude währte aber nicht lange, denn ich wurde für einen Crughon gehalten. Dafür sprachen die Dokumente, die ich mit mir führte, und die Tatsache, daß mein genetischer Fingerabdruck noch nirgends gespeichert war – wie sollte er auch. Das Gericht wollte mich nicht anhören und verurteilte mich zum Tode durch Erschießen.

Die Jungs waren kurz vorm Abdrücken, als Lo'Ann mit meiner Begnadigung in der Hand heinstürzte und mich so rettete. Aber wie kam sie überhaupt hierher? Ganz einfach, durch einen geringfügigen Fehler war sie ein paar Stunden später gelandet als

ich. Sch... technik! Immerhin, das Gericht entschuldigte sich bei mir, und aus den Dokumenten gingen Position und Zeit der letzten Bombe hervor.

Klar, daß Lo'Ann und ich uns bereit erklärten, die Welt zu retten und die Bombe zu entschärfen. Nachdem ich eine Hypno-Schulung erhalten hatte, bestiegen wir erneut die Chronoport, und wir reisten zirka 65 Millionen Jahre zurück, in die Zeit der Dinosaurier. Wie ich bereits erwähnte, gibt es Tage, an denen nichts klappt. Als wir ankamen, waren die Crughons schon wieder am Wegfliegen. Lo'Ann überreichte mir ein Airgun und machte sich auf den Weg. Da die Bombe nicht mehr zu entschärfen ist, wenn sie erst einmal installiert wurde, blieb uns nichts anderes übrig, als den Kampf mit der Übermacht aufzunehmen und die Crughons auszuschalten.

Die folgenden Gedanken, die mir kamen, in denen es um Mäuse und Klicken ging, sind mir heute noch nicht klar, aber sie schienen den Adressaten vor der Mattscheibe erreicht zu haben, wo immer das auch sein mochte. Das Prinzip des Kampfes war ganz einfach (trotzdem sollte zur Sicherheit gespeichert werden). Die Crughons mit dem roten Rahmen befanden sich in den Positionen, in denen sie uns treffen konnten. Das war der Bereich um den großen Scheinwerfer herum. Diese mußten also sofort ausgeschaltet werden. (Sind einmal keine "roten" Gegner mehr auf dem Schirm, sollte man die Lücke um die "roten Positionen" vergrößern. Auch der kleine Crughon auf der Schwebplattform ist ein nettes Ziel, muß aber nicht unbedingt abgeschossen werden.)

Wir waren schon der Meinung, daß wir alle erledigt hätten, als doch noch einer übrigblieb und Lo'Ann erwischte. Doch sie lebte noch. Nachdem ich meine Scheu überwunden hatte, durchsuchte ich sie und fand eine Unsichtbarkeitspille und ein Medaillon, mit dessen Hilfe ich mit Albert II in Verbindung treten konnte, der Lo'Ann in die Zukunft zurückholte. Ganz auf mich allein gestellt begab ich mich also zu dem Crughon-Raumer und durchsuchte zunächst den Toten direkt vor der Luke. Ich förderte eine Magnetkarte zutage, die ich im Cockpit in den dazu gehörenden Leser schob. Der Glaskasten ließ sich nun leicht öffnen, und ich nahm das Kleidungsstück an mich. Da ich mich durch die Kamera beobachtet fühlte, verhängte ich sie damit. Nachdem ich mich in den Kasten gelegt hatte, machte ich meinen ersten Raumflug mit Hyperraumeinlage durch.

Gelandet wurde das Schiff in der Hauptzentrale der Crughons. Um von den Crughons nicht entdeckt zu werden, öffnete ich den Kasten, stand auf und stellte mich neben die Tür. Als sich diese öffnete, schluckte ich Lo'Ann's Pille und wurde unsichtbar. Da die Wirkung nicht sehr lange anhielt, mußte ich mich sehr beeilen. Ohne die Wachen zu berühren, verließ ich das Schiff und versteckte mich bei den Kisten. Eine der Kisten ließ sich öffnen, und so kletterte ich hinein. Als ich die Kiste wie-

der öffnete, fand ich mich in einem Lagerraum wieder. Die vertraute Stimme von Albert II meldete sich wieder und eröffnete mir, daß ich genau sechs Minuten Zeit hatte, den Computerraum zu finden und die Bombe umzuprogrammieren. Die Menschen waren nämlich gerade dabei, die Station anzugreifen, und konnten sich nicht länger halten. Ich machte mich also auf den Weg durch das Schiff. (Da die Zeit wirklich sehr knapp bemessen ist, empfiehlt es sich, eine Karte zu zeichnen).

Im Computerraum steckte ich die Karte in den linken Terminal, und Albert managte den Rest. In der restlichen Zeit mußte ich jetzt zurück zum Hangar, der Tür mit den Pfeilen, spurten. Der Rest ist Geschichte. Die Archäologen würden sich allerdings drum reißen, aber mehr wird nicht verraten.

Kurz und knapp – ein Sack voll Minutips

Wer bei »Rotor« immer noch in den niederen Missionen oder gar noch in den Simulationen herumkraucht, wird sich sicher über diese Passwörter freuen, durch die auch höhere Levels zugänglich werden.

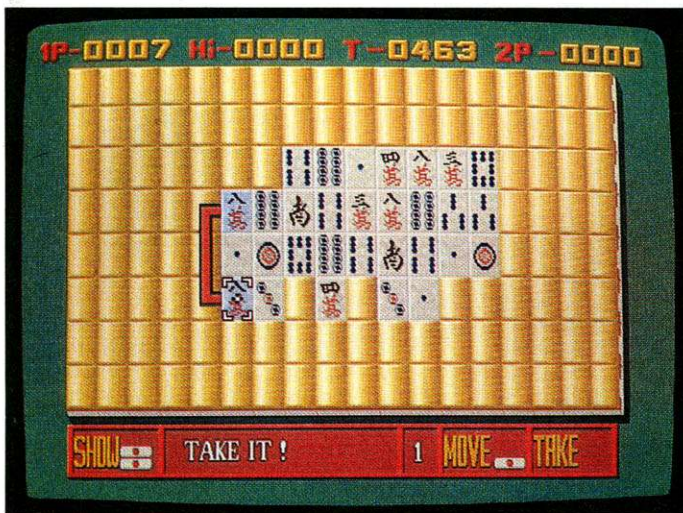
PIT
GAG
LIP
SLY
MEW
AWE
TNT

Alle, die bei »Hybris« den Spielablauf ein wenig manipulieren möchten, sollten während der Aufforderung, einen Commander zu wählen, statt des Feuerknopfes die Leertaste betätigen. Daraufhin gelangt man in ein Menü, in dem von der Anzahl der eigenen Schüsse bis zum Intervall, in dem die Erweiterungscontainer erscheinen, so ziemlich alles ausgewählt werden kann, was das Herz begehrt.

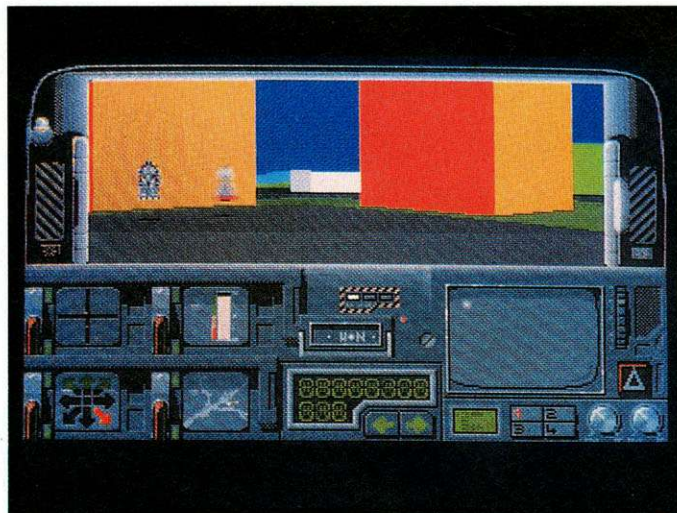
Damit sind wir für diesen Monat am Ende der Spieletips angekommen. Wir hoffen, auch für Ihre Probleme den einen oder anderen Tip dabei gehabt zu haben.

Also, viel Spielspaß bis nächsten Monat.

(Robert Marz/hs)



Fallen erhöhen den Schwierigkeitsgrad erheblich



Verbrecherjagd in ferner Zukunft

Lin Wu's Challenge

Mah Jongg ist ein altes japanisches Brettspiel, das sich mit unserm Memory vergleichen läßt und viele Varianten, wie zum Beispiel Shanghai, besitzt.

Bei Lin Wu's wird eine Anzahl von Spielsteinen, die mit unterschiedlichen Symbolen bemalt sind, nebeneinander auf dem Spielbrett angeordnet. Die Aufgabe des Spielers ist es nun, das Spielfeld zu leeren, indem jeweils zwei zueinander passende Steine entfernt werden. Dazu müssen sie aber über eine Linie verbunden werden können, die maximal zwei Ecken haben darf. In ausweglosen Situationen kann der Spieler auf drei Hilfsfunktionen zurückgreifen, die aber nur begrenzt zur Verfügung stehen.

Da gibt es die Show-Funktion, die eine mögliche Verbindung zwischen zwei Teilen anzeigt, die Take-Funktion, mit der man zwei Teile auch über mehr als zwei Ecken verbinden kann und eine Move-Funktion, die es erlaubt, Teile zu verschieben.

Ziel des Spiels ist es, sich gegen Meister des Spiels an verschiedenen Schauplätzen in der Welt nach oben zu kämpfen.

Lin Wu ist ein spannendes Strategiespiel, das unter Zeitdruck gespielt wird. Aber es existieren einige Mängel. Es gibt keine Möglichkeit, in höhere Levels einzusteigen, weder über Passwörter noch über gespeicherte Spielstände. Der Cursor muß auch im Ein-Spieler-Modus mit dem Joystick

bewegt werden, und es sind immer wieder Griffe zur Tastatur nötig.

(Robert Marz/mm)

AMIGA DOS Blitzlicht

Name: Lin Wu's Challenge
Hersteller: Lasersoft
Quelle: Fachhandel
Preis: 69,95 DM

Positiv:

- hübsche Grafik
- spannendes Spielgeschehen
- zwei Spieler gleichzeitig
- Verpackung aus massivem Holz

Negativ:

- Steuerung nur über Joystick
- keine Möglichkeit zum Speichern beziehungsweise kein Passwortsystem
- in höheren Levels zu wenig Zeit



Resolution 101

Durch eine rapide wachsende Verbrechensrate sah sich die Regierung der USA zu drastischen Maßnahmen genötigt. Resolution 101 wurde erlassen...

Dieses Papier besagt, daß jedem Gefangenen Amnestie gewährt wird, der einen anderen A-Klasse-Kriminellen zur Strecke bringt. Auch Ihr Bit-Alter-Ego bekommt seine Chance. Sie sollen die kleine Stadt Los Vegas von vier Drogendealern, die dort ihr Unwesen treiben, befreien. Der Staat stellt Ihnen dazu einen mit einer Licht-Maschinen-Pistole bewaffneten Ground Skimmer Theta 4000 zur Verfügung. Das Fahrzeug ist versichert, so daß Ihr Skimmer bei Verlust fünfmal ersetzt wird. Die Dealer sind ebenfalls versichert, sie kommen dreimal wieder.

Sie fliegen in Ihrem Skimmer mit einer hohen Geschwindigkeit durch die in 3D-Vektorgrafik gezeichnete Stadt und verfolgen den Dealer, der eine bestimmte Route abfliegt und seine Drogen in Form von roten Pillen verteilt. Zu seinem Schutz kann er Bandenmitglieder herbeirufen, die Ihnen das Leben ganz schön schwer machen, zumal Sie den Dealer erst dann dingfest machen dürfen, wenn Sie genügend Beweise, sprich Drogen, eingesammelt haben. Abgeschossene Fahrzeuge hinterlassen Geld oder Drogen, die Sie natürlich einsammeln können. Mit dem Geld können Sie Ihren Skimmer in Shops, die es in jedem Stadt-

viertel gibt, aufrüsten oder reparieren lassen.

»Resolution 101« ist ein sehr schnelles Ballerspiel, das ein gewisses Maß an Taktik erfordert und den Spieler nicht zuletzt wegen seiner sehr exakten Steuerung viele Stunden Ballerspaß zu beschern vermag.

(Robert Marz/hs)

AMIGA DOS Blitzlicht

Name: Resolution 101
Hersteller: Millenium
Quelle: Fachhandel
Preis: 84,95 DM

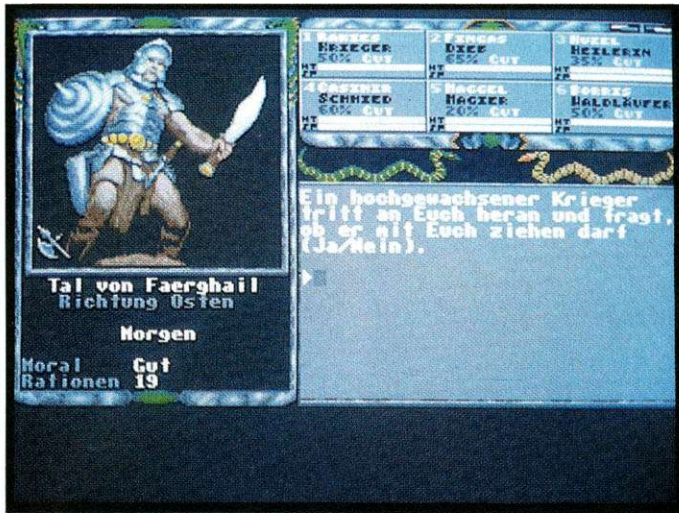
Positiv:

- schnelle 3D-Vektorgrafik
- verschiedene Waffensysteme
- exakte Steuerung

Negativ:

- wenig Musik





In der Wildnis wimmelt es von gefährlichen Monstern

Legend of Fearghail

Krieg herrscht zwischen den Elfen und den Menschen der Grafschaft Thyn. Die Menschen sind unterlegen, und so sucht der Graf nach wackeren Helden, die Hilfe aus der benachbarten Grafschaft anfordern.

Natürlich fühlen Sie sich als junger, aufwärtsstrebender Abenteurer dazu berufen, ihre Heimat zu retten. Nachdem Sie dem Grafen Ihre Dienste angeboten haben, werben Sie in der Taverne eine Hilfstruppe an. Danach wird noch der Laden aufgesucht, wo die Helden ausgerüstet werden. Nun kann das Abenteuer beginnen.

Der Aufbau von Legend erinnert ein wenig an den von »Bard's Tale«, ist grafisch aber ansprechender gestaltet. Die Kommandos sind über Tastatur und per Maus erreichbar, so daß für jeden Geschmack das Richtige dabei sein sollte. Die Anzahl der Gegenstände, die eine Spielfigur tragen kann, richtet sich hauptsächlich nach ihrem Gewicht. Ein schwacher Held kann nicht so viel tragen wie ein starker.

Schon bald trifft die Gruppe auf einen Fremden, der, einmal aufgenommen, die Helden schlagkräftig unterstützt und sogar Tote wieder zum Leben erwecken kann. In den Zwergenminen ist eine Kristallkugel versteckt, die das Kartographieren für den Spieler übernimmt. Ein umfangreiches Magiesystem rundet den positiven Eindruck ab, der nur von dem etwas anti-

quierten Kampfsystem beeinträchtigt wird. Da es komplett in Deutsch gehalten wurde, stellt es eine willkommene Alternative zu anderen Rollenspielen dar, die ja meistens englischsprachig sind.

(Robert Marz/hs)

AMIGA DOS Blitzlicht

Name: Legend of Fearghail
Hersteller: ReLine
Quelle: Fachhandel
Preis: 89,95 DM

Positiv:
– umfangreiches Magiesystem
– komplett in Deutsch
– viele Details

Negativ:
– einfaches Kampfsystem



Per Panzer auf Verbrecherjagd in der Zukunft

Dark Century

Im 23. Jahrhundert sind die Gefängnisse von der Erde auf kleine Satellitenfabriken verlagert worden, in denen die Häftlinge arbeiten müssen. Eigentlich ist alles in Butter, bis einige der Verbrecher fliehen...

Eine kleine Gruppe von Kriminellen hat ein paar der Panzer, die auf ihrem Satelliten produziert werden, in ihre Gewalt gebracht. Die Panzer wurden so präpariert, daß sie explodieren, sobald sie die Grenzen der Fabrik erreichen.

So könnten durch die zerstörten Zäune viele weitere Häftlinge fliehen, es sei denn, es gelingt, die Panzer vorher unschädlich zu machen.

Zu Beginn des Spiels wird über die Anzahl der Spieler und Panzer entschieden und darüber, wer wieviel davon erhalten soll.

Da man immer nur einen Panzer gleichzeitig steuern kann, sind die Panzer programmierbar. Sie fahren ihrem Programm entsprechend selbstständig über das Spielfeld und halten nach Gegnern Ausschau.

Das Drumherum von »Dark Century« ist gut gelungen und grafisch vielversprechend gestaltet. Das Spiel selbst ist allerdings sehr flach. Die Darstellung des Spielfeldes beschränkt sich auf die Hälfte des Bildschirms. Treten zwei Spieler gleichzeitig gegeneinander an, wird auf der zweiten Monitorhälfte das Spielfeld des zweiten Spielers dargestellt. Fährt man Kurven, bewegt sich der Hintergrund,

und ab und zu erscheint ein eigener oder gegnerischer Panzer oder eine Energieblase, mit deren Hilfe man seine Vorräte aufstocken kann. Schade, daß das gute Drumherum und vor allem das Intro ein so laues Spiel übertönen sollen.

(Robert Marz /hs)

AMIGA DOS Blitzlicht

Name: Dark Century
Hersteller: Titus
Quelle: Fachhandel:
Preis: zirka 85 DM

Positiv:
– gutes Intro
– Handbuch ausführlich und in Deutsch

Negativ:
– Spiel selbst sehr flach
– zu wenig Abwechslung



Ein Nomadenstamm, die Orodrid, fanden eines Tages das Skelett und bauten sich Wohnhöhlen in die Rippen und Knochen des toten Orodroiden. Mit der Zeit wurde die Stadt eine sehr wohlhabende Festung, und es entwickelte sich eine eigene Religion. Den Priestern war das Labyrinth des Gehirns, das Tismit, vorbehalten. Das Geheimnis, daß die ehemaligen Gehirnzellen zu Kristallen versteinert sind, die, wenn man sie zusammenbringt, zu pulsieren beginnen, ist ebenfalls nur ihnen bekannt. Einige besonders aktive Kristalle können von einem Menschen mit hervorragenden geistigen Eigenschaften zum Leben erweckt werden. Dann entsteht ein Crystallion. Das Crystallion ist ein schimmerndes Kristallpferd. Es ist kein normales Reittier, sondern ein Freund und Lebensgefährte. Nur jenen, die ihrem vorbestimmten Crystallion begegnen, ist es vergönnt, im Rat von Orodrid einen Platz zu erhalten. Sie, als junger, aufstrebender Führer der Familie Adreni, sind der Meinung, daß Sie es schaffen können. Doch um zum Geburtsort des Crystallions zu gelangen, müssen Sie alle vier Ebenen des Tismits durchqueren und dessen Geheimnisse lösen.

Über dieser Aufgabe dürfen Sie natürlich nicht Ihre Familie vernachlässigen, deren Führer Sie sind. Die Familie muß mit Rohstoffen für die Produkte versorgt werden, die sie herstellt, und natürlich muß sie auch genügend zu Essen haben. Nebenher muß aber noch genügend Geld erwirtschaftet werden, um den Wächtern des Tismits die regelmäßig von ihnen geforderte Summe "spenden" zu können. Ausgangsposition des Spiels ist immer die Stadt. Hier können Sie wählen, was Sie als nächstes unternehmen wollen. Zum Beispiel können Sie von hier aus das Tismit betreten oder sich im Bosu, einem Strategiespiel, üben.

Sie können aber auch Ihre telepathischen Fähigkeiten im Deketa verfeinern. Das Deketa ist ein spezielles Kartenspiel, das die Konzentrationsfähigkeit steigern soll. Dabei gibt es sechzehn Karten, von denen jeweils zwei mit einer Zahl von eins bis acht versehen sind und die mit der Rückseite nach oben auf den Tisch gelegt werden. Ziel ist es jetzt, jeweils zwei Karten mit der



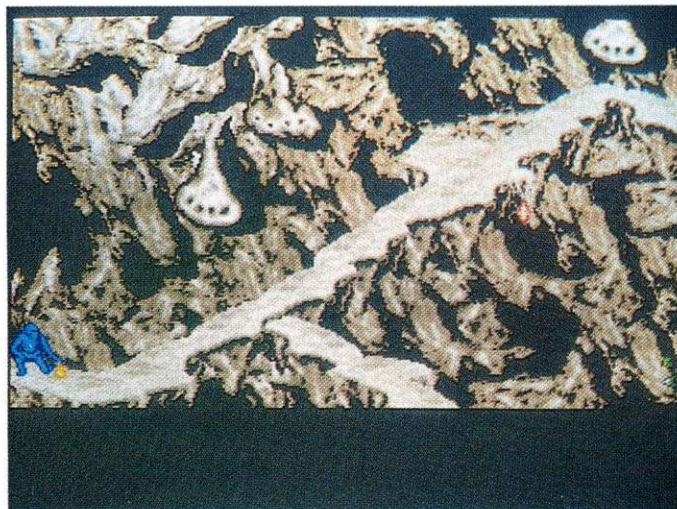
Im Kartenspiel Deketa können Sie Ihre Fähigkeiten erweitern

Knights of the Crystallion

Vor Millionen von Jahren durchkreuzte ein Orodroid, ein unglaublich großes Tier, die Meere des Planeten. Eines Tages verfiel es sich in einem schmalen Flußtal und mußte elendig verhungern. Sein Skelett überdauerte die Zeit, und seine Gehirnzellen erstarrten zu Kristallen.

gleichen Nummer aufzu decken. Mit zunehmenden Erfolgen sieht man ab und zu die Nummern schwach durch die Deckel durchschimmern. Durch Deketa kann der Spieler zum ersten Mal Verbindung mit seinem Crystallion aufnehmen, das einem dann den Weg im Tismit weist. Im fortgeschrittenen Stadium kann man sich sogar hinausteleportieren. Im Tismit bekommt man einen Kristallan-

zug, der zum einen leuchtet und zum anderen die Fähigkeit besitzt, seinen Träger wieder zum Leben zu erwecken. Die Anzahl der Ladungen läßt sich im Proda erhöhen, indem man mit im Tismit aufgesammelten Kristallen Energie erzeugt und diese in den Anzug speist. Im Tismit selbst bewegen Sie sich mit Hilfe der Maus und verteidigen sich mit glühenden Plasma-Bällen gegen alle Ar-



Das Tismit ist ein weitläufiges Labyrinth im Schädel des Orodroiden

ten von Monstern, die Ihnen ans Leder wollen und ebenso auf die herumliegenden Kristalle aus sind wie Sie auch. Jedesmal, wenn Sie eine Ebene des Tismits erforscht haben und in die nächste vordringen wollen, stellen Ihnen die Wächter Aufgaben, die Sie lösen müssen. Unter anderem müssen Sie einen Meister im Bosu, einem Strategiespiel, schlagen, was natürlich vorher geübt werden kann, und eine Frage aus einem dem Spiel beiliegenden Gedichtband beantworten. »Knights of the Crystallion« ist ein sehr vielseitiges und komplexes Spiel, das Komponenten aus fast allen Spielgenres gut in sich vereint.

Grafisch haben sich die Macher von »Knights of the Crystallion« alle Mühe gegeben; farbenfrohe Illustrationen der Schauplätze und kinoreife Animationen runden das Spiel zu einem optischen Genuß ab. Lediglich der Sound hinkt dem Gesamteindruck hinterher. Obzwar ambitioniert und phantasievoll gemacht, beginnen die stets dumpfrhythmischen Klänge schnell an den Nerven des Spielers zu sägen.

Trotz kleiner Mängel bleibt ein rundes Spielvergnügen, das sich sowohl als Denk- als auch als Geschicklichkeitsprobe entpuppt.

(Robert Marz/hs)

AMIGA DOS Blitzlicht

Name: Knights of the Crystallion
Hersteller: US Gold
Quelle: Fachhandel
Preis: 99,95 DM

Positiv:

- gute Grafik
- sehr vielseitig
- komplett in Deutsch
- Gedichtband und Musik-kassette im Lieferumfang

Negativ:

- Sound wird mit der Zeit nervend





Projectyle - das Sportspiel einer fernen Zukunft

Projectyle

Im 21. Jahrhundert wird Fußball als Kultsport ausgedient haben, denn dann gibt es Projectyle. Man nehme sechs Tore, drei Mannschaften zu je fünf Mann und eine Kugel und verteile sie auf ein Spielfeld mit fünf Feldern, die sternförmig angeordnet sind.

In jedem Feld ist ein Spieler von jeder Mannschaft postiert, in drei der Randfelder besitzt jede Mannschaft ein Tor, im vierten Randfeld sind dagegen drei Tore. Ziel des Spiels ist es, eine Kugel, eben jenes "Projectyle", in ein Tor eines Gegners zu schießen. Die Kugel wird in einer Art Billard von den Spielern angestoßen und folgt den Reflektionsgesetzen. Von Zeit zu Zeit tauchen Symbole auf dem Spielfeld auf, die das Verhalten des Balls beeinflussen oder Geld bedeuten. Mit diesem Geld kann man seinen Spielern auch ein Training zur Verbesserung der Spielleistung spendieren.

Technisch kann man gegen »Projectyle« eigentlich nichts einwenden – im Gegenteil! Besonders Auge und Ohr werden verwöhnt. Zwar haben die Spielfelder alle die gleiche Struktur, ihre grafische Gestaltung ist jedoch variabel und farbenfroh, gleiches gilt für die Sprites. Bemerkenswert ist auch der Sound, der entweder aus fetziger Musik (acht verschiedene Titel) in sehr guter Sample-Qualität oder aus Spielgeräuschen besteht. Andauernd nur gegen

den Computer zu spielen wird mit der Zeit zwar langweilig, jedoch verschaffen insbesondere die Optionen für mehrere Spieler dem Spiel einen zusätzlichen Reiz.

(Michael Anton/hs)

AMIGA DOS Blitzlicht

Name: Projectyle
Hersteller: Electronic Arts
Vertrieb: Fachhandel
Preis: voraussichtlich 80 DM

Positiv:
– guter Sound
– einfach, aber effektiv

Negativ:
– ohne Partner etwas langweilig



Im Osten nix Neues, dem Ninja-Kämpfer fehlt's an Pep

Ninja Spirit

Der Geist eines Tsukikages, eines weißen Wolfs, hat die Form eines Ninja angenommen, um dem Bösen den Kampf anzusagen.

Kein Zweifel, Sie als Spieler sollen mal wieder dazu animiert werden, gegen das Böse der Welt zu kämpfen. Der Ninja macht sich also auf seinen gefährlichen Weg durch die von rechts nach links scrollende Landschaft und beseitigt alles, was den Mut hat, sich ihm in den Weg zu stellen. Dazu hat er die Auswahl zwischen verschiedenen Waffen, die über die Tastatur anwählbar sind. Am Anfang verfügt er über sein Schwert, Shuriken (Wurfsterne), Wurfbeile und eine Axt, die er an einer Kette immer wieder zurückholen kann. Besiegt er auf seinem Weg eine besondere Art von Ninja, so erhält die momentan aktivierte Waffe mehr Durchschlagskraft oder wird auf andere Art wirkungsvoller. Bei den Shuriken ist es dann möglich, drei Sterne gleichzeitig zu treffen, während das Schwert eine größere Reichweite erhält.

Wird man der zahlreichen Gegner nicht müde und hält das Spiel einige Zeit durch, so gelangt man, wie sollte es auch anders sein, zu einem Oberbösewicht, den es natürlich auch zu eliminieren gilt. So zieht sich das routiniert gemachte Spiel, ohne wesentliche Höhepunkte aufzuweisen, unerhört in die Länge.

»Ninja Spirit« ist die Umsetzung des gleichnamigen Au-

tomaten und hat dem verwöhnten Spieler wenig Neues zu bieten, so daß eher Langlebigkeit als Spielspaß an der Tagesordnung ist. Hinzu kommt eine mäßige Grafik, verbunden mit mäßigem Sound und langen Ladezeiten, so daß das Geld nur für echte Fans des Schwertkampf- und Prügel-spielgenres gut investiert ist.

(Robert Marz/hs)

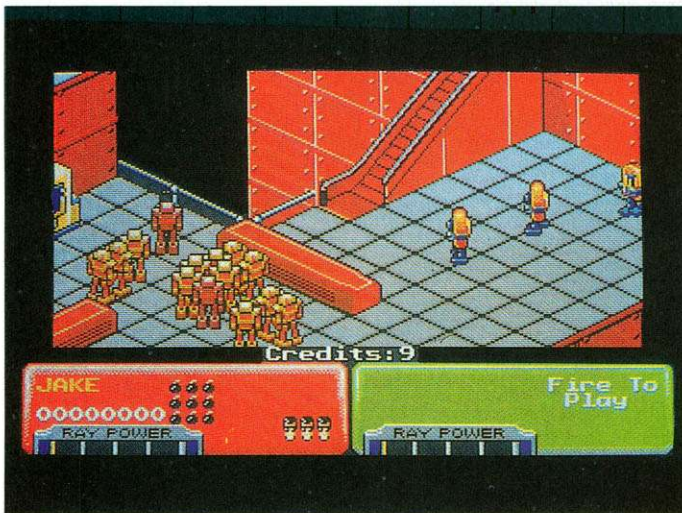
AMIGA DOS Blitzlicht

Name: Ninja Spirit
Hersteller: Activision
Quelle: Fachhandel
Preis: 84,95 DM

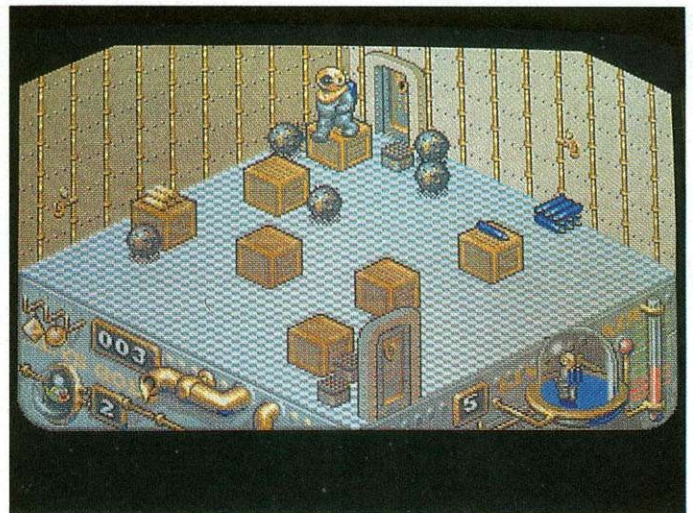
Positiv:
– Anleitung deutsch

Negativ:
– wenig Abwechslung





Abenteuer auf dem Planeten der Robot-Monster



Die Goldschätze der Esmeralda gilt es zu heben

ESCAPE FROM THE PLANET OF THE ROBOT MONSTERS

Die bösen Reptilions haben Professor Bellum und zahlreiche Geiseln auf den Planeten X entführt und zwingen sie dort, eine Waffe zu bauen, mit der die Erde vernichtet werden kann.

Die beiden Spezialagenten Jake und Duke werden mit der Mission beauftragt, die Geiseln zu befreien und die Fabriken auf dem Planeten X zu zerstören, bevor es zu spät ist. Sie machen sich also an die Arbeit.

Wenn Sie einen Freund zur Hand haben, kann dieser die Rolle von Duke übernehmen, und Sie können den Kampf gemeinsam aufnehmen. Die Kämpfer betreten die Fabriken des Planeten und machen sich auf die Suche nach den Geiseln, die sie durch Berührung befreien. In Schränken befinden sich Akten oder zusätzliche Bomben, die die Ausrüstung vervollständigen. Mit Hebeln können Rolltreppen aktiviert und Geiseln aus Glaskäfigen befreit werden. Daß die Reptilions die Zerstörung ihrer Fabriken nicht so einfach hinnehmen, dürfte klar sein. Mit jedem Schuß läßt aber die Stärke nach, die mit Hilfe kleiner grüner Kristalle, die die Roboter hinterlassen, wieder erlangt werden kann. Bei Escape kann man mal wieder so richtig nach Herzenslust ballern und den

Robotern eins drauf geben.
(Robert Marz/hs)

AMIGA DOS Blitzlicht

Name: Escape from the Planet of the Robot Monsters
Hersteller: Domark
Quelle: Fachhandel
Preis: 64,95DM

Positiv:

- hübsche Animationen
- Zwei-Spieler-Modus
- Anleitung in Deutsch
- Continue-System mit Credits

Negativ:

- Steuerung gewöhnungsbedürftig



Treasure Trap

Ein Arcade-Adventure im klassischen Stil ist für den Amiga angesagt. Kann die Suche nach verborgenen Schätzen auch den Computerspieler von heute noch begeistern?

Es ist schon etwas länger her, daß die HMS Esmeralda mit einer riesigen Ladung Gold an Bord in den Tiefen des Ozeans versunken ist. Im Laufe der Zeit haben sich in dem Wrack jedoch die lokalen Vertreter der Meeresflora und -fauna breitgemacht, so daß sich die Bergung der Schätze als recht schwierig erweist. Ein einsamer Taucher übernimmt die Aufgabe, alle Goldbarren zu bergen. Dabei stehen ihm nicht nur die bereits erwähnten Gegner in Form von Quallen, Piranhas und ähnlichem Ungetier, sondern auch so profane Probleme wie die Suche nach Sauerstoff entgegen. Erschwerend wirkt auch, daß die Schätze nicht immer leicht zugänglich sind, so daß Gehirnakrobatik angesagt ist.

Von Technik und Inhalt her ist »Treasure Trap« eigentlich ein alter Hut, mit diesem Prinzip hat Ocean Soft schon vor Jahren auf 8-Bit-Rechnern für Furore gesorgt. Die Umsetzung für den Amiga ist sehr gut gelungen, Grafik und Sound sind bemerkenswert. Leider wurden auch die Schwächen des Prinzips übernommen, stellenweise wird es etwas unübersichtlich. Die dadurch verlorenen Leben frustrieren etwas, die (nicht nur mit dem Joystick) mehr als gewöhnungs-

bedürftige Steuerung und die etwas ungenauen Kollisionsabfragen tragen ebenfalls dazu bei. Alte Hasen, die bereits mit »Batman« (noch vor der Kino-Kommerz-Version!) oder »Head over Heels« gut zurechtgekommen sind, finden auch hier schnell eine Heimat.

(Michael Anton/hs)

AMIGA DOS Blitzlicht

Name: Treasure Trap
Hersteller: Electronic Zoo
Vertrieb: Fachhandel
Preis: voraussichtlich 80 DM

Positiv:

- gute Gestaltung
- effektive Nutzung von Speichererweiterungen
- Spielstände speicherbar

Negativ:

- Spielprinzip etwas betagt



Dort plagt man sich als erobersfreudiger Thronerbe eines Kleinstaates sowohl mit dem täglichen Regierungskram als auch mit Trollen, Riesen, Meuchelmördern und anderen unfreundlichen Zeitgenossen ab. Um den Glanz des alten Reiches Aragon wieder aufleben zu lassen, ist es die Hauptaufgabe des neuen Königs, eine möglichst große und schlagkräftige Armee auf die Beine zu stellen. Nur so kann schnell an viel Geld kommen und sein Punktekonto ergänzen, was den Charakteren zu höheren Levels und damit zu mehr Effizienz verhilft. Durch Heldentaten (das heißt hauptsächlich Monster meucheln) vergrößert sich schnell die Zahl der Paladine, die in des Königs Dienste treten wollen. Die Dankbarkeit der Bevölkerung steigt in gleichem Maße, und damit erhöhen sich sowohl die Einkünfte als auch die Zahl der Rekruten.

Neben den obligatorischen Feldzügen gegen andere Regenten mit Expansionsgelüsten muß sich ein Herrscher allerdings auch um die heimliche Bevölkerung kümmern, indem er die verschiedenen Wirtschaftszweige fördert, Steuern festsetzt, Recht spricht, die Hauptstadt ausbaut, befestigt und eine Garnison zur Verteidigung einrichtet. Diese Aktionen gehen in monatlichen Intervallen vor sich. Die Ergebnisse der Regierungsarbeit sind ebenso wie der Zustand der Truppen in gut gegliederten Menüleisten abzulesen.

Die durch einen Zufallsgenerator bestimmten Ereignisse sorgen für reichlich Abwechslung und Action. Ob man auf dem Weg zur Nachbarstadt von ständig neuen Monsterhorden angegriffen wird, ein Monat Hagel die gesamte Ernte vernichtet oder für die Befreiung des von Goblins entführten Kaufmannssohnes eine stattliche Prämie ausgesetzt wird: Kein Spiel gleicht dem anderen. Deshalb ist häufige Benutzung der leider etwas umständlich ausgefallenen Save- beziehungsweise Restore-Funktion ein Muß, will man keine unliebsamen Überraschungen erleben. Die Zufallsfunktion bedingt aber leider auch eine nicht immer ganz logische Abfolge im ökonomischen Bereich, was den Spieler manchmal zur Verzweiflung treiben kann.



Eine strategische Kriegssimulation, die in einem fantastischen Mittelalter spielt

Sword of Aragon

Neues vom Rollenspielmarkt: Die für ihre strategischen Kriegssimulationen bekannte Firma SSI hat als Hintergrundscenario diesmal keinen Ost-West-Konflikt gewählt, sondern den Kampf um die Macht in ein mittelalterliches Fantasy-Land verlegt.

Das gleiche schafft allerdings auch die automatische Steuerung bei militärischen Auseinandersetzungen. Die Kampfszenen, während derer man nicht eingreifen kann, dauern oft ziemlich lang, ohne daß Nennenswertes auf dem Bildschirm geschieht. Besonders frustrierend ist es, wenn sich Truppenteile vom Kampfgeschehen absetzen und durch nichts in der Welt dazu zu bewegen sind, zurückzukehren und ihre "Pflicht" zu tun.

Priester und Ranger können während des Kampfes der Truppe mit (je nach Level mehr oder minder) hilfreichen Zaubersprüchen beistehen. Die taktische Aufstellung vor der Schlacht ist laut (schlecht gegliedertem) Handbuch zwar sehr wichtig, die von der Automatik gesteuerten Charaktere tun dann im Ernstfall allerdings doch nur das, was sie gerade für richtig halten. Der Verlust eines Lebens zieht den Verlust eines

Levels nach sich. Abbruch des Kampfes ist erst nach einer gewissen Anzahl von Zügen möglich, taktischer Rückzug wird dadurch ausgeschlossen, hat die Aktion erstmal begonnen.

Die Steuerung per Maus und Tastatur ist gut durchdacht, es stehen reichlich Menüpunkte zur Auswahl. Der Sound dagegen beschränkt sich fast ausschließlich auf Kampfgeräusche und klagendes Jodeln der Spellcaster. Grafikmäßig haben sich die Programmierer leider nicht allzuviel einfallen lassen. Die Städte sind zwar recht hübsch dargestellt, das ziemlich unübersichtliche Terrain jedoch wird leider von nur sehr einfach gemachten, minimal animierten Gestalten bevölkert, die reichlich chaotisch durch die Landschaft ruckeln.

Trotz allem ist »Sword of Aragon« ein spannendes Programm, an dem nach dem etwas mühseligen Anfang gerade Einsteiger viel Freude haben dürften. Insbesondere die Idee, ein kriegerisches Strategiespiel einmal nicht in irgendeinem Weltkrieg stattfinden zu lassen, verdient Lob. Obwohl man sich eigentlich darüber im klaren sein sollte, daß ein feuerspeiender Drache aus einem Märchenland im Grunde nur ein verkleiderter Flammenwerfer ist.

(A. Hink/hs)

AMIGA DOS Blitzlicht

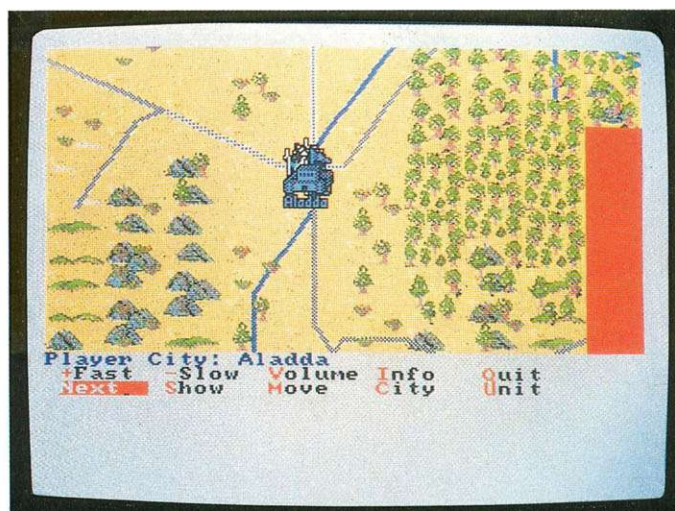
Name: Sword of Aragon
Hersteller: SSI
Quelle: Fachhandel
Preis: voraussichtlich 90 DM

Positiv:

- übersichtlich aufgebaut
- auf Harddisk installierbar
- schneller Bildschirm-aufbau

Negativ:

- lange Wartezeiten bei Kampfszenen
- nicht immer ganz logisch



Feldzüge sind die eine Seite des Spiels, wirtschaftliche Entscheidungen die andere



Fantasy-Rollenspiele erfreuen sich rund um die Welt ständig wachsender Beliebtheit. Der Spieler übernimmt hier die Aufgabe, eine Truppe von Helden zu steuern. Dies geschieht innerhalb einer ausgedehnten Spielwelt, die im Spielverlauf erkundet werden muß. Rollenspiele entstanden als papiergebundene Gesellschaftsspiele; jedoch zeigte sich recht bald, daß diese Art Spiel sich ausgezeichnet für eine Umsetzung als Computerspiel eignet. So dauerte es nicht lange bis TSR, Hersteller von »Dungeons & Dragons«, dem beliebtesten papiergebundenen Rollenspiel, und S.S.I., renommierter Hersteller von Strategiecomputerspielen, gemeinsam ein Dungeons & Dragons-Computerspiel initiierten. »Pool of Radiance«, so der Titel dieses ersten Spiels, wurde ein erstaunlicher Erfolg und es folgten in rascher Folge weitere Dungeons & Dragons-Computerspiele. Obwohl »Champions of Krynn« eigentlich schon das dritte Produkt aus dieser Serie ist, stellt es für Amiga-Benutzer das erste D&D (Dungeons & Dragons) Rollenspiel dar.

Hoch im Norden der Welt Krynn liegt ein kleines Fort. Weit vorgeschoben im Niemandsland, das noch während des Lanzenkrieges von Drakonianern beherrscht wurde, sichert dieser Außenposten nun die Siedlungen der Menschen gegen die verbleibenden bösen Horden. Hier nehmen auch die Abenteuer der sechs Helden, die der Spieler führen muß, ihren Anfang.

Die Truppe verpflichtet sich beim Fortkommandanten als Späher und erhält sogleich einen ersten Auftrag. Der Befehlshaber einer Patrouille, die den Auftrag hatte, die nahegelegene Stadt Throtl zu erkunden, muß dringend wieder zurück zum Außenposten beordert werden. Die Helden brechen zu besagter Stadt auf und stellen fest, daß hier das Hauptquartier einer Horde Kobolde ist, die den Helden den Eintritt in die Stadt verweigern. Ist dieses Empfangskommando überwunden, steht Throtl den Abenteurern offen. Die Stadt ist verlassen und wird von vielen bösen Wesen als Unterschlupf benutzt. Tatsächlich gelingt es, den gesuchten Caramon zu finden. Seine Patrouille wurde in einen Hinterhalt gelockt



Spannende Abenteuer erwarten Sie in der Spielwelt Krynn

Champions of Krynn

Der Krieg der Drachenlanze ist vorbei. Es gelang den Menschen, die Eroberungspläne der Drakonianer und ihrer Verbündeten zu zerschlagen. Doch noch immer streifen Horden von Drakonianern durch das Land und verbreiten Angst und Schrecken.

und niedergemetzelt. Es gelang Caramon jedoch, in Erfahrung zu bringen, daß im Tempel der Stadt Dracheneier lagern, die kurz vor dem Auschlüpfen stehen. Sollte es den bösen Kräften gelingen, die Dracheneier auszubrüten, könnte allein dadurch der Krieg zwischen Menschen und Drakonianern wieder aufflammen. Also müssen die Helden in den abgeriegelten

Tempelbereich eindringen und dort die Dracheneier finden und vernichten. Der Schlüssel, der den Weg ins Tempelviertel öffnet, wird einem bösen Priester und seiner Zombie-Armee abgerungen, worauf der Weg frei ist. Im Tempel stellen die Helden jedoch mit Entsetzen fest, daß die Dracheneier in Sicherheit gebracht wurden; nun lagern sie tief unter der Stadt im ver-



Das gute Kampfsystem läßt auch strategisches Vorgehen zu

wirrenden Labyrinth der Katakomben...

Diese und noch viele weitere Abenteuer erwarten den Spieler, der sich an die Erkundung der Welt Krynn macht. Auf einer großen Übersichtskarte werden die Helden in Städte und Burgen gelenkt. Kommt es zu Konfrontationen mit feindlichen Truppen, ändert sich die Bildschirmdarstellung. Von der großen Übersichtskarte wird auf eine vergrößerte Wiedergabe des Schlachtfeldes umgeschaltet. Hier findet der Spieler jede seiner Figuren und jeden Gegner nebst umgebenden Landmarken als Grafik. Die Helden werden jeder für sich gesteuert, so daß jede Figur einer eigenen Taktik folgen kann.

»Champions of Krynn« begeistert durch ein wohldurchdachtes Spielkonzept und ein ausgedehntes interessantes Spielareal. Ärgerlich ist mal wieder, daß nur das Programm für den Amiga konvertiert wurde, nicht aber die Grafik. Hier muß sich der Spieler mit Bildern begnügen, die zwar gut sind, aber den Amiga bei weitem nicht ausreizen. Ebenso verhält es sich mit Sound und Musik, hier muß man sich mit einigen kleinen Effekten und einer mäßig komponierten Titelmelodie zufriedengeben.

(hs)

AMIGA DOS Blitzlicht

Name: Champions of Krynn
Hersteller: S.S.I.
Vertrieb: Fachhandel
Preis: voraussichtlich 90 DM

Positiv:

- viel Spiel fürs Geld
- gutes Spielkonzept
- viele Teilabenteuer
- tolle Atmosphäre

Negativ:

- Sound nicht überzeugend
- Grafik nicht Amiga-Qualität

AMIGA DOS

URTEIL

GRAFIK SOUND MOTIVATION

7 3 9



Beim Catchen ist (fast) alles erlaubt...

International Championship Wrestling

Wenn im TV Muskelberge im Ring Tango tanzen und dazu grölen wie dereinst Tarzan im Busch, ist unter Garantie Catchen und eine Mordsgaudi angesagt. Ob das auch auf dieses Catchspiel zutrifft, werden wir sehen...

Drei Kämpfer treten nach dem K.-o.-System in Vorrundenkämpfen gegeneinander an, deren Sieger schließlich dem amtierenden Weltmeister gegenübersteht, um ihm den Titel streitig zu machen. Die drei Fighter sind frei wählbar, die Spielstärke des Computergegners ist ebenfalls von leicht über normal bis schwierig einstellbar.

Irrsinnig lange Ladezeiten und ein Sound, der eines Amiga absolut unwürdig ist, sind die ersten Eindrücke, die man von dieser Catch-Simulation erhält. Derlei Unbill wäre noch zu ertragen, würde sich das Spiel selbst als Juwel entpuppen. Doch weit gefehlt. Die Steuerung ist langsam, ungenau und umständlich, die Animation stereotyp, die "realistische Geräuschkulisse" erschöpft sich nun in "Uh-uh" und "Ah-ah", die Highscore-Liste ist nicht abzuspeichern, und das Gameplay selbst ist so aufregend wie lauwarmer Zwiebackbrei. Da hilft auch keine Zwei-Spieler-Option mehr, um Schwung in ein Programm zu

bringen, bei dem die Wartezeiten um ein Vielfaches länger sind als das eigentliche Spiel. (A. Hink/hs)

AMIGA DOS Blitzlicht

Name: International Championship Wrestling
Hersteller: Hewson
Quelle: Fachhandel
Preis: 84,95 DM

Positiv:

- Zwei-Spieler-Option
- auf Harddisk installierbar

Negativ:

- miserabler Sound
- lange Ladezeiten
- langweiliges Gameplay



Wer wird neuer Weltmeister in der Formel 1?

Grand Prix Circuit

Quietschende Reifen, das Donnern der schweren Boliden und der knochenharte Kampf um Punkte: In dieser Simulation für Formel-1-Rennen ist man hinter einem Sportlenkrad hautnah dabei!

Das Programm ist nach bewährter Accolade-Manier aufgebaut. Im Eingangsmenü kann man sich für Trainings-, Einzel- oder Meisterschaftsrennen entscheiden, fünf Schwierigkeitsgrade stehen zur Auswahl, und auch die Zahl der zu fahrenden Runden ist frei einstellbar. Acht Rennstrecken stehen zur Verfügung, über die man einen der drei Supercars jagen kann: einen Ferrari, einen Williams oder einen McLaren. Alle drei weisen recht unterschiedliches Fahrverhalten auf und sollten den Fahrkünsten des Spielers entsprechend ausgewählt werden.

Die Strecken selbst sind Repliken der berühmtesten Pisten der Welt. Fährt man einzelne Rennen, sind die Kurse frei anwählbar.

Im untersten Schwierigkeitsgrad muß man weder schalten, noch kann der Wagen bei Querfeldeinfahrten beschädigt werden. Das ändert sich bei den höheren Stufen, wo sowohl die anderen Fahrer ziemlich rücksichtslos zur Sache gehen als auch der Motor bei Schaltfehlern überdreht und in einer Qualmwolke den Geist aufgibt.

Die Hintergrundgrafiken und das Cockpit sind besser gestaltet als bei vergleichbaren

Programmen, und auch der Sound ist durchaus akzeptabel, wenn auch nicht sonderlich originell. Der einzige Schwachpunkt dieser Simulation liegt in der Steuerung, die extremes Fingerspitzengefühl erfordert. Hat man den Dreh aber erst einmal heraus, bekommt Alain Prost heiße Konkurrenz!

(A. Hink/hs)

AMIGA DOS Blitzlicht

Name: Grand Prix Circuit
Hersteller: Accolade
Quelle: Fachhandel
Preis: 84,95 DM

Positiv:

- spannender Aufbau

Negativ:

- Steuerung zu sensibel



Ein Autorennen besonderer Art ist »Test Drive«. Hier geht es nicht um WM-Punkte, sondern darum, einen von fünf Sportwagen als Testfahrer über ganz normale Landstraßen zu scheuchen. Absolviert man alle fünf Strecken in guter Zeit, winkt als Belohnung der Besitz des Flitzers. Die fünf Strecken zeichnen sich nicht gerade durch große Abwechslung aus, lediglich der Verkehr nimmt zu. Desgleichen allerdings auch die Fahrkunst der Cops, die mit ihrem Radar überall lauern. Hat man die eingebaute Warnanlage ignoriert und wurde erwischt, bekommt man ein Knöllchen verpaßt, was zwar kein Geld, aber eine Menge Zeit kostet. Statt anzuhalten, kann man allerdings auch höchst regelwidrig auf die linke Fahrbahn rüberziehen und die grünen Jungs solange am Überholen hindern, bis sie entnervt aufgeben.

Das Cockpit der Wagen (von Lamborghini über Porsche bis Ferrari ist alles dabei, was einen heißen Reifen fährt) ist detailliert gepixelt, die Hintergrundgrafiken dagegen weniger (bis auf die zermatschten Fliegen auf der Scheibe). Auch der Sound mit quietschenden Reifen und Motorengedröhn ist eher Mittelklasse. Da die Steuerung aber Spitze ist und die Highscores der ewigen Bestenliste in immer neue Höhen schraubt, ist für anhaltenden Spielspaß dennoch gesorgt.

Golfsimulationen gibt es wie Sand am Meer, doch dieses Programm ist, wenn es auch schon einige Jährchen auf dem Buckel hat, immer noch der Spitzenklasse zuzuordnen.

»Mean 18« besticht durch seinen schnellen Bildschirmaufbau und die abwechslungsreiche Grafik. Die sieben Kurse, die mit ihren zum Teil echt gemeinen Löchern dem Titel alle Ehre machen, sind so unterschiedlich gestaltet, wie sie auch in Wirklichkeit aussehen. Ist man trotzdem nicht zufrieden, können mit dem mitgelieferten Kurs-Designer eigene Ideen zur Genüge verwirklicht werden.

Übungsmöglichkeiten stehen genauso zur Verfügung wie eine ewige Highscore-Liste. Die Animation ist flüssig, das Publikum geht bei einem Super- oder Beinahe-Superschlag voll mit. Einziger Schwachpunkt ist die an sich



Mit quietschenden Reifen um die Kurven...

All Time Favourites

Daß auch ältere Programme ihre Daseinsberechtigung behalten können, beweist diese Zusammenstellung von immer noch aktuellen Accolade-Hits in eindrucksvoller Weise. Denn hier wurden wieder mehrere ältere Titel zu einer »Spielesammlung« zusammengestellt.

recht exakte Steuerung, mit der das »Putten« auf dem Grün leider ziemlich zum Glücksspiel wird. Wie im echten Leben eben...

Mean 18 zeigt auch auf eindrucksvolle Weise, daß Computerspielen nicht zur Vereinsamung führen muß. In Gesellschaft einiger Mitspieler eine Runde über den Golfplatz zu ziehen und dabei zu versuchen unter Par zu bleiben, hat schon seine Reize.

Wer noch keine Golfsimulation besitzt, dem ist mit diesem Programm als Einstieg jedenfalls bestens gedient.

»Hardball« - das sind achtzehn Spieler, ein rautenförmiges Spielfeld und ein kleiner, sehr harter Ball, der dieser Abart des Baseball den Namen gegeben hat.

Das Programm ist eine Mischung aus Strategie und Action. Zum einen muß die Aufstellung der neun Spieler pro

Team den wechselnden Gegebenheiten angepaßt werden, was während des Spiels jederzeit geschehen kann, zum anderen erfordern das richtige Werfen, Schlagen und Fangen gutes Augenmaß und schnelle Reflexe. Durch die gut durchdachte, schnelle Steuerung, die mit Maus oder Joystick problemlos zu bedienen ist, bleibt man jedoch immer Herr der Situation.

Die Geräuschkulisse entspricht voll der eines gut besuchten Stadions. Der Knall, wenn der Schläger des Batters den Ball trifft, ist ebenso hörenschrift wie der Sound, wenn ein Läufer mit vollem Schwung in eine Base hineinfliegt. Auch die Grafik ist, besonders bei Nahaufnahmen, sehr schön detailliert und flüssig animiert. Lediglich bei Screens, die das ganze Spielfeld zeigen, wird sie etwas grober und ruckeliger.

Hardball ist das beste und gleichzeitig auch schwächste Programm dieser Sammlung. Das Beste, weil es unheimlich spannend ist, und das Schwächste, weil es nur dann spannend ist, wenn man die Regeln dieser amerikanischen Sportarten in- und auswendig kennt. Und da die Baseballregeln in Europa nicht unbedingt zum Allgemeinwissen gehören, wird dieses Programm wohl kaum die Fangemeinde finden, die es ganz ohne Zweifel verdient.

(A. Hink/hs)



Der Wurf muß genau über das kleine Quadrat vor dem Fänger gehen, um gültig zu sein

AMIGA DOS Blitzlicht

Name: All Time Favourites
Hersteller: Accolade
Quelle: Fachhandel
Preis: 99,95 DM

Positiv:

- Programme unterschiedlichen Charakters

Negativ:

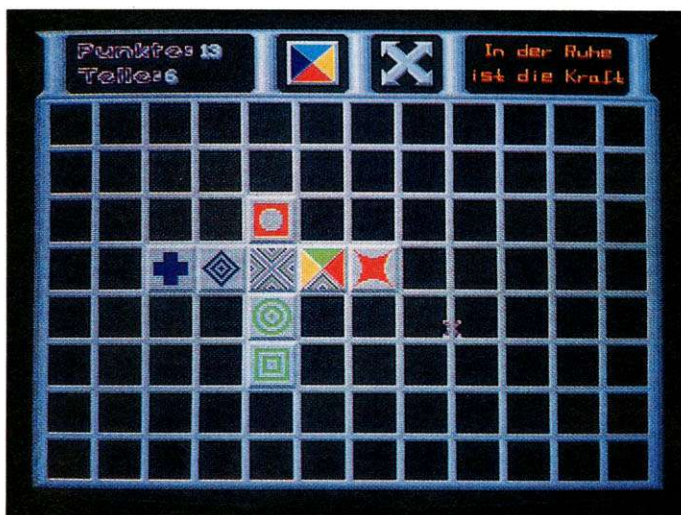
- Preis nicht dem Alter der Programme entsprechend

AMIGA DOS

URTEIL

GRAFIK SOUND MOTIVATION

5 5 6



Wohin, wohin mit der Formen Sinn...

Titano

Was haben ein grüner Kreis und mehrere rote Rechtecke miteinander gemeinsam? Wenn Sie das herausfinden, haben Sie schon die erste Hälfte der Anleitung von »Titano« verstanden.

Für Abende, an denen das Fernsehen nur Schonkost liefert, Freunde nicht auffindbar sind und nur das Lieblingsgetränk im Kühlschrank auf Gesellschaft wartet, lohnen sich Geduldsspiele oder »Knobeleyen«, wie man im Volksmund sagt, ganz besonders. »Titano« heißt ein etwas ausgefallenes Spiel, welches weder bekannte Spielarten kopiert, noch irgendeine Verwandtschaft mit anderen Vertretern seiner Art erkennen läßt. Es gilt, insgesamt 72 Spielsteine auf ein Spielfeld mit 96 Kästchen zu verteilen. Hört sich gut an, nicht? Doch »Titano« wäre nicht »Titano«, wenn da nicht noch der bekannte Haken wäre, denn an einen Stein mit roter Farbe paßt links, rechts, oben und unten nur einer mit roter Farbe – es sein denn, er ist blau, lila, gelb oder grün und zeigt das gleiche Symbol. Manche Steine erlauben eine beliebige Farbe, manche entfernen Steine, die schon gesetzt waren, wiederum andere lassen die Punktzahl schrumpfen oder steigen.

»Titano« ist nichts Weltbewegendes: Recht einfach aufgemacht, weder mit großartiger Grafik noch mit opulenten Sounds versehen, bezieht dieses Spiel seine Reize aus dem Spielgeschehen und dem im-

mer kniffliger werdenen Problemen. Sicher hätten gute Grafik und feine Musik das Produkt noch viel attraktiver gemacht, aber was letztendlich zählt, ist der Spielspaß, und davon ist bei »Titano« reichlich vorhanden.

(jb)

AMIGA DOS Blitzlicht

Name: Titano
Hersteller: Software 2000
Vertrieb: Fachhandel
Preis: 69,95 DM

Positiv:

- schönes Knobelenspiel für lange Abende
- leicht zu verstehen, aber nicht leicht zu spielen

Negativ

- nur eine Spielvariante
- keine Festplatteninstallation



Softwarenoten und wie sie zustandekommen

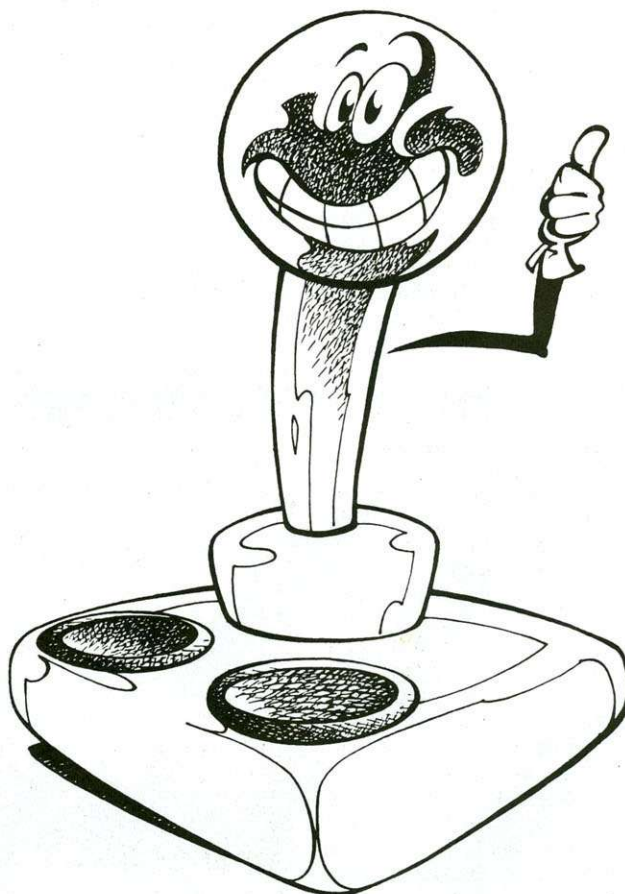
Daß Ihre AMIGA DOS bei den Spiele-Tests ein Urteil vergibt, haben Sie selbst sicherlich schon gesehen. Wie dieses Urteil funktioniert, das soll hier noch einmal im Detail erklärt werden.

AMIGA DOS beurteilt jedes Spiel anhand verschiedener Kriterien: Die Grafik, die Musik und den Spielspaß. Die Noten dafür werden in der kleinen Urteilsbox, die am Ende jeder Review zu finden ist, unter den Oberbegriffen Grafik, Sound und Motivation dargestellt. Die Note selbst richtet sich nach der Höhe des wiedergegebenen Wertes. Je höher, desto besser. Zu diesen Noten gesellt sich ferner ein individuelles Urteil des Testers. Eine vierte Note also, die allerdings nicht wiedergegeben wird.

Aus diesen vier Werten bildet sich schließlich das Gesamturteil:

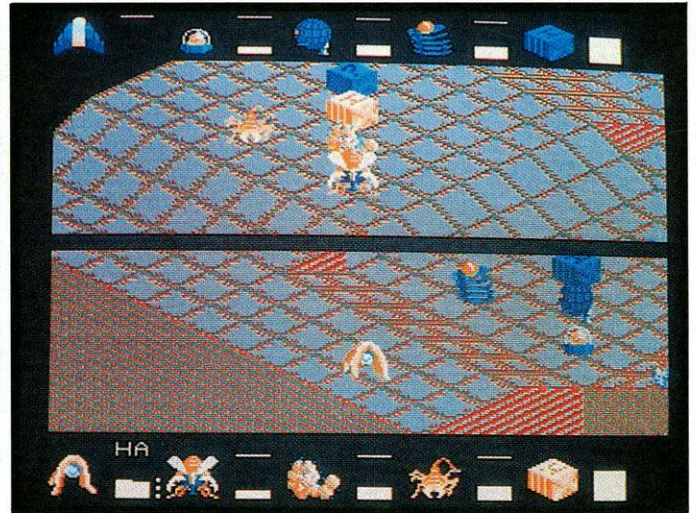
- Sehr gut
- gut
- o.k.
- geht so
- schlecht

Die Bedeutung dieser Bewertungen erklärt sich von selbst. Uns bleibt noch, Ihnen viel Spaß mit dem AMIGA DOS Spieleteil zu wünschen.





»Alle Hunde kommen in den Himmel« auch auf dem Amiga



»Domination« – Einladung zum Roboterduell

All Dogs go to Heaven

Zeichentrickfilme zu Computerspielen umzubauen, hat auf dem Amiga schon so etwas wie eine Tradition. Auch dieses Spiel wurde wieder unter Benutzung von Originalzeichnungen aus dem Film erstellt und lehnt sich in seiner Spielhandlung deutlich daran an. Bleibt nur zu hoffen, daß der Spieler hier nicht nur opulente Grafik zu sehen bekommt, sondern auch ein vernünftiges Spielgeschehen.

Domination

Dieses Spiel versetzt Sie in eine Kampfarena, in der verschiedene Robotertypen gegeneinander antreten. Zunächst jedoch gilt es, die besten Ausrüstungsgegenstände zu erringen, um damit die eigenen Robbies besser auszustatten als die des Gegners.

Demnächst auf Ihrem Computer

Das Marketing feiert wieder einmal fröhliche Urstände: "Alle Hunde kommen in den Himmel" hat es bis zum Computerspiel gebracht. Aber auch sonst darf man wieder gespannt darauf sein, was der Software-Markt in Zukunft bereithält.

Gespielt wird auf einem Spielfeld, das spontan an Marble Madness erinnert. Der gesplittete Bildschirm macht sich besonders bei zwei Spielern positiv bemerkbar.

Gates of Dawn

Rollenspieler hergehört: Wer die Nase von bierernsten Fantasy-Szenarien voll hat, wird sich sicherlich darüber freuen,

daß mit »Gates of Dawn« ein humoriges Spiel des Genres auf uns zukommt. In der Tradition von Münchhausen und Alice im Wunderland angesiedelt, darf der Spieler die Machenschaften eines fieschen Zauberers aufdecken.

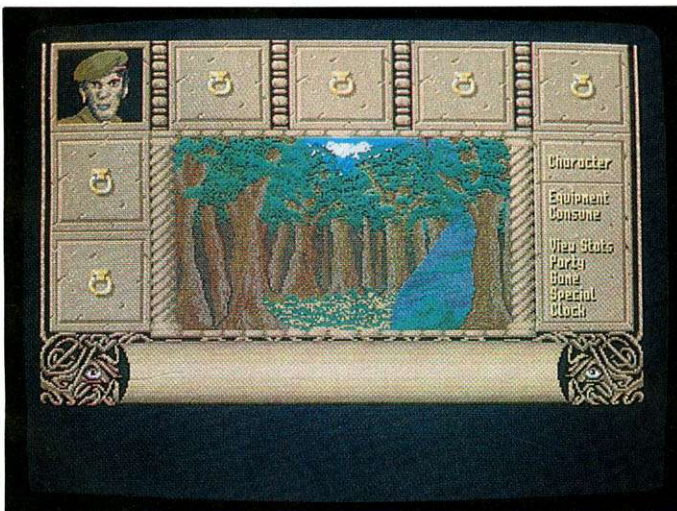
Dazu muß ein grafisch opulent gestaltetes Spielareal ausgekundschaftet werden. Städte, Urwälder und Verliese sind ausgefüllt mit vielen phanta-

sievoll animierten Spielcharakteren und Monstern. Auch wenn »Gates of Dawn« auf den ersten Blick an »Bards Tale« erinnert, merkt man schon bald, daß hier ein weit ausgeklügelteres Grafiksystem dem Spieler besseren Einblick in das Szenario gewährt.

Dynasty Wars

Das mittelalterliche China: Die Han-Dynastie wird von blutigen Aufständen erschüttert. Rebellierende Kriegsherren tauchen das Land in ein Blutbad. Nur wenn es gelingt, die Rebellenarmeen zu schlagen, kann der rechtmäßige Herrscher den Thron besteigen und dem Lande den Frieden wiederbringen.

Machen Sie sich auf ein Computerspiel gefaßt, das den wüstesten chinesischen Histo-rienschinken im Säbel-schwingen um nichts nachsteht. (hs)



»Gates of Dawn« – ein neues Rollenspiel



Dynasty Wars – Duell der Kriegsherren

Impressum

Herausgeber

Christian Widuch

Redaktionsleitung

Stefan Ritter

Chefredaktion

Markus Matejka (mm)

Leitender Redakteur

Jürgen Borngießer (jb)

Redaktion

Heinrich Stiller (hs), Claus Daschner (cd),
Vera Brinkmann (vb), Thomas Baum (tb)

Freie Autoren dieser Ausgabe

Bernd Rudolf, Robert Marz, Jürgen Seibel,
Michael Anton, Andreas Polk, Peter Woop,
Franz-Josef Reichert, Antje Hink, Oliver Wagner,
Peter Engels, Wolf-Jürgen Faust, Edgar Meyzies,
Guido Wimmel, Andreas Vorgeitz, Robert Münch,
Ralf Zuber, Ottmar Röhrig, Oliver Heggelbacher

Redaktionsassistenten

Anke Kerstan (ke), Susanne Eska (es)

Chef vom Dienst

Matthias Bloß

Produktionsleitung

Gerd Köberich

Bereichsleitung

Claudia Ebbrecht (Fotosatz/Lektorat),
Margarete Schenk, Helmut Skoupy
(Montage/Reprografie)

Layout

Michael Grebenstein

Fotografie

Christian Heckmann

Fotosatz

Marcus Geppert

Lektorat

Heike Wiegand

Montage/Reprografie

Manuela Eska, Peter Gajewski

Werbeplanung

Mohamed Hava

Anzeigenleitung

Wolfgang Brill

Anzeigenverkauf

DMV-Verlagsbüro München
Zaunkönigweg 2c, 8000 München 82
Telefon (089) 439 1087, Telefax 089/439 1080

Leitung: Britta Fiebig

Anzeigenverkauf: Monika Schöbel, Peter Schätzle,
Hannelore Schulzki, Michael Hofmann

Anzeigenverwaltung und Disposition

Andrea Giese, Karina Ehrlich, Beate Kranz,
Christina Füllgraf

Anzeigenpreise

Es gilt die Anzeigenpreisliste Nr. 1 vom 01.05.1990

Anschrift Verlag/Redaktion:

DMV Daten & Medien-Verlag
Widuch GmbH & Co. KG
Fuldaer Straße 6
3440 Eschwege, Telefon (0 56 51) 809-0,
Telefax (0 56 51) 809-333

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 Datenträ-
ger 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 gehen nicht
in jedem Fall die Meinung der Redaktion wieder.

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, Cheat-modes 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 sind es nur ein Wort oder ein kleiner Hinweis, die dem Spielgeschehen neues Leben einhauchen.

Aus diesen Gründen möchten wir Sie um Ihre Mitarbeit bitten. Wenn Sie also über Speiletips – gleich welcher Art – verfügen, würden wir uns freuen, von Ihnen Post zu bekommen.

Und hier nun die Adresse, an die Sie Ihre Speiletips schicken können:

**DMV-Verlag
Redaktion AMIGA DOS
Stichwort: Speiletips
Postfach 250
3440 Eschwege**

PS: Unter allen Einsendungen verlosen wir jeden Monat fünf Software-Titel für die besten Speiletips.

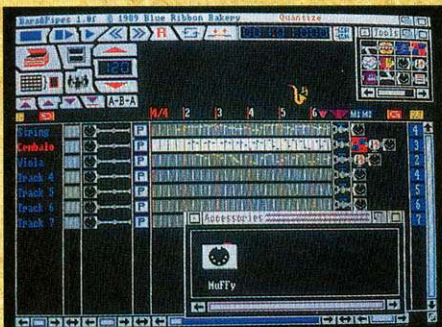
Die Inserenten

Alpha 2000.....	69
Alsdorfer PD Center.....	86
A.P.S. Elektronik.....	85
Auriga.....	85
B + S Computer.....	85
Bonito.....	73
CIK-Computertechnik.....	87
CompiMate.....	91
Computing.....	91
CSV Riegert.....	17
DELTA SOFT.....	87
Diezemann.....	23
DMV.....	18,36,37,131,132
Dombrowski.....	85
Donau-Soft.....	86,93
fhn Computer.....	86
FSE Strauß-Elektronik.....	27
German-Soft.....	86
GFA-Systemtechnik.....	11
GNE.....	86
HK-Computer.....	2
IC-InterComputing.....	43
Kirschbaum.....	69
Kramer.....	85
Kunze.....	85
Mac-Soft.....	86
Mailsoft.....	85
MVC.....	73
Neuroth Hardware-Design.....	85
Nürnberger PD-Studio.....	45
Omega Datentechnik.....	85
Pawlowski.....	86
PUBLIC DOMAIN CENTER.....	86
Pulsar.....	23
P.V. Computershop.....	86
G. v. Thienen.....	45
WENNGATZ.....	93

Im nächsten Heft

■ ■ ■ Grafik, Animation, Raytracing

Worte, die dem eingefleischten Amiga-Besitzer angenehme Schauer über den Rücken laufen lassen. Um auch den Anfängern die Gelegenheit zu geben, haben wir uns umgesehen, welche Möglichkeiten es zur Grafikentwicklung gibt und was beachtet werden sollte. Eine kleine Marktübersicht rundet das Bild ab und informiert Sie über die besten Programme.



Bars&Pipes ist das Allerneueste auf dem Amiga-MIDI-Musiksektor

■ ■ ■ Schnell, schneller, am schnellsten

Der Amiga gehört mit seinem 68000-Prozessor nicht zu den langsamsten Computern. Doch bei einigen Anwendungen wünscht man sich mehr Power, was die Rechengeschwindigkeit angeht. Wir haben uns deshalb mehrere 60030-Prozessor-Karten vorgenommen. Wer das Rennen gewonnen hat, können Sie in der nächsten Ausgabe erfahren.

■ ■ ■ PC-DOS, einfach so?

Daß beim Amiga 2000 eine PC-Karte eingesteckt werden kann, dürfte hinlänglich bekannt sein. Doch dem Amiga-500-Benutzer wurde nichts in dieser Richtung geboten. Bis jetzt. Wir haben für Sie in der nächsten Ausgabe ein Software-Programm

und ein Einsteckmodul getestet, mit dem sich PC-Software "fahren" lassen soll. Wie das geht und was es kostet, erfahren Sie in der neuen AMIGA DOS.

■ ■ ■ Assembler nur für den absoluten Profi?

"Lernen, verstehen und anwenden" heißt die Devise in der nächsten AMIGA-DOS-Ausgabe. Dort geht es um die Programmierung in Assembler, die maschinennahe Sprache.

■ ■ ■ Auf was Sie noch gespannt sein können

Empfehlenswerte Tips zu: Modula2, CLI, GFA-BASIC und C, viele Software-Tests, nützliche Hardware-Erweiterungen, anregende Workshops und viele weitere knallige Informationen, die einfach jeder Amiga-Besitzer wissen sollte. Und für die Spiele-Fans unter uns die allerneuesten HyperGamma-Spiele.



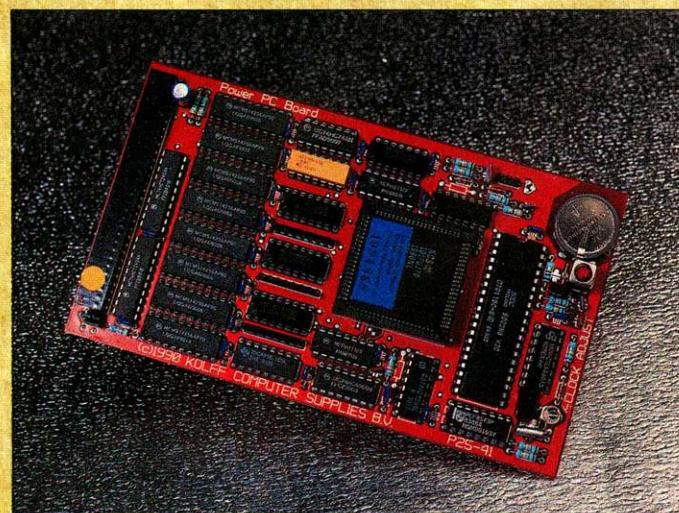
Der Power-Farbscanner SHARP JX-100. Leistet er das, was er verspricht?



finden Sie ab

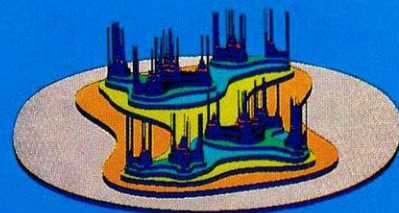
12. September '90

bei Ihrem Zeitschriftenhändler



Der PC im Amiga. Was wird dem Anwender für sein Geld geboten?

Amiga Fraktal Generator 3D



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 Farbrequerter mit stufenloser Schieberegung

Mehrere separate Bildspeicher

Bis zu vier Bilder gleichzeitig abrufbar
Separate Farbzuordnung und Animationsmöglichkeit

Farb-Animationen

Phantastische Effekte durch Amiga-Color-Cycling

Amiga Fraktal Generator 3D

3 1/2"-Disk. Best.-Nr. 2901

69,- DM (unverbindliche Preisempfehlung)

Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:

Inland:

Einzelpreis 69,- DM
zzgl. Versandkosten 4,- DM
Endpreis 73,- DM

Ausland:

Einzelpreis 69,- DM
zzgl. Versandkosten 6,- DM
Endpreis 75,- DM

– Bitte benutzen Sie die Bestellkarte im Heft. –

DMV-Verlag · Postfach 250 · 3440 Eschwege



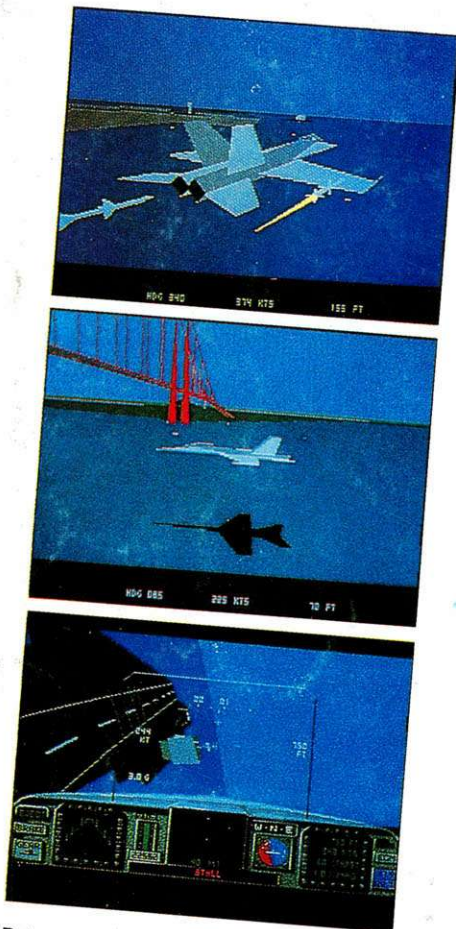
AMIGA DOS

Spielbox

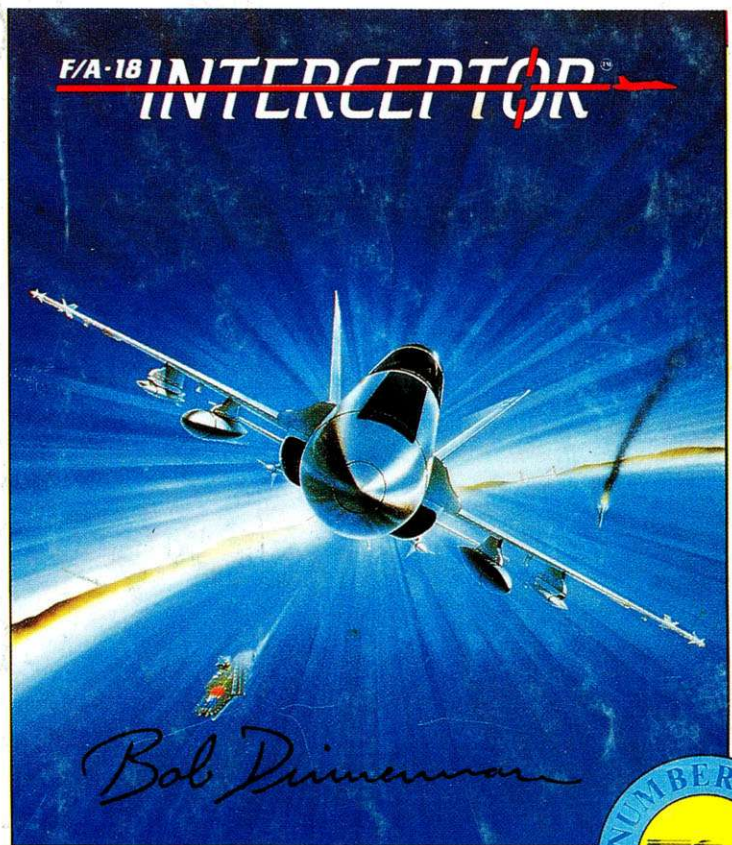
Solange der Vorrat reicht

Fliegen Sie das faszinierendste Flugzeug unserer Zeit!

Fliegen Sie entweder mit der F16 oder F18, und starten Sie von einem der drei Landstützpunkte oder vom Flugzeugträger aus • Realitätsnahe 3D-Grafik der San-Francisco-Bucht, komplett mit Golden Gate Bridge, Alcatraz und der Pyramide in San Francisco • Holografisches HUD-Display und realistische Sound-Effekte • 8 unterschiedliche Ansichten aus dem Cockpit sowie 8 unterschiedliche Außenansichten • 7 Missionen mit unterschiedlichen Schwierigkeitsgraden.



Robert Dinnerman entdeckte seine Leidenschaft für Computerspiele, als er Hardware für Motorola entwickelte. Sein Interesse führte ihn zu der Coin-Op-Abteilung von Bally, wo er an 3D-Perspektiven- und Flugsimulationen arbeitete. Nachdem er einen Evans-Sutherland-Flugsimulator gesehen hatte, beschloß Bob, ein vergleichbares, aber erschwingliches Programm für Home-Computer zu entwickeln. Wenn Sie seinen F/A 18 Interceptor spielen, werden Sie sehen, daß er in der Tat etwas Außergewöhnliches vollbracht hat.



Unser Sonderangebot an alle Amiga-Leser.
Nur solange der Vorrat reicht.

F/A 18 Interceptor, Electronic Arts,
Amiga 3,5-Zoll-Diskette

DM 39,95 *

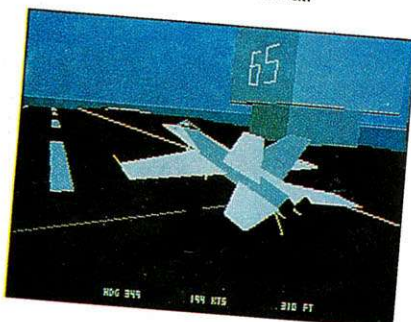
und aus unserem laufenden Programm

DMV-Fraktalgenerator 3D Amiga
Amiga 3,5-Zoll-Diskette

DM 69,- *

* Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland DM 4,- bzw. für das Ausland DM 6,- Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 2 50 · 3440 Eschwege



ORGATEC
INTERNATIONALE
BÜROMESSE KÖLN
25.-30. OKT. 1990

Wir stellen aus
HALLE 2.1
GANG F, STAND 42

DMV
Daten- und
Medienverlag