

AMIGA DOS

11/90ISSN 0937-2717
DMV-Verlag

Im Test:

Transputerkarte
Mega Midget
Becker Text 2.0
3D Professional



Sprachen:

Profis informieren

DER HAMMER!

Preise im Wert von

50.000, –
zu gewinnen!!!

TOP SECRET:

Alle Neuheiten auf
der AMIGA '90

AMIGA PUBLIC DOMAIN

über 9000 Disketten im Bestand!

MODULES-DISK
über 80 randvolle
Disk mit abspiel-
baren Modulen!!!
Stück 4, - (3,5")

KATALOG-SET
5 deutsche Disk
und die neueste
TIME > 20, - (VK)

Info-Diskette für DM 2, - (VK)

SAMPLES-DISK
28 volle Disk mit
ausgewählten
Samples!!!
Stück 5, - (3,5")

A.P.S.-electronic

Sonnenborstel 31, D-3071 Steimbke
Tel. 0 50 26/17 00 FAX 0 50 26/16 15

POWERPACKER professionell

Der Supercruncher mit diversen
Möglichkeiten. Mit ausführlicher
deutscher Anleitung!

DM 39, -

INTROMAKER professionell

Erstellen Sie Ihre eigenen Intros!
Einbinden eigener Animationen,
Bilder, Musik, - versch. Fonts!!!

DM 29, -

Zwei, die aus dem

Besuchen Sie
uns auf der
AMIGA 90
in Köln



Rahmen fallen!

Halle 12/2
Stand 502
9. - 11.11.90

PD-Katalog
3 deutsche Disk
und neueste
Anti-Virendisk
DM 15, - (VK)

Herrmanns & Kommelter

Vom-Bruck-Platz 45, D-4150 Krefeld
Tel. 0 21 51/39 98 33 FAX 0 21 51/39 95 69

INFO-DISK
aktuelle News,
Musik, Preisaus-
schreiben, pp.
DM 2, - (VK)

SHARE: Die deutsche Serie, bei der die Programmierer an
jeder Diskette finanziell beteiligt sind!
Auf der Messe nur bei uns zu bekommen!

Auch alle anderen gängigen PD-Serien erhalten
Sie zum fairen Preis bei uns!

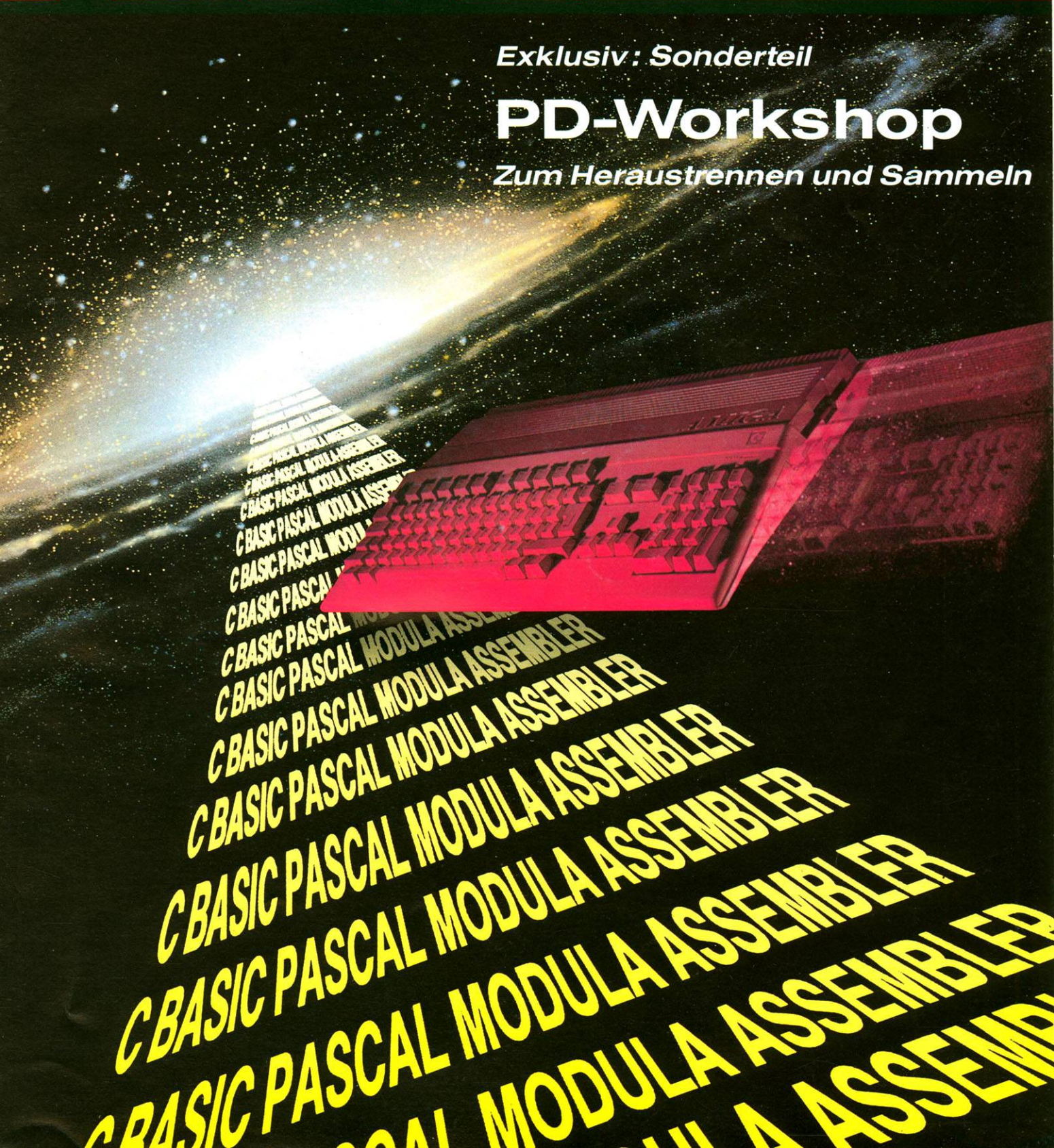
DAS MAGAZIN FÜR AKTIVE AMIGA-ANWENDER

AMIGA DOS

Exklusiv: Sonderteil

PD-Workshop

Zum Heraustrennen und Sammeln





Autorisierter Commodore-Fachhändler
Commodore Commercial Developer

- Entwicklung
- Produktion
- Hardware
- Software
- Service

HK-Computer

Ihr Amiga-Spezialist



*** HK-Professional ***
Top in Qualität, Funktion und Design

Professional Drive Diskettenlaufwerke

- 3 1/2" Laufwerk AMIGA 2000 intern DM 159,—
- komplett mit Einbaueinheit und Anleitung
- 3 1/2" Laufwerk für alle AMIGAs extern DM 199,—
- abschaltbar, Busdurchführung, AMIGAFarben
- 5 1/4" Laufwerk für alle AMIGAs extern DM 249,—
- abschaltbar, Busdurchführung, 40/80 Tracks, AMIGAFarben

Professional RAM-Board IIC A500

- auf 1 MB DM 149,—
- jetzt noch leistungsfähiger!
- superschnelle Megabit-RAMs (4*514256)
- mit accupufferter Uhr & Datum
- Writeprotect für die Uhr
- Accu abschaltbar
- Hard- und Softwaremäßig abschaltbar
- Superniedriger Stromverbrauch
- dto. Platine mit Uhr & Schalter ohne RAMs DM 89,—

Professional RAM-Board III A500

- auf 2,3 MB DM 498,—
- 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 IIIB A500

- auf 2,5 MB DM 498,—
- 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

- 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 1098,—
- dto. Platine bestückt mit 8 MByte DM 1698,—
- dto. Platine teilbestückt ohne RAMs DM 498,—
- RAM-Satz für 2 Megabyte DM 300,—

Professional MIDI für alle Amigas

- DM 139,—
- Superschnell
- Optokoppler mit 700 %! Kopplungsfaktor
- 1" In, 1" Thru, 3" Out
- Leitungstreiber an allen Ausgängen für lange Datenkabel
- Amigafarbenes Metallgehäuse
- abschaltbar, mit Betriebs-LED

Professional SCSI II

- Harddisk-Controller DM 498,—
- A2000 Slotkarte als Filecard-Träger ausgelegt
- voller 16bit-Datenbus
- vergoldete Kontakte
- durchgeführter SCSI-Bus für bis zu 8 SCSI-Devices
- Autoboot
- abschaltbar
- Komplette Filecards in verschiedenen Größen auf Anfrage

Disketten

- | | 10 Stück | 100 Stück |
|---------------------------|----------|-----------|
| 3 1/2" NoName 2 DD | DM 12,90 | DM 119,— |
| 3 1/2" Verbatim Verex 2DD | DM 18,90 | DM 179,— |
| 5 1/4" NoName 2S2D | DM 5,90 | DM 57,— |
| 5 1/4" Verbatim Verex | DM 9,90 | DM 95,— |
- Größere Staffeln auf Anfrage

— NEU — NEU — NEU — NEU — NEU — NEU —

Maus & Joystick-Adapter

DM 49,—

- jetzt AUTOMATISCH!, manuelles Umschalten überflüssig
- für gleichzeitigen Anschluß von Maus und Joystick
- Umschaltung erfolgt einfach durch Betätigen von Maus/Joystick
- alle Maussteuerleitungen elektronisch geschaltet, dadurch keine Spannungsspitzen

— NEU — NEU — NEU — NEU — NEU — NEU —

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,—

- der Highscore-Killer jetzt auch extern!

Kick-ROM

DM 49,—

- Kickstartumschaltplatine für zwei Original-ROMs

- Kick-ROM mit ROM DM 98,—
- OriginalROM 1.2 oder 1.3 DM 65,—

Kickstartumschaltplatine

3-fach DM 59,—

- für zwei OriginalROMs und eine Epromversion

- Umschaltplatine mit OriginalROM DM 108,—
- Brennserie incl. einem Epromsatz DM 79,—
- (gegen Einsendung einer OriginalDiskette)

► 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

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 Decoder mit FTZ (Drews)

DM 199,—

Original Amiga-Maus

DM 69,—

Reisware Maus für Amiga

DM 89,—

Staubschutzhauben

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

Die Staubschutzhauben sind aus Kunstleder mit weichem antistatischen Innenfutter

Software

- MEDUSA der Atari ST-Emulator DM 444,—
- RAM-Test Amiga DM 24,50
- zeigt defekte Speicherstellen grafisch an, 100 %
- Assembler jetzt auch für 32Bit-RAM, z.B. A2500/3000
- PACKIT DM 39,—
- Superschneller Cruncher nicht nur für Text und Grafik, sondern auch für Programme. Verschiedene Kompaktmodi, Auto- oder Loaderstart, räumt nicht nur Ihre Disketten, sondern auch Festplatten auf.

- XCOPY II DM 49,—
- XCOPY II mit Hardwarezusatz DM 69,—
- XCOPY professional DM 99,—
- Turboprint II DM 89,—
- Turboprint professional DM 188,—
- Quarterback (Festplatten-Backup) DM 119,—
- DPaint III DM 248,—
- Beckertext II DM 289,—
- GFA-Basic 3.5 DM 229,—

AMIGA Köln 90

Wir stellen aus
08.-11. November 1990
Halle 10, Stand 824

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 1798,—

Amiga 500 DM 848,—

Harddisk A590 20 MB für A500 DM 898,—

Colormonitor Commodore 1084 SP1 DM 598,—

Targa Multisync-Monitor TM 1480 DM 1098,—

— Lochmaske 0,28 mm, incl. A3000-Anschlußkabel

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-Hard- & Software ausgeliefert.

AUTOBOOTMODUL für A2000 DM 119,—

AUTOBOOTMODUL für A500 DM 149,—

AUTOBOOT-KARTE für A2090-Controller DM 119,—

TURBO-CHIPSET 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

Autoboot-Filecardträger

Harddisk-Trägerplatine mit DM 299,—

integriertem Autoboot-Modul,

Controlleradapter, Autoboot-Software

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

Haben Sie Hard- oder Software für den Amiga entwickelt?

Wir bieten Ihnen eine großzügige Umsatzprovision und eine ehrliche Abrechnung.

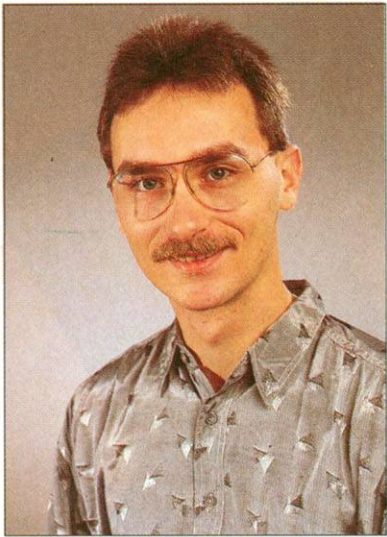
► Sprechen Sie uns an ◀

HK-Computer

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

Telefon: 0221/31 1606 · Telefax: 0221/32 1166 · BTX: * HK *
Geschäftszeiten: Mo.-Fr. 10.00 - 13.30 u. 14.30 - 18.30
Sa. 10.00 - 14.00 Uhr
Stadtparkasse Köln: BLZ 37050198, Kto. 634 2133

Nachnahme-Versand innerhalb Deutschlands per UPS oder Post DM 10,—. Nachnahme Ausland per Post DM 20,—.
Großgeräte nach Gewicht.
Fordern Sie unser kostenloses Info an!



Winterzeit — Messezeit

Der Siegeszug des Amiga ist nicht mehr aufzuhalten. Er ist auf dem besten Weg, sich neben dem Industriestandard PC als Kreativsystem zu etablieren.

Denn ein gut ausgestatteter Amiga ist sehr wohl in der Lage, Ergebnisse hervorzubringen, die sonst nur mit ungleich teureren Workstations erzielt werden können.

Und wurden bisher die brandheißen Neuheiten für den Amiga meist außerhalb von Deutschland präsentiert, so ist es um so erfreulicher, daß die größte und bedeutendste Amiga-Messe des Jahres 1990 in Köln stattfindet. Mit zirka 150 erwarteten Ausstellern ist sie schon jetzt die größte Messe zum Thema Amiga überhaupt.

Im Gegensatz zur letztjährigen Veranstaltung, die gelinde gesagt aus allen Nähten platzte, haben die Veranstalter sich auf den zu erwartenden Besucheransturm eingestellt. Einem ausgedehnten Messebesuch steht somit nichts mehr im Wege. Und es lohnt sich in jedem Fall!

Neben brandheißen Hard- und Software, die im Rahmen dieser Messe zu besichtigen und auch zu kaufen ist, möchten wir Ihre spezielle Aufmerksamkeit auch auf das Forum lenken, in dem Sie interessante Informationen zu vielen Themen rund um den Amiga erhalten.

Wir von der AMIGA DOS haben weder Kosten noch Mühen gescheut, um Ihnen etwas ganz Besonderes zu bieten. Unser Forum bietet an allen Messtagen Information, Unterhaltung, Interviews und Personalities. Ein Ereignis, daß Sie keinesfalls verpassen dürfen. Wir freuen uns heute schon darauf, Sie in unserem Bistro oder bei unseren Forumsveranstaltungen begrüßen zu können. Einzelheiten entnehmen Sie bitte unserer Messezeitung, die täglich neu kostenlos für Sie bereit liegt.

Im Bistro und auf unserem Messestand (Halle 12, Stand 307) haben Sie die Gelegenheit, die Redakteure der AMIGA DOS einmal persönlich kennenzulernen. Und mit etwas Glück können Sie sogar einen Amiga 3000 oder einen der zahlreichen anderen Preise gewinnen.

Neugierig? Dann schauen Sie doch einfach einmal bei uns vorbei!

Herzlichst Ihr

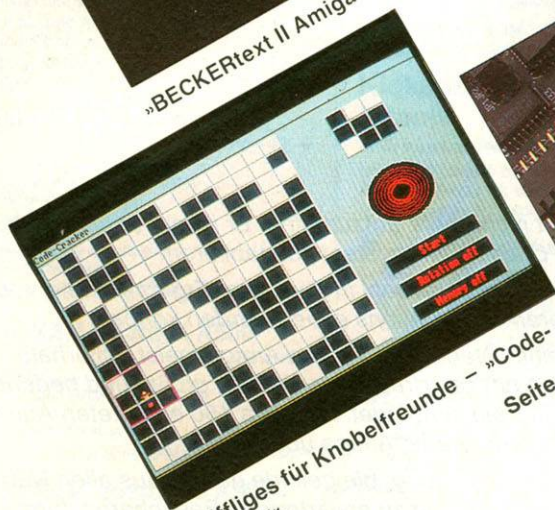
Markus Matejka

Markus Matejka

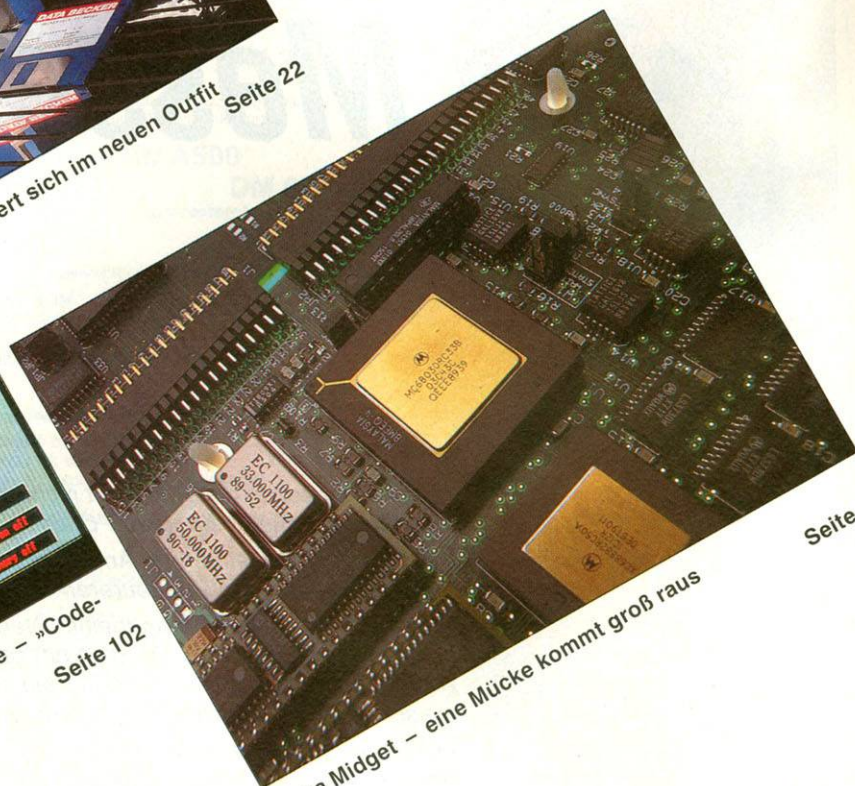
INHALT



„BECKERtext II Amiga“ präsentiert sich im neuen Outfit Seite 22



Kniffliges für Knobelfreunde – „Code-Cracker“ Seite 102



Mega Midget – eine Mücke kommt groß raus Seite 4

AMIGA DOS

AMIGA NEWS

Amiga '90 in Köln 8

HIGH TECH

Profi Amiga mit 16 Millionen Farben 16

SOFTWARE

Simply the Best? 20
BECKERtext II Amiga

The Megamighty Introdiesigner 24
Intros schnell erstellt

Goldene Werkzeuge 26
Utilities für den Programmierer

Ein neuer Standard? 28
Supergrafik mit 3D Professional

Charley's Tante auf dem Amiga 32
DTP leicht im Griff

HARDWARE

DIGI TOWER mit Power 36
Digitalisieren in Echtzeit

Streamen, was das Band hält 37
Daten sichern par excellence

Video-Digitizer VD 4 Amiga 40
Bildmaterial vom Feinsten

Der Mega-Racer 42
Mega Midget – ein rasantes Turbo-board

Kleiner Speicherriese 44
Massig Speicher auf kleinem Raum

TIPS & TRICKS

Stars and Bytes 69
Wie ein Sternenhimmel im Amiga entsteht

Gewußt wie 74
Tips für Einsteiger und Fortgeschrittene

Wie geht das in Assembler? 76
Knackig kurze Programmiertips

Joke-Handler 80
Diffiziles leicht gemacht

TITEL

Amiga, ich befehle dir... 46
Wie sehen Profis Ihre Sprachen?

Modular zum Erfolg 47

Besser geht's mit Oberon 48

Eine Lanze für BASIC 50

C – Eine Sprache nur für Könner? 52

Bekenntnisse eines Pascal-Programmierers 54
Inside Amiga 55

KURS

Das AmigaDos leicht und verständlich! 113
Multitasking

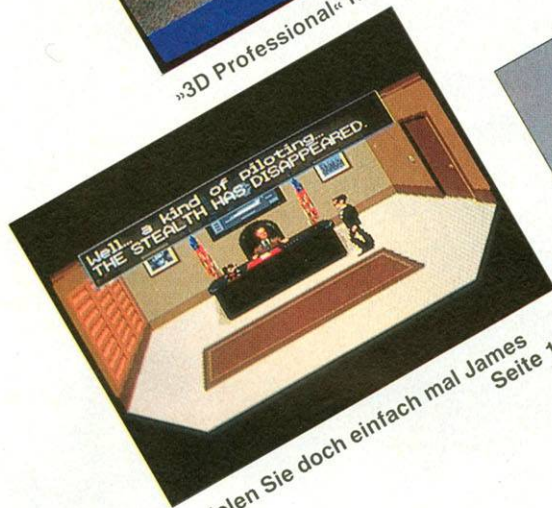
Freie Fahrt für Modula2 106
Datentypen und Variablen

Assembler ist keine Hexerei 117
Hinweise zur Codierung

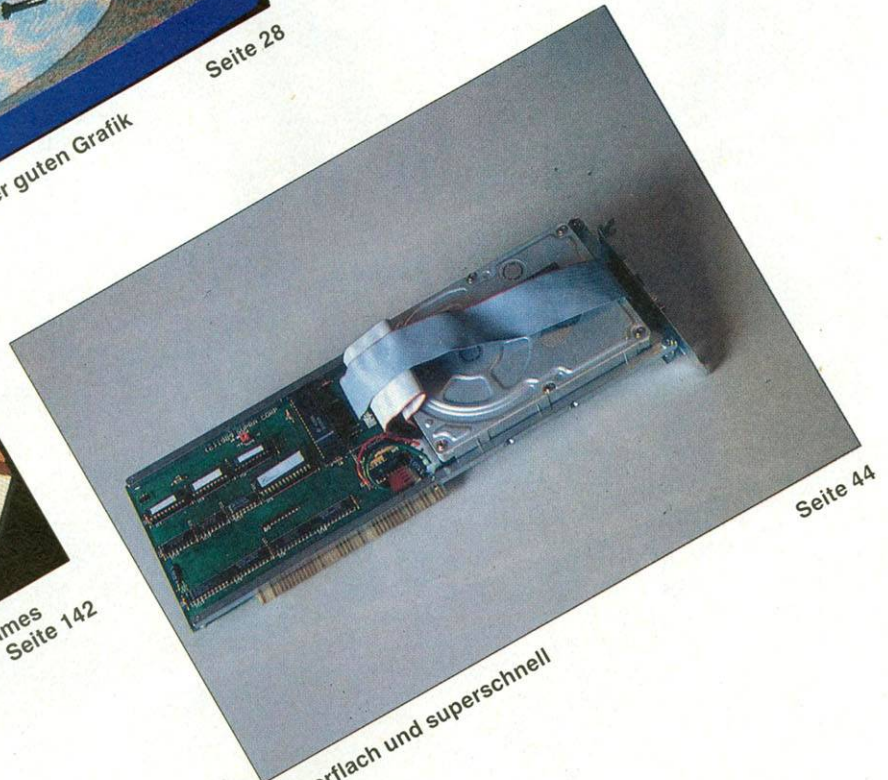


Seite 28

„3D Professional“ für Freunde der guten Grafik



Spielen Sie doch einfach mal James Bond
Seite 142



Seite 44

Superflach und superschnell

LISTING

Schau an!	94
<i>Texte lesen und verändern</i>	
Knacken Sie den Code!	102
<i>Denkspielereien in AmigaBASIC</i>	

WERKSTATT

Vielseitige Requester für alle Fälle	56
<i>Wie benutzt man die Library in anderen Sprachen</i>	
Reflections-Werkstatt	64
<i>Raytracing mit dem A500</i>	

PUBLIC DOMAIN

PD-Workshop	85
Msh – MS-DOS-Daten für alle!	86
MyBBS – Ein Zerberus-NetzPoint	88
Schnapp zu!	90
PD-Werkzeugkiste	122
PD aktuell	124

Kopieren fast zum Nulltarif	127
PD-Spieleshows	129

SPIELE

Spieletests	
Thunderstrike	131
Moonblaster	132
Rorke's Drift	132
Last Ninja 2	133
Pipe Rider	133
Powerboat USA	135
Tip Trick	135
All Dogs go to Heaven	136
Litti's Hot Shot	136
Matrix Marauders	137
Fruitmania	137
Italy 90	138
Starblade	138
Labyrinth	139
Wo in aller Welt ist	
Carmen Sandiego	139
Flood	140
The Colonel's Bequest	141
Operation Stealth	142

Anarchy	143
Demnächst auf Ihrem Computer	144
<i>Preview</i>	
AMIGA-DOS-Spieletips	146
<i>Helpline</i>	

RUBRIKEN

Editorial	5
Es lebe die DFÜ	35
Kleinanzeigen	82
Postkarte genügt	104
AMIGA-DOS-Tip	152
Leserbriefe	156
Wir suchen den schönsten Amiga	157
Das Superding	158
Wettbewerb	160
Impressum	161
Inserentenverzeichnis	161
Bücher	162
Vorschau	164



AMIGA 90

EINSTELLUNG

KOLN
REGISSEUR

AUFNAHME

SPULE

STELLER

DATUM

9. 11. 1990

Es ist soweit! Das Amiga-Ereignis des Jahres steht unmittelbar bevor. Die weltgrößte Messe für den kreativen Computer findet vom 9. bis 11. November 1990 in den Hallen 10 und 12 des Kölner Messegeländes statt.

Ottmar Röhrig

AMIGA
Köln 90

Auf den nächsten Seiten finden Sie unter anderem zwei schematische Messepläne. Getrennt für Halle 10 und 12 zeigen diese alle Aussteller, die sich zum Zeitpunkt der Drucklegung dieses Artikels angemeldet haben, inklusive ihrer Standnummern. Eine weitere Hilfestellung gibt Ihnen die Einfärbung der Stände im Hallenplan. Entsprechend der beim Plan zur Halle 12 abgedruckten Legende können Sie schnell erkennen, welche Art von Produkten auf einem bestimmten Stand vorherrschend ausgestellt wird.

Halle 12

Atlantis, einer der größten Software-Vertreiber für den Amiga, zeigt keine neuen Produkte. Doch kann hier jede vorrätige Soft- und Hardware direkt gekauft werden.

Auch im PD-Bereich gibt's Neues. **A.P.S. Electronic** vertreibt exklusiv auf der Messe »Power Packer Professional«. **Ariolasoft** ist zusammen mit **United Software** auf einem Stand. Zahlreiche Unteranbieter, viele Möglichkeiten zum Austesten der neuesten Spiele und Wettbewerbe mit Preisen machen diesen Stand vielleicht auch dieses Jahr wieder zum Treffpunkt der Spiel-Interessierten. Möchten Sie bestimmte Programme besitzen, scheuen aber noch vor der notwendigen Investition zurück, so bietet der Stand von **ComPlay** vielleicht das richtige für Sie. Dort wird »Software-Leasing« für Spiele und Anwenderprogramme vorgestellt, ein auf diesem Sektor neues Prinzip.

Die Firma **Dynamics** stellt einen neuen Joystick in den Mittelpunkt ihrer Messe-Präsentation. In edlem Design (transparent-blauer Kunststoff mit verchromten Feuerknöpfen) stellt der Competition Pro Star die Weiterentwicklung des Competition Pro dar. Möchten Sie nicht mit Ihren Computer-Problemen allein gelassen werden, so empfiehlt sich die Mitgliedschaft in einem der vielen Computer-Clubs. Ein besonders großer Vertreter stellt sich mit **Euro-comp** den Messebesuchern. Auf dem Stand können Sie direkt Mitglied werden und sich von Fachleuten beraten lassen.

Für alle technisch Interessierten bietet der Stand des **Fischertechnik-Computing** Neues. Neben den bekannten Produkten wie Dreiachs-Roboter, Plotter/Scanner und Interface, mit denen Sie Modelle von Industrie-Robotern nachbauen und vom Amiga aus steuern können, wird hier der neue Baukasten »Computing experimental« vorgestellt. Mit verschiedenen Sensoren ausgestattet, erlaubt er die Erfassung und Regelung von Meßdaten mit insgesamt 15 unterschiedlichen Modellen.

Die deutsche Share-Serie, bei der die Programmierer finanziell beteiligt sind, wird von **Herrmanns & Kommel** vorgestellt.

Heureka Techware produzieren Software für den Einsatz im schulischen Bereich. Zu verschiedenen Englisch- und Französisch-Lehrbüchern stehen Vokabelprogramme bereit, die den im Schulbuch vorkommenden Stoff behandeln. Weiterhin wird »Zenon« zu sehen sein, ein Programm zur Kurvendiskussion, das die Arbeit im Mathematikunterricht bis hin zur Oberstufe unterstützend begleiten soll. **Kingsoft** wird sicherlich mit neuen Spielprogrammen aufwarten. Ganz aktuell zur Diskussion steht da »Hägar«. Ob andere Programme bis zur Messe fertig werden, war noch nicht abzusehen.

Spielerherzen können auch auf dem Stand von **Leisure-soft** höher schlagen. Verschiedene Unteranbieter stellen dort ihre neuesten Produkte vor. **Stefan Ossowski** wird auf der Messe wahrscheinlich nur mit kommerziellen Programmen vertreten sein. Angekündigt sind AmigaBASIC-Erweiterungen, ein Quiz-Spiel, eine Notenverwaltung für Lehrer, Adressenverwaltungen und ein historisches Simulationsspiel.

Haben Sie Interesse an PD-Software, so sei hier schon gleich auf den Stand der Firma **Pielago Software** verwiesen. Disketten aus 38 Serien mit mehr als 4500 Ausgaben sind zu absoluten Messepreisen zu erwerben.

Pulsar zeigt den »ADAPT« 68030-Assembler mit Unterstützung für MMU-Bausteine, umfangreicher Makroverwaltung und BLink-kompatiblen Linker. Ein Disassembler, der

auch die Zykluszeiten für die Ausführung bestimmter Befehle abgibt, rundet das Bild ab.

»AmiExpress BBS« ist ein neues Mailboxprogramm. Eine kleine Sensation ist »Macro Paint«. Es erlaubt mit allen 4096 Farben des Amiga im Hi-Res-Modus (Dynamic HiRes) zu zeichnen. Die 12 Bit tiefen Farbgrafiken können über einen AREXX-Port automatisch bearbeitet und das gesamte Programm von anderer Software gesteuert werden.

Speziell an den Profi-Bereich wendet sich ein neues DTP-Programm namens »Saxon Publisher«. In Köln in neuer Version zu bewundern, unterstützt es nun auch die europäischen Standards und kann EPS-Dateien importieren. Die ausschließliche Ausgabe der Daten als PostScript-Code stört im professionellen Bereich wenig, weshalb sich ein Blick auf das Programm nun sicherlich lohnt.

Ein weiteres Rendering-Paket kommt mit Painter 3D auf den Markt. Editor und Render-Algorithmen erlauben Ihnen, in einem Programm vereint, die Erstellung von Extrude-oder Rotationskörpern, die Sie mit Farben und beliebig vielen Lichtquellen versehen können. Die Objekt-Formate sind kompatibel zu denen von Sculpt 4D – ein einfacher Datenaustausch ist also gewährleistet. Bis zu 6,5 MByte innerhalb eines A500 sind mit dem »Base-Board« und der Erweiterungskarte »XRAM board« zu erreichen. Diese Erweiterung wird ohne Lötarbeiten intern eingebaut und ist ebenso voll kompatibel zu den verschiedenen Revisionen der Hauptplatine, wie zu AGNUS, Fat AGNUS und Big AGNUS.

Rainbow Arts, eine Firma, die den Spielefans sicherlich bekannt ist, stellt mindestens zwei neue Programme vor. »M.U.D.S.« stellt die Umsetzung einer (gottseidank) noch nicht entdeckten Sportart dar. Die Kriminellen einer Stadt treten in mehreren Runden gegeneinander an und werden nach einem Sieg amnestiert – wenn sie überleben! Ein brutales Spiel, bei dem Sie als Manager die Band leiten und Einsätze koordinieren müssen. »Masterblazer« ist da schon eine wesentlich friedlichere Sportart. In der Umsetzung des bekannten Lucasfilm-

Spielautomaten wird hier im Universum Ballblazer gespielt, eine moderne Variante des Fußballs. Sie steuern die Rotofoils Ihres Spielers und treten gegen bis zu sieben menschliche Gegner an. Mit mehr als 100 Farben auf einem Bildschirm sicherlich ein Spektakel höchster Güte!

Für Programmierer und PD-Fans bietet der **Schneider Verlag** neue Produkte. Die Sammelwerke »Das aktuelle Praxishandbuch zu Amiga Public-Domain« und »Das aktuelle Praxishandbuch für Amiga-Profis« werden vorgestellt. Frische Fish-Disks sind dabei ebenso zu Messe-Preisen zu haben, wie Gespräche mit Autoren und Herausgebern diverser Bücher aus dem Schneider-Verlag.

Soft+Sound stellt ein Konzept zur Gründung eines Computerspiele-Ladens vor. Finanzierungsmodelle, fachkundige Beratung und Informationsmaterial sollten von jedem betrachtet werden, der einen Gedanken in dieser Richtung mit sich trägt.

Wieder auf ihre Kosten kommen Spielefans auf dem Stand von **Software 2000**. Dort sind die Programmierer verschiedener Spiele aus diesem Hause anzutreffen. Zehn neue Projekte sollen vorgestellt werden. Der Berliner Verlag **technic-Support** wird die gesamte PD-Bibliothek zum Kauf anbieten. Diese über 1300 Seiten starke Sammlung von deutschen Beschreibungen guter PD-Programme kann inklusive der Disketten am Stand begutachtet werden. Des weiteren wird der DTP-Service vorgestellt, der nicht nur für Berliner Firmen interessant ist. Auf hochwertigen Geräten wird professionelles Desktop-Publishing von der Gestaltung einzelner Logos bis hin zur Erstellung mehrseitiger Prospekte gezeigt.

Anwendernah wird es dann wieder auf dem Stand von **Wolf Computertechnik**. Die neueste Version der Finanzbuchhaltung »FIBU deluxe+« ist nun mit einem deutschen Handbuch versehen worden und erlaubt die Verwaltung von bis zu 2000 Konten. Gewinn- und Verlustanalysen, Dialogverbuchung, Kassenberichte, verschiedene Steuerschlüssel und Mandatenfähigkeit sind nur einige der Stichwörter zu

diesem Programm. Auch die neueste Version 3.0 des PD-Satzprogrammes »TeX« ist auf dem Stand zu sehen. Das Strategie-Spiel »Manager« rundet das Angebot der Neuvorstellungen ab.

Halle 10

In Halle 10 ist der größte Teil der Hardware-Entwickler und -Vertreiber zu finden. Im Gegensatz zur Halle 12 wenden sich die Aussteller hier mehrheitlich einem professionellen Publikum zu und zeigen an vielen Beispielen, daß der Amiga bereits heute nicht mehr aus professionellen Anwendungen wegzudenken ist.

A+L, die Entwickler des bekannten Modula-Compilers »M2Amiga«, zeigen diesen in der Version 3.32. Alle bekannten Fehler der Vorgängerversionen wurden behoben. Insbesondere ist nun auch eine Lauffähigkeit der Programme unter der Oberfläche »WShell« gegeben. Andere Erweiterungen zu M2Amiga beinhalten einen neuen Editor und einen optimierenden Linker. Auch die erste kommerzielle Umsetzung der Programmiersprache »Oberon« wird auf der Messe vorgestellt. Die auf Modula-Programme spezialisierte PD-Reihe »AMOK« können Sie ebenfalls bei A+L erstehen.

Das Platinenlayoutprogramm »Newio 3.0« wird bei **Alphatron** zu sehen sein. Neu sind unter anderem ein komfortabler Bauteileditor, verbesserte Autorouting-Algorithmen sowie die Möglichkeit zur Konvertierung der Plotdaten in das Gerber-Format.



Alle Musikinteressierten finden auf dem Stand der Firma **AV Soft** neue MIDI-Software. Die Frequenzerkennung eines analogen Signals mit anschließender Umwandlung in MIDI-Signale ist vielleicht eine der kleinen Sensationen auf der Messe. So lassen sich nämlich mit jedem analogen Instrument, wie zum Beispiel einer akustischen Gitarre, MIDI-Geräte ansteuern. »DigiMIDI« vereint MIDI-Interface und Audio-digitizer in einem Gerät. Die maximale Samplingrate von 100 KHz und eine regelbare Eingangsempfindlichkeit ma-

chen das Programm ideal für die Zusammenarbeit mit dem eben erwähnten »Sound-2-MIDI«. **bsc** und **Fujitsu** stellen auf einem gemeinsamen Stand aus. Dort sind unter anderem Multifunktionskarten zu sehen, auf der sich Speichererweiterungen und zusätzliche Schnittstellen befinden. Auch optische Laufwerke mit der neuen ALF3-Treibersoftware sind an diesem Stand im Einsatz mit dem CAD-Programm »Ultra-Design CAD« zu sehen.

Als Schirmherr der Veranstaltung ist natürlich auch **Commodore** vertreten. Der gesamte Stand wird mit A3000-Rechnern ausgestattet sein, auf denen Neues und Altbewährtes zu sehen sein wird. Als neues CAD-Programm sehen Sie »DynaCADD«, das den bisherigen Spitzenreiter in diesem Bereich (»X-CAD«) weit in den Schatten stellen kann. Höhere Verarbeitungsgeschwindigkeit, Datenkompatibilität zu AutoCAD und die Möglichkeit, echte 3D-Konstruktionen zu erstellen, sprechen für sich.

Midi

Auf dem Stand von **CompuStore** sind Produkte der Firmen **MicroBotics**, **ASDG** und **Black Belt Systems** zu sehen. Neben den deutschen Versionen der Handbücher zum Editor »Cygnus Ed 2.0« sowie zum Shell-Ersatz »WShell« dürfte vor allem das Produkt »HAME« von Interesse sein. Diese Kombination aus Hard- und Software erlaubt die Darstellung von 256 oder 262144 Farben gleichzeitig an jedem Amiga – und das aus einer Palette von 16,7 Millionen Farben.

ASDG zeigt die Scanner JX100 und JX350/450 jeweils mit der aktuellsten Version der Scanner-Software, die für Profis und engagierte Anwender gleichermaßen interessant ist. Das Bildbearbeitungsprogramm »The Art Department« können Sie in der Version 2.0 begutachten. Es erlaubt die Vergrößerung/Verkleinerung, Rotation, Format-Konvertierung (GIF, Sculpt, Turbosilver, IFF, DigiView, TIFF, Rendition), Farbpalettenmodifikation und einiges mehr von 24-Bit-Grafiken mit 16,7 Millionen Farben. Zahlreiche weitere Funktionen machten das Programm schon in der Version 1.0 zu einer wahren Hilfe für jeden, der mit

24-Bit-Grafiken arbeiten muß. Die Speichererweiterungen von MicroBotics sind auf die verschiedensten Bedürfnisse und Rechnertypen ausgelegt. Besonderheit ist jedoch ein »RAM-Laufwerk«, das Daten mit Hilfe eines eigenen Netzteils auch bei ausgeschaltetem Rechner behält. Dies könnte eine Alternative zu den ebenfalls angebotenen Hardframe-Festplattencontrollern darstellen. Weiterhin können Sie auf dem CompuStore-Stand Modems der Firma Applied Engineering und ein 1,44-MByte-Diskettenlaufwerk betrachten.

Computertuning Ralf Jochheim zeigt unter anderem eine in Zwei-MByte-Schritten aufrüstbare Speichererweiterung für den A2000 und eine Grafikkarte, die in Zusammenhang mit Monitoren der Firmen Sony und Eizo vorgeführt wird.

Data Becker zeigt auf seinem Stand wohl die hoffentlich endgültig fertige Version der Textverarbeitung »BECKERText II«, die bereits auf der CeBIT in einer Vorversion zu sehen war. Eventuell sollen auch Workshops zu diesem Programm durchgeführt werden. An Literatur steht das neue Amiga-Intern an, das die beiden bisher verkauften Bände zusammenfaßt. Auch ein Animationsprogramm ist angekündigt.

Wer sich schon immer einmal über die Entfernung zwischen verschiedenen Orten in der BRD informieren wollte, kann auf dem Stand von **Delta Konzept** eine Software kaufen, die 9000 Orte und 10 Millionen Entfernungen gespeichert hat. Auch ein Hotel- und Restaurantführer mit insgesamt 14000 Häusern ist hier zu sehen. Insgesamt zehn Terminals mit A2500- und A3000-Rechnern zum Spielen und Testen der angebotenen Software stehen sicherlich bei den angesagten Fahrsimulator-Wettbewerben im Vordergrund.

Im Btx-Bereich zeigt **Drews** die neueste Version 1.5 des Makro-Manager-Moduls, das die Automatisierung von immer mehr Vorgängen innerhalb des Btx-Systems erlaubt. Zusammen mit dem Software-Decoder »Btx/Vtx-Manager 2.2x« kann hier mit jedem Modem oder Akustikkoppler beziehungsweise mit der Anschlußbox der Post Btx betrie-

ben werden. **DTM** zeigt Produkte verschiedenster Firmen auf dem Stand. Der derzeit wohl schnellste SCSI-II-Controller ist von **GVP** und wird in Zusammenarbeit mit optischen Laufwerken, einer neuen Wechselplatte von Ricoh oder mit bis zu 200 MByte Speichervermögen als Einsteckkarte gezeigt. Natürlich können Sie sich hier auch über die Turboboards von GVP informieren. Das EasyI-Grafiktablett wird mit einem neuen Grafikprogramm an anderen Stellen gezeigt.

Bekannte Produkte wie »Pixmate«, »Diskmaster«, »IntroCAD«, »MAC-2-DOS«, »DOS-2-DOS«, »DISK-2-DISK« sind ebenfalls zu sehen. Echte Neuerungen könnten die restlichen Module der PI-Serie sein. Angekündigt sind – neben den erhältlichen Plotter- und Matrix-Modulen – zumindest der Taschenrechner, PI-Optimierung und PI-Statistik. Alle Programme sind in erster Linie für den Einsatz im mathematischen Bereich der Oberstufe und des Studiums gedacht. Auch der bereits mehrmals angekündigte »Publishing Partner Master«, die deutsche Weiterentwicklung des DTP-Programms »PageStream«, soll auf dem Stand zu sehen sein. Neben dem verbreiteten Y-C-Genlock wird **electronic design** auf der Messe drei neue Produkte vorstellen. Ein Y-C-Farbsplitter erzeugt exzellente Farbauszüge aus jedem VHS/S-VHS/Hi8/Video8-Signal und stellt sie einem Digitizer zur Verfügung. Ein passender Video-Konverter sorgt für die Aufzeichnungsmöglichkeit von RGB-Signalen des Computers als FBAS- oder Y-C-Signal. Des weiteren ist ein neues PAL-Genlock zu sehen.

Festplatten mit einer effektiven Zugriffszeit von 11ms, die durch den 64 KByte großen Cache auf dem Controller erreicht werden, stellt **FSE** aus. Sie erreichen eine Übertragungsrate von bis zu 600 KByte/s und arbeiten allesamt mit 16 Bit breitem Bussystem. Möchten Sie Ihre Festplatte dem A500 einverleiben, gibt es derzeit nur ein Gerät von **Gigatron**, das derartige Wünsche erfüllt. »Arriba« faßt 20 MByte, wird unsichtbar im A500 eingebaut und direkt vom Computer mit Strom versorgt. Speichererweiterungen für denselben



AMIGA OFFICE

Alles aus einer Hand

nur DM 398.-

Amiga Office - das sind sechs leistungstarke Programme. Jedes für sich ist eine durchdachte Hilfe für Ihre individuellen Problemlösungen, und darüberhinaus, für übergreifende Anwendungen, optimal abgestimmt auf das Zusammenspiel mit den anderen Programmen. Amiga Office besteht aus den folgenden Modulen:

Textverarbeitung

Mit Office Write steht Ihnen eine effiziente Textverarbeitung zur Verfügung, mit der die Arbeit - von der einfachen Korrespondenz bis hin zur umfangreichen Dokumentation - dank Makros und vielen anderen Features zum reinen Vergnügen wird.

Dateiverwaltung

Office Files bietet Dateiverwaltung auf höchstem Niveau. Jetzt können Sie schnell Adreß- oder Kundendateien, und z.B. CD- oder Video-Sammlungen organisieren, und mit dem Amiga verwalten und vielfältig auswerten.

Businessgrafik

Trockene Zahlenreihen grafisch auszuwerten und visuell aufzubereiten ist die Spezialität von Office Graph. Durch einfache Bedienung und die Vielfalt der verschiedenen Charts werden Sie sehr schnell ins Staunen geraten.

Tabellenkalkulation

Office Calc ist eine Tabellenkalkulation, die keinen Vergleich scheut. Hohe Rechengeschwindigkeit, einfache Bedienung und volle Lotus 1-2-3 Kompatibilität sind nur einige der umfangreichen Funktionen und Möglichkeiten dieses Programms.

Desktop Publishing

Mit Postscriptunterstützung, Vektorfonts und einer umfangreichen Funktionspalette zählt Office Page bereits zum Feinsten, wenn es darum geht, Ihre Drucksachen in die richtige Form zu bringen.

Rechtschreibkorrektur

Office Korrektur überprüft Ihre Texte auf Rechtschreibfehler und besticht dabei vor allem durch die enorme Arbeitsgeschwindigkeit und das umfangreiche Lexikon (ca. 400.000 Stichwörter).



GOLD DISK

Marktplatz 16 • 4018 Langenfeld
Telefon 0 21 73 / 7 10 93

Vertrieb Österreich:

DARIUS

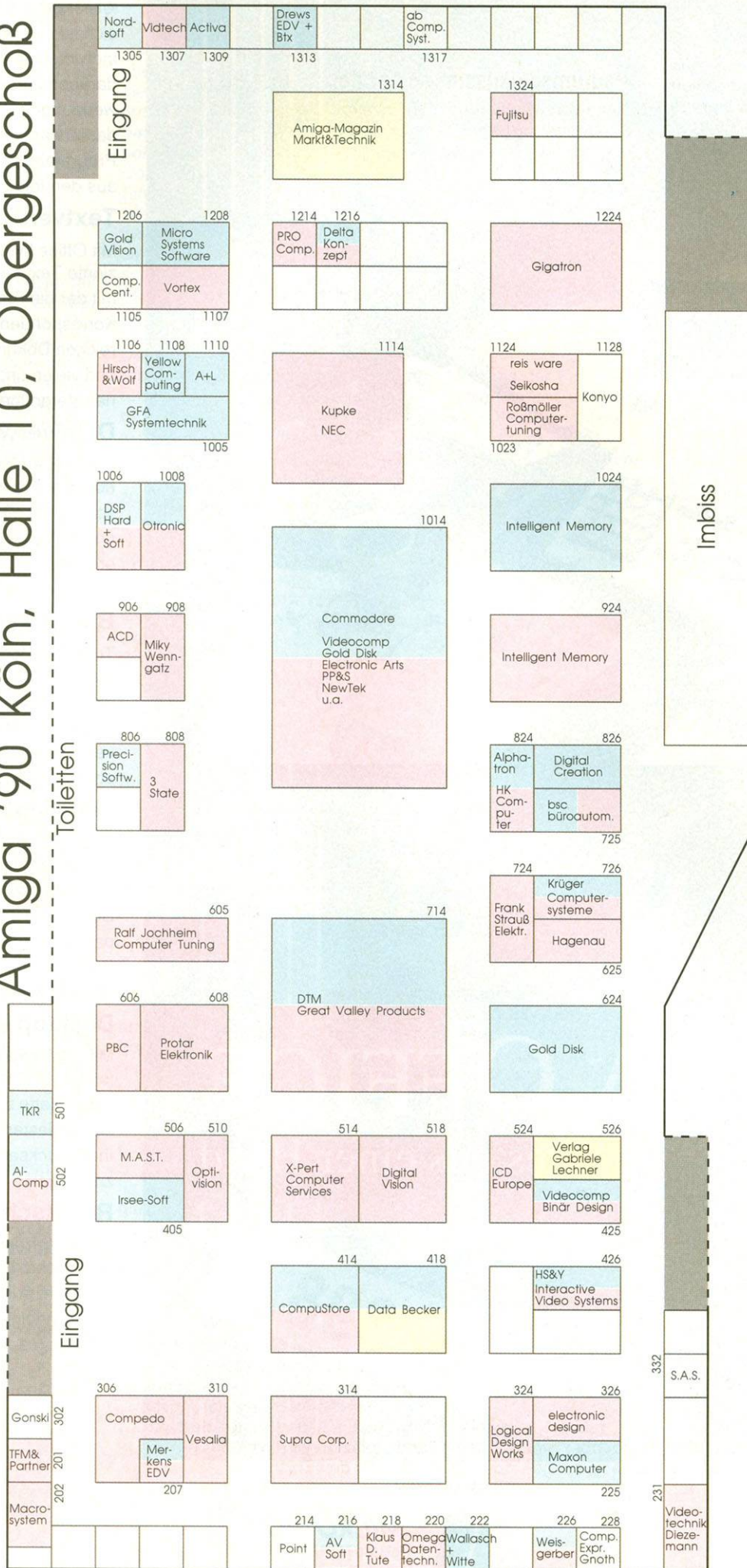
Andreas-Huger-Gasse 56/1
A-1220 Wien

Vertrieb Schweiz:



MICROTRON
COMPUTERPRODUKTE
Postfach 69 Bahnhofstr. 2
CH-2542 PIETERLEN

Amiga '90 Köln, Halle 10, Obergeschoß



Rechnertyp sowie für den A2000 können in Schritten von 512 kByte aufgerüstet werden und sind ebenfalls vorführbereit.

Multimedia und Profi-DTP sind die Schwerpunkte auf dem Stand von **Gold Disk**. Lange angekündigt und wahrscheinlich auf der Messe tatsächlich zu sehen, ist »Professional Page 2.0«, das durch zahlreiche neue Funktionen andere DTP-Programme in den Schatten stellen soll. Natürlich ist auch das schon im Handel erhältliche Vektor-Zeichenprogramm »Professional Draw 2.0« in direkter Zusammenarbeit mit dem DTP-Programm zu sehen.

»ShowMaker« erlaubt die interaktive Erstellung ganzer Multimedia-Shows. Dabei werden Grafiken, Texte und Animationen in fast allen bekannten Formaten ebenso miteinbezogen, wie die Zuspaltung von Daten aus Videorecordern, Dia-Projektoren, CDV-Playern, CD-Playern oder optischen Laufwerken. Raytracing-Freunde finden auf dem Stand auch ein neues Raytracing-Programm für den Heimbereich. Drei DTP-Arbeitsplätze mit den neuesten Versionen der Programme »PixelScript«, »PageStream« und »VectorTrace« sind bei **Gold Vision** zu sehen.



Anwender aus dem Videobereich finden bei **Hagenau Computer** einen neuen RGB-Splitter, dessen Joystick- und Parallel-Port durchgeschleift ist und trotz geringer Abmessungen den Einbau eines »Digiview« oder »Deluxe View« erlaubt, dessen Version 4.1 der Software ebenfalls vorgestellt wird. Eventuell ist auch die Version 3.0 der Audiodigitizer-Software »Deluxe Sound« zu sehen, was bei Redaktionsschluß jedoch noch nicht feststand.

Auf dem Stand von **Heinrichson, Schneider & Young** (HS&Y) sind neben den Trumpcard-Controllern der Firma **Interactive Video Systems** auch Speichererweiterungen zu sehen, die mit dem Controller kombiniert werden können. Angekündigt ist außerdem der »Broadcast-Titler«, ein neues Videobetitelungsprogramm der Extraklasse.

Podiumsdiskussionen auf der Amiga '90 in Köln

Diese Veranstaltungen unterscheiden sich eindeutig von den vollkommen getrennt laufenden Masterclasses, da sie innerhalb der Halle im Studio-5-Cafe stattfinden. Dort sind folgende Themen geplant:

Freitag 9.11.: Raubkopien/Software-Piraterie

Samstag 10.11.: Viren/Public Domain

Sonntag 11.11.: Ist Computerkunst Kunst?

Übrigens werden sämtliche Veranstaltungen des Forums von AMIGA DOS gesponsert und organisiert. Selbstverständlich stehen ständig Redakteure der AMIGA DOS und namhafte Persönlichkeiten der Amiga-Szene den Besuchern zur Verfügung.

Zu allen Podiumsdiskussionen werden mehrere kompetente Vertreter von meist gegensätzlichen Meinungen geladen. An der Diskussion kann und soll sich das Publikum beteiligen. Alle Podiumsdiskussionen sind kostenfrei, eine Zeitbegrenzung wird nicht erwartet.

Richtig innovativ ist ein kleines neues Produkt von **HK-Computer**. Der Maus & Joystick-Adapter schaltet zwischen beiden angeschlossenen Geräten elektronisch um, sobald man eines bewegt – kein lästiges Umstecken oder Umschalten mehr. »Professional MIDI-Interface« ist ein hochwertiges MIDI-Interface, mit dem durch spezielle Opto-Koppler mit digitalem Ausgang Signalverluste nahezu der Vergangenheit angehören.

ICD Europe, dieses Jahr erstmals mit einem eigenen Stand vertreten, stellt drei neue Produkte vor. Ein Flickerfixer für den internen Einbau in A500, A1000 oder A2000 erlaubt den Anschluß eines handelsüblichen VGA-Monitors, unterstützt NTSC und PAL sowie volles Overscan, was bisher noch mit keinem Gerät dieser Art möglich war. Des weiteren ist eine 68000-Prozessorkarte mit 14,8 MHz zu sehen. Ein Bussystem für AT-Bus-Festplatten soll zu späterer Zeit einmal den Einbau einer AT-Festplatte in den A500 erlauben.

Video

Wenig neue Produkte, dafür aber Altbewährtes hat **Irsee-Soft** zu bieten. Die Entwickler von »TURBOprint« und »TURBOprint Professional« werden anwesend sein und die Fähigkeiten dieser exzellenten Programme unter Beweis stellen. Nebenbei wird auf dem Stand eine günstige und präzise Alternative zur Amiga-Maus verkauft.

Einen Echtzeit-Farbdigitizer können Sie auf dem Stand von **Krüger Computersysteme** sehen. Der »Framer-O« benötigt zum Digitalisieren eines Bildes kein Standbild und erstellt Grafiken mit maximal 352x704 Pixeln. Die dazu notwendige Software »VDController« ist auch in Anpassungen für andere Geräte erhältlich, kann also auch ohne den »Framer-O« in Betracht gezogen werden. Außerdem sehen Sie auf dem Stand eine Hard- und Softwarekombination zum Empfang des Videotext-Signals aus dem Fernsehprogramm.

Den neuen Versionen des Betriebssystems wird auf dem Stand der Firma **Kupke** schon Rechnung getragen. Umschaltplatinen für die neue Kickstart 2.0, die 512 KByte umfaßt, (für A2000) sowie eine SCSI-Festplatte mit integrierter Speichererweiterung und Kickstart-Umschaltplatinen für den A500 sind auf dem Stand zu sehen. Des weiteren wird ein neues System vorgestellt, das dem A500 durch die Verwendung von 68030/882-Prozessoren zu einer extrem hohen Geschwindigkeit verhilft.

Vielleicht sind ja auch hier schon die ersten Erweiterungskarten für den A3000 zu sehen.

LDW wird auf der Messe die Produkte von **California Access** vorstellen. Insbesondere ist dies eine Erweiterungsbox, die auf und hinter dem A500 platziert wird, vier Zorro-II-Slots für den Einsatz von A2000-Karten (inklusive der Bridgeboards), ein eigenes Netzteil

für die Versorgung auch des A500, einen Ventilator, Platz für mehrere Laufwerke und Festplatten sowie Prozessorkarten bereithält. Diese Bodega Bay kann mit den ebenfalls dort zu sehenden Erweiterungskarten aufgerüstet werden.



Als wirkliche Neuheit auf dem professionellen Software-Markt kann sicherlich »Imagine« angesehen werden, das auf dem Stand des **Lechner Verlages** zu bewundern ist. Dieses neue Raytracingprogramm ist der stark erweiterte Nachfolger von Turbo Silver 3.0 und stellt nicht nur neue Animationsfunktionen zur Verfügung, sondern weist auch eine bisher nicht dagewesene Raytracing-Qualität auf. Natürlich sind auf dem Stand auch die neuesten Bücher des Verlages zu sehen.

Eine echte Sensation dürfte auch das neueste Produkt der Firma **Merkens** werden. Der »VD2001«-Digitizer ist nicht nur in der Lage, Farbbilder aus dem laufenden Videosignal zu digitalisieren, sondern stellt gleichzeitig einen Framebuffer zur Verfügung, der Bilder – beispielsweise aus Raytracing-Programmen – mit 16,7 Mio Farben darstellen kann. Die Software von Peter Reuschling wartet mit Bildbearbeitungsfunktionen auf, die bisher auf dem Amiga noch nicht zu sehen waren: Maskenplane zur Kombination von Videosignal mit Grafik ohne Genlock, Bild-In-Bild-Funktion, Bearbeitung der Bilder direkt im Frame-Buffer. Des weiteren sind die Produkte »miniGen« und »maxiGen« zu sehen, die Genlocks für den Heimbereich und den Studioeinsatz darstellen.

Raytracing

Die Textverarbeitung »excellence!« wird auf dem Stand der Firma **MSS** in der Version 2.0 zu sehen sein. Sie wurde um einige Funktionen erweitert. **Omega Datentechnik** wartet auf der Messe mit der ersten professionellen Audio-Steckkarte für den A2000/3000 auf. Samplingraten bis hin zur CD-Qualität, aktive Höhen- und Tiefenregelung, eigene Tonausgänge und Lautstärkerege-

Masterclasses der Amiga '90 in Köln

Natürlich werden auch dieses Jahr wieder Sonderveranstaltungen zu verschiedenen Themen durchgeführt.

Sie kosten jedoch eine von Veranstaltung zu Veranstaltung unterschiedliche Gebühr, die im Eintritts-Preis nicht enthalten ist.

Alle Masterclasses werden von Profis des jeweiligen Gebietes durchgeführt, dauern drei Stunden lang und werden während der Messezeit insgesamt dreimal mit dem gleichen Inhalt abgehalten.

Die vorliegende Liste informiert über die bereits geplanten Veranstaltungen, die jedoch unter Umständen noch erweitert werden können.

Thema: Animation

Referent: Erfried Prenissl, H. Czerny (Studio 5)

Inhalt:

- von der Idee zum fertigen Film
- Planung, Konzeption und Durchführung komplexer Animationen
- Unterschiede und Möglichkeiten verschiedener Grafikprogramme

Zeitpunkt: Freitag 14 Uhr, Samstag 10 Uhr, Sonntag 14 Uhr

Thema: Musik/MIDI

Referent: Chris Hülsbeck, Peter Thierolf (Programmierer, Musiker)

Inhalt:

- Komposition von Musik für professionelle Spiele

- Umgang mit einem Editor für Musikdaten und Amiga-Sounds

- Einbinden des Amiga in den MIDI-Datenverbund
- Kompositionstechniken mit einem MIDI-Sequencer
- Praktische Beispiele

Zeitpunkt: Freitag 14 Uhr, Samstag 10 Uhr, Sonntag 14 Uhr

Thema: Grafik und Animation

Referent: Gabriele Lechner, Walter Friedhuber (Lechner Verlag, Buchautor)

Inhalt:

- Produktion eines Werbespots mit »DPaint III«
- Konstruktion von 3D-Objekten mit »Turbo Silver«
- 3D-Techniken mit »Imagine«

- Tips und Tricks in Kameraführung, Trickfilmszenierung und weiteres

Zeitpunkt: Freitag 10 Uhr, Samstag 14 Uhr, Sonntag 10 Uhr

Thema: Video

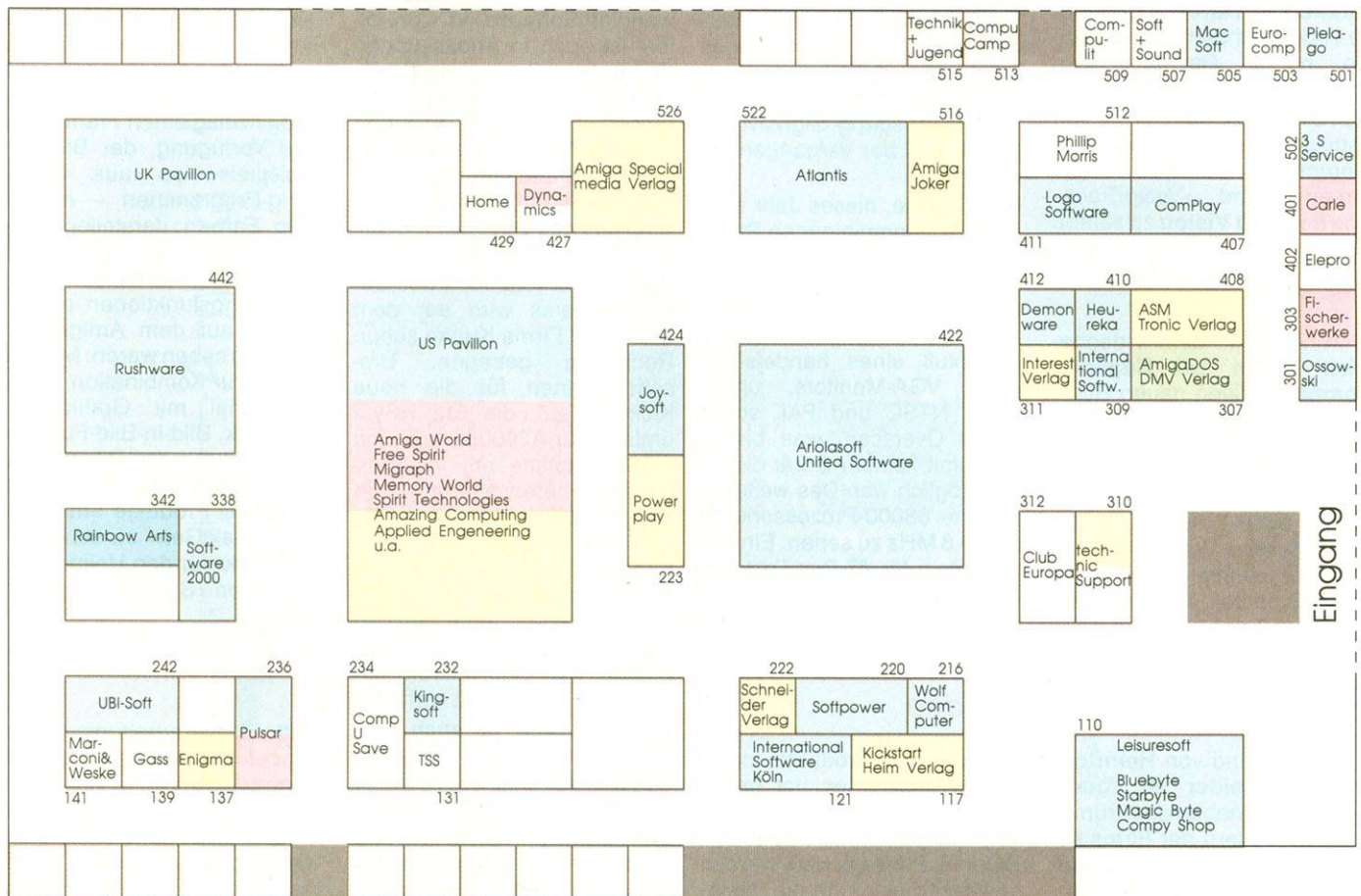
Referent: Jürgen Schneider (Videoprofi, HS&Y)

Inhalt:

- Amiga-Konfigurationen im Video-Bereich (Soft- und Hardware)
- Elemente der Videoproduktion
- Beispiele von Studioproduktionen
- Anwendungen für Fortgeschrittene

Zeitpunkt: Freitag 10 Uhr, Samstag 14 Uhr, Sonntag 10 Uhr

Amiga '90 Köln, Halle 12, Obergeschoß



Übergang zur Halle 10
Restaurant/Imbiss

- Software
- Hardware
- Verlage
- Dienstleistungen/Sonstiges
- AmigaDOS
- Auf/Abgänge

lung werden mit der Software »A.D.D.A.«, die grundsätzlich mit 16-Bit-Daten arbeitet, voll ausgenutzt. So kann ein digitaler Schnittplatz realisiert werden, was vor allem im Bereich der Multimedia-Präsentationen Anklang finden dürfte. Neben dem Echtzeit-Digitizer »Vidi-Amiga« spezialisiert sich **Optivision** auf der Messe hauptsächlich auf die Präsentation aktiver Arbeit im Videobereich.

Ebenfalls auf dem Videosektor aktiv ist die Firma **PBC**, die mit dem »Digi-Tower« die Weiterentwicklung des bekannten RGB-Splitters »Digi-Splitt-Junior« zeigt. Das Produkt hat ein vollkommen neues Design, erlaubt jetzt die nachträgliche Aufrüstung mit einem Genlock-Modul und bietet darüber hinaus eine serienmäßige Druckerumschaltung für den parallelen Port.

Sicherlich gespannt darf man auf die Einführung des »Videomaster« sein, der bereits auf der CeBIT einer erlauchten Auswahl von Besuchern gezeigt wurde. Dieses Gerät ist für die professionelle Video-Postproduction gedacht und ist modular aufgebaut. Ausgehend vom Grundmodell, das Farbkorrekturen auf digitaler Ebene in Echtzeit erlaubt, können zahlreiche Erweiterungskarten eingesteckt werden. Fernsehnormen können ohne Verluste in Echtzeit konvertiert, Bilder in den Amiga übertragen und Zeitbasen korrigiert werden. Als Nebenprodukt enthält das Grundgerät auch noch einen Flickerfixer.

Für den DTP- oder DTV-Bereich stellt **reis-ware** drei verschiedene Handy-Scanner vor, von denen einer mit 4096 Farben, die anderen jedoch in Graustufen arbeiten. Zwei spezielle Mäuse, eine davon mit Infrarot-Signalübertragung, sind ebenfalls zu sehen.

Roßmüller wird auf der Messe einen sehr preisgünstigen Flickerfixer für A500, A1000 und A2000 vorstellen. Für A500-Besitzer ist das XT-Board interessant, welches einen kompletten PC-XT im Gehäuse des Amiga unterbringt und zusätzlich eine RAM-Erweiterung darstellt. Wer mit diesem Speicher nicht auskommt, kann auf die 10-MByte-Karte zurückgreifen, die jedoch nicht für den A2000 erhältlich ist. Dieser kann jedoch

– wie der A3000 auch – mit einer eigenen Version um bis zu 8 MByte aufgerüstet werden. Video-Professionals werden auf dem Stand von **TFM & Partner** gern gesehen sein. Die Genlock-fähigen »PAL-Encoder« von Magni für den Einsatz in den »großen« Amigas 2000, 2500 und 3000 weisen Studioqualität auf.



Die erste Btx-Editier-Software für den Amiga wird auf dem Stand der Firma **TKR** zu sehen sein. Als Weiterentwicklung des Btx-Decoders »Multiterm-pro«, der auch in einer neuen Version vorgestellt wird, ist »Multiterm-ED« für Btx-Anbieter gedacht. Ein neues Modem von **BEST** ermöglicht mit der hier vorgestellten Software die Versendung von Fax-Mitteilungen aus dem Amiga heraus. Besonders für die angekündigte UUCP-Netzwerk-Version des Mailbox-Programms »Euromail« ist auch das nächstgrößere Modem der Firma interessant, das 9600 Baud mit MNP5-Datenkompression erlaubt und so auf eine effektive Übertragungsrate von 19200 Baud kommt.

BTX

Der professionellen Video-Postproduction verschrieben hat sich die Firma **Video-comp**, die zusammen mit **Binär Design** auf einem Stand ist. »DVE10« ist ein semiprofessionelles System, welches Videoprozessor, PIP-Prozessor, Genlock, Digitizer und Video-/Audiomischer in sich vereint und den Amiga kostengünstig in einen Titelgenerator verwandelt. Für den Studioeinsatz gedacht und auf die Zukunft ausgerichtet ist das modulare Video-Bus-System, das vom Black-Burst-Generator bis hin zur Framestore-Karte bis zu 21 Steckkarten aufnehmen kann. Ob das angekündigte System »Miranim«, »Mirashading« und »Sabrina« zu sehen sein wird, darf bezweifelt werden, da dies auch von der Fertigstellung des Betriebssystems »Helios« für den Amiga abhängt.

Ein weiterer Prototyp eines Gerätes wird vielleicht auf dem Stand von **Videotechnik Diezemann** für Furore sorgen.

Der »3D Videoprozessor« erlaubt die Digitalisierung von Videosignalen mit 24 Bit und einer Auflösung von 768x580 Pixeln in Echtzeit. Spezielle Grafikprozessoren sorgen dann für eine dreidimensionale Animation des laufenden Videosignals mit 50 Bildern pro Sekunde. Effekte wie Drehen, Spiegeln, Zerren, Zoomen, Kippen, Rollen oder ähnliches können in Echtzeit am laufenden Videobild vorgenommen werden.

Eine fast jungfräuliche Sparte hat die Firma **Wallasch & Witte** auf dem Amiga entdeckt. Vorgestellt wird die Reihe »CHARTech«, die Programme zur Aktienanalyse in drei unterschiedlich umfangreichen Versionen darstellt.

Die Videobackup-Software von **Weisgerber** wird auf der Messe in einer Version zu sehen sein, die auch die Abspeicherung selektierter Bereiche einer Festplatte ermöglicht. Das System stellt eine interessante Alternative zu den (lästigen) Diskettenbackups und den (teuren) Streamern dar, da es als Speichermedium einen handelsüblichen Videorecorder zur Datensicherung verwendet. Eine besonders schnelle Festplatte mit 512 KByte Cache-RAM, SCSI und einer 2. parallelen Schnittstelle rundet das Angebot dieses Entwicklers ab.

Leute, die mit dem Design ihres A500 nicht mehr zufrieden sind, können bei **Miky Wenn-gatz** ein neues Gehäuse erstellen, das nach etwas Umbauarbeit sogar eine externe Tastatur zur Verfügung stellt. Doch werden vor allem die Messerabatte auf Produkte wie PC-Karten, Btx-Kits, Laufwerke, MIDI-Interfaces und Drucker viele Leute zu diesem Stand locken.



Die ersten Erweiterungskarten speziell für den A3000 werden auch auf dem Stand der High-Tech-Firma **X-Pert** zu sehen sein. Beachtenswert sind hier besonders eine RAM-Karte mit bis zu 32 (!) MByte Nibblemode-RAM und eine Cache-Karte für den Prozessorslot, die auch einen mit 50 MHz getakteten 68882 aufnehmen kann. Mindestens ebenso interessant ist jedoch die Grafikkarte, die mit bis zu 4 MByte (A3000: 64

MByte) speziellen Video-RAMs vollkommen neue Anwendungsgebiete erschließt. Diese können vor allem mit den Transputerkarten für alle Amiga-Modelle ausgenutzt werden. Die Grafikkarte beinhaltet eine Einzelbildansteuerung für Videorecorder und kann auf Auflösungen bis zu 1600 x 1280 Pixeln frei programmiert werden. Vor allem die Tatsache, daß es sich hier um eine echte Grafikkarte, nicht jedoch um einen Framebuffer handelt, macht das Gerät einzigartig auf dem Amiga-Markt.

Erweiterungskarten

Yellow Computing hat sich auf die Verbindung kleinster Rechner mit dem Amiga spezialisiert. Kombinationen aus Hard- und Software ermöglichen mittlerweile die Datenübertragung zwischen den wichtigsten auf dem Markt befindlichen Pocket-Computern. Dazu zählt die SF-Serie von Casio, der IQ von Sharp, die gesamte SHARP-PCxxxx-Serie sowie einige andere Modelle.

Die interessantesten und innovativsten Neuvorstellungen kommen sicherlich aus dem Video-Bereich. Doch auch im Bereich DTP wird der Amiga immer leistungsfähiger. Aber auch die Neuheiten auf allen anderen Gebieten werden sicher helfen, das Image des Amiga als reine »Spielmaschine« zurückzurücken. Es gibt viele professionelle Anwendungen, die die Qualitäten dieses Computers deutlich unterstreichen. Wir wünschen Ihnen einen interessanten und möglichst streßfreien Messeaufenthalt. Besuchen Sie uns doch einmal am Stand des DMV-Verlages, Halle 12, Stand 307 oder bei den Veranstaltungen unseres Forums.

(cd)

Berichtigung

Das in Ausgabe 10 '90 vorgestellte **KCS Power Board** ist nicht, wie von uns dargestellt für 510,- DM, sondern für 798,- DM erhältlich. Das Gerät wird übrigens nicht ausschließlich von der Firma Roßmüller, sondern auch von der Firma Miky Wenn-gatz vertrieben.

Ute Bahn/Manfred J. Heinze

Profi-Amiga mit 16 Millionen Farben

Kooperierende Prozessoren, sogenannte Transputer, sind das neueste Konzept in der elektronischen Bildverarbeitung.

Was Rechen-
einheiten mit
Teamgeist leisten, zeigt
eine der anspruchsvoll-
sten Anwendungen: 3D-
Animation mit 16 Millio-
nen Farben.

Der gute alte Amiga –
immerhin seit rund fünf
Jahren bewährt er sich
als „Mediencomputer“
nicht nur bei den zahlrei-
chen Heimanwendern,
sondern auch in der pro-
fessionellen Videopro-
duktion als Titelgenera-
tor, Grafikcomputer und
Animations-Workstation.
Vom Ur-Amiga 1000 bis
zum High-Tech-Boliden
A-3000 führt eine konti-
nuierliche Weiterent-
wicklung in Richtung auf
ein solides, den hohen
Qualitätsansprüchen
des Medienprofis genü-
gendes Gerät. Die Work-
bench und das Filesy-
stem wurden verbessert,
Peripheriegeräte wie
Wechselselfplatten und
leistungsfähige Scanner
kamen hinzu. Die für
Studiozwecke geeignete
Software wie »Sculpt-An-
imate 4D« und »DPaint«
kann sich nicht nur se-

hen lassen: Selbst An-
wender von Spezial-Gra-
fikgeräten wie „Paint-
box“ und „Harry“ billigen
den Programmen eine
hohe Leistungsfähigkeit
zu.

Das Image des „Spiele-
computers“ Amiga sollte
also ein für allemal der
Vergangenheit angehö-
ren – wenn nicht noch
ein letztes Hindernis auf
dem Weg zur echten
Profimaschine im Wege
gestanden hätte. Alle
Verbesserungen konn-
ten bisher nicht ausglei-
chen, daß nach wie vor
höchstens 4096 Farben
zur Verfügung standen
– der berühmt-berüch-
tigte HAM-Modus also.

Bei der professionellen
Computergrafik und
-animation, wie man sie
aus den Senderkennun-
gen und Magazinvor-
spännen des Fernse-
hens kennt, geht es um
Millionen von Farbabstu-
fungen – genau gesagt,
mindestens um 16,7 Mil-

Leistungsfähigkeit beruht allein auf der Vernetzung jeder einzelnen Nervenzelle mit etwa tausend anderen. Zehn Milliarden Neuronen stürzen sich in einem solchen Gehirn-Netzwerk parallel auf ein einzelnes Problem, wobei jedes Neuron völlig gleichberechtigt ist und selbständig Entscheidungen trifft. Diese vernetzte neuronale Struktur führt zu der unerhörten Leistungsfähigkeit

ter diese Zeit für seine eigentliche Aufgabe, nämlich zum Rechnen. Damit ist bereits der einzelne Transputer gegenüber den herkömmlichen Prozessoren aufgrund seiner einfachen Struktur und eines reduzierten Befehlsumfanges – der sogenannten RISC-Technik – um einiges schneller. Der eigentliche Fortschritt neben dieser Integration liegt in einer ebenfalls im Chip selbst

blem benötigt werden. Wenn die bisherige Rechnerarchitektur der klassischen Handwerksarbeit entsprach, dann kann man die Parallelarchitektur mit einer modernen, arbeitsteiligen Industriegesellschaft vergleichen.

Mit dieser Technologie ist jetzt endlich eine geeignete 3D-Animations- und Rendersoftware für den professionellen Anwender greifbar. Zur Photokina stellt die Frankfurter ArtTec GmbH »Miranim« und »Sabrina« vor, zusammen ein komplettes 3D-Konstruktions- und Animationspaket mit 16 Millionen Farben. Dabei konnte auf ein bereits bewährtes System zurückgegriffen werden, das an der Universität Montreal von dem Professoren-Ehepaar Nadia und Daniel Thalmann auf einem VAX-Computer programmiert wurde. ArtTec gelang es, die Features dieses interessanten Paketes in die parallelisierte Version zu übernehmen. »Miranim« ist ein algorithmisches Animationsprogramm, das Objekte mit Hilfe mathematischer Formeln animieren kann. Bewegungsgesetze, wie lineare, oszillierende, explodierende oder Brownsche Bewegungen, werden einer Animationsvariablen zugeordnet.

Um zum Beispiel das Verhalten eines springenden Balles oder einer sich dehnen

Was ist ein Transputer überhaupt?

higkeit des menschlichen Gehirns. Kein Wunder, daß das Gehirn als bisher unerreichtes Vorbild für die Forschungstätigkeit auf dem Gebiet der Rechnerarchitektur gilt. Weltweites Ziel ist der Bau eines derartigen nervenähnlichen, eben neuronalen Netzes im Computer.

Einer der Entwicklungswege auf dem Weg zum neuronalen Computer ist das Transputer-Konzept. Ein Transputer ist zunächst ein Hochleistungschip, genau wie die bekannten Intel 80386 oder Motorola 68030 mit ihren mathematischen Coprozessoren. Zwischen dem Prozessor und seinem Mathe-Co entstehen natürlich Unmengen von Übertragungsprozessen, ebenso wie zwischen dem Prozessorpaar und dem Speicher.

Ein Transputer führt alle diese Operationen auf einem einzigen Chip aus. Er stellt also einen kompletten Mikrocomputer dar, mit allen Teilen, die bei der klassischen Architektur nebeneinander existieren: Ein einzelner Chip enthält die Recheneinheit mit integriertem Coprozessor und eigenem schnellen Zwischenspeicher für Daten und Befehle. Statt hauptsächlich Daten zwischen der Recheneinheit, dem Coprozessor und den Speicherbausteinen hin und her zu transportieren, verwendet der Transpu-

installierten Steuereinheit mit vier schnellen Kommunikationsverbindungen zu weiteren Transputern, den Links. Alle diese Teile erfüllen die Forderung nach unabhängiger paralleler Verarbeitung. Der Transputer kann also Daten, das heißt Integer- und Real-Zahlen, parallel verarbeiten, während er gleichzeitig mit anderen Transputern über die Links kommuniziert.

Damit wird ein verteiltes System möglich, in dem die Transputer unterschiedliche Aufgaben ausführen und – wenn nötig – miteinander kommunizieren und zusammenarbeiten. Beliebige viele Transputer können verkoppelt und so angeordnet werden, wie sie für ein spezielles Pro-

was mit der Wirklichkeit zu tun hatte. Tatsächlich besteht ein Rechner aus lediglich zwei Teilen, dem Prozessor und dem Speicher. Jede Stubenfliege besitzt eine ungleich komplexere Gehirnstruktur. Sämtliche Daten müssen aus dem Speicher durch den Engpass des Prozessors geschleust werden, der alle Daten nacheinander zu verarbeiten hat. Allerdings verbringt der Prozessor die meiste Zeit damit, auf das Eintreffen dieser Daten zu warten. Die bisherige Rechnerarchitektur gleicht – bildlich gesprochen – klassischer Handwerksarbeit: Jedes Problem wird in Form eines Programmes Schritt für Schritt von einem Einzelnen bearbeitet. Wie bei einer einspurigen Autobahnbaustelle kommt es dabei an jedem Engpaß zum unerfreulichen Stau. Das Hindernis ist der Prozessor selbst, durch den alle Datenströme geschleust werden. Diese sequentielle, also nacheinander erfolgende Datenverarbeitung schränkt die Leistungsfähigkeit der konventionellen Computer unweigerlich ein, so schnell der einzelne Prozessor selbst auch sein mag.

Organische »Prozessoren«, menschliche Gehirnzellen oder Neuronen, sind erstaunlicherweise wesentlich langsamer als moderne Rechenwerke. Die Fähigkeit solcher Einzelzellen, unabhängig voneinander und dennoch gemeinsam ein Problem in Angriff nehmen zu können, läßt ein menschliches Gehirn derart effektiv arbeiten – jedenfalls meistens. Diese

lionen. Diese Farbpalette und eine möglichst hohe Bildauflösung sind die Kriterien für eine kommerzielle Anwendung – und da konnte der Amiga bisher nicht mithalten. Bisher – denn jetzt hat sich die Situation schlagartig geändert. Relativ unauffällig hat eine neue Technologie das »color gap« zwischen dem Amiga und der Profianwendung geschlossen.

Man nehme einen beliebigen Amiga 2000 oder 3000 und setze eine Transputer-Grafikkarte sowie einen speziellen Adapter ein, schließe einen Multisync-Monitor an und staune über die besagten 16 Millionen Farben!

In den Anfangstagen der Computertechnik nannte man die Rechner oft euphorisch »Elektronengehirne«, eine griffige Bezeichnung, die nichtsdestoweniger kaum et-

Mini-Glossar:

- Keyframes: Hauptphasen, zwischen denen die Bewegung eines Objektes interpoliert, das heißt vom Computer berechnet wird. Hauptphasen sind markante Eckpunkte einer Bewegung.
- Solid texturing: Geht über das »texture mapping« hinaus und zeigt beispielsweise beim Schnitt durch einen massiven Körper auch an den Schnittflächen die typische Materialstruktur
- Splines: Weiche Kurven, die Punkte verbinden. Im traditionellen Zeichnerhandwerk wurden Konstruktionspunkte durch einen biegsamen Blechstreifen, den »spline«, miteinander verbunden, so daß sie auf einer gemeinsamen Kurve lagen. »spline editing« ist die Verschiebung von Punkten, ohne daß der geschwungene Kurvenverlauf beeinträchtigt wird.
- Texture mapping: Verfahren, mit dem ein in einem 3D-Programm erzeugtes Objekt mit einem IFF-Bild überzogen wird

Feder zu simulieren, genügt die Angabe der entsprechenden Formel. Normalerweise muß der Animation Designer sogenannte Keyframes festlegen, Hauptphasen also, nach denen der Computer Zwischenwerte berechnet. Die zusätzliche algorithmische Eingabe erspart natürlich erhebliche Test- und Probeläufe. »Miranim« ist zudem frei programmierbar, so daß der Animation Designer das System entsprechend seiner speziellen Anforderungen erweitern kann. Möglicherweise kann – nach einer erfolgreichen Anpassung – sogar eine spektakuläre Software namens »Human Factory« eingesetzt werden, mit der die bekannten Marilyn-und-Humphrey-Clips realisiert wurden. Mit »Human Factory« werden komplette menschliche Bewegungsabläufe und Mimiken simuliert, so daß – durch die Eingabe einer komplexen Formel – eine Figur sehr realistisch zu gehen oder ein computergeneriertes Gesicht zu lachen beginnt. »Sabrina« heißt der dazugehörige Modeler, mit dessen Hilfe dreidimensionale Objekte erzeugt und modelliert werden. Recht interessante Features wie solid texturing, texture mapping, vier Kameras, ein reichhaltiges spline editing, Reflektions- und Transparenzmodi ergeben ein funktionelles, mit vernünftigen kreativen Werkzeugen versehenes Programm. Die Programme liegen zur Zeit in einer Feldtest-Version vor, die bereits

alle Funktionen enthält: Hier zählt sich der Rückgriff auf eine bewährte Software aus. Die Software läuft auf einer Sang-Transputerkarte Mega-Link02 mit 2 MB Video RAM und mindestens 4 MB Hauptspeicher. Als Betriebssystem dient HELIOS, ein spezielles Transputer-Operating System. Durch dieses externe, nur den Transputern vorbehaltene Betriebssystem kann das Soft- und Hardware-Ensemble auf nahezu allen gängigen PCs eingesetzt werden, also auf jedem IBM-kompatiblen PC, einer Sun386i oder auch auf einem Amiga. Auf dem Amiga wird lediglich eine kleine Adapterkarte zusammen mit dem Transputerboard eingesetzt. Genaugenommen enthält die Sang-Transputerkarte also einen kompletten Grafikcomputer mit eigenem Betriebssystem. Der Amiga wird eigentlich nur zum Betrieb der Festplatten und zur Stromversorgung benötigt. Sang hat den T-800-Transputer mit einem Immos G300 Videocontroller-Chip mit bis 2 MByte Bildwiederholungspeicher ausgestattet, der entweder 16 Millionen Farben gleichzeitig darstellt oder 256 aus 16 Millionen bei einer Auflösung von 1280 * 1024 Bildpunkten. Die volle Farbpalette von 16 Millionen reicht immer noch für eine videogerechte Auflösung von etwa 800 * 600 Pixeln. Auch bei der erreichbaren Höchstauflösung spielen die Transputer ihre systemimmanenten Vorteile aus: drei Mega-Link02-Boards hinterein-

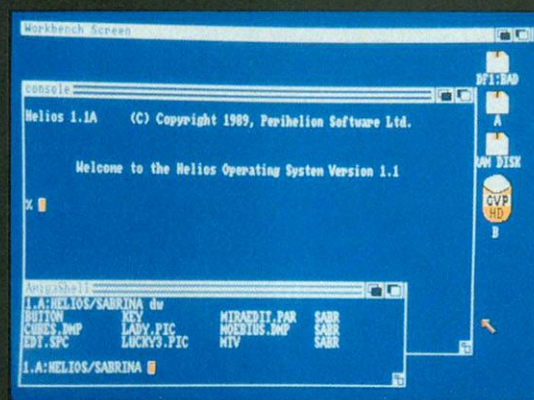


Bild 1: Das HELIOS-Betriebssystem läuft als eigene Task auf dem Amiga

andergeschaltet ergeben zum Beispiel eine Auflösung von 1024 * 1024 Pixeln in 16 Millionen Farben.

Weitere Transputer können einfach mit einer Kabelverbindung dazugeschaltet werden. Und dann spielt der Transputerrechner seine Vorteile voll aus: Statt ein Einzelbild sequentiell, Pixel für Pixel oder Zeile für Zeile zu berechnen und darzustellen, arbeiten alle Transputer gemeinsam an der Generierung. Das Bild wird, je nach Anzahl der eingesetzten Trans-

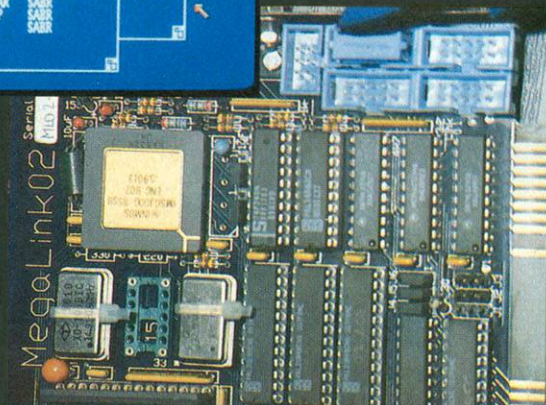


Bild 2: Im Detail: Golden der Grafik-Controller, blau die Links

puter, simultan in mehreren Bereichen auf dem Bildschirm aufgebaut. Da nicht alle Bereiche des Bildes gleich schwierig zu berechnen sind, können Transputer, die mit ihrer Arbeit fertig sind, ihren Kollegen bei der Darstellung der rechenintensiveren Teile helfen. Der zu erzielende Geschwindigkeitszuwachs kann sicher nur im Einzelvergleich anhand eines Polygonidentischen Objektes genau beziffert werden, aufgrund des Konzeptes und der Schnelligkeit der einzelnen Transputer reduzieren sich die Rechenzeiten, im Vergleich zu einem konventionellen Grafikboard jedoch entscheidend.

Artec, Sang und der Vertreiber Videocomp sind zuversichtlich, bis zur Photokina im Oktober ein komplettes, einsatzfähiges System bereitstellen zu können, das alle Ein- und Ausgabereinheiten wie Scanner und Videoausgang umfasst.

Wenn die Soft- und Hardware zur Photokina verfügbar ist, kann man sicher von einer Revolution in der Amiga-Animation und -Grafik ausgehen. Mit der Mischung aus hochmodernem Konzept und bewährten Programmen hat der Amiga endlich alle Chancen, den Durchbruch ins professionelle Grafiklager zu schaffen. (mm)

AMIGA DOS Info

Hochleistungs-Transputer-Grafik-System MEGA-Link02

- Steckkarte mit T800-Transputer
- 1,2,4 oder 8 MByte Hauptspeicher
- Auflösung bis 1280x1024 Pixel frei programmierbar
- 16,7 Mio. oder 256 aus 16,7 Mio. Farben
- Immos G300 Color Video Controller
- 1 oder 2 MByte spezielles Video-RAM (VRAM)
- 10 bis 110 MHz Pixelfrequenz
- RAM-Bestückung: 1 MByte DRAM, 1 MByte VRAM bis 8 MByte DRAM, 2 MByte VRAM

Preis: zwischen DM 6726,00 und DM 14250,00 je nach RAM-Bestückung

3D-Grafik- und Visualisierungsprogramm Sabrina

- 16,7 Mio. Farben
- Design und Modeling von 3D-Objekten
- grafischer 2D-3D-Input
- Fünf Shading-Modelle: Constant, Lambert, Gouraud, Phong, Interphong
- Multiple Kameras
- Transparenz- und Image-Mapping
- Solid Texturing
- Scanline- und Wireframe-Ausgabe

Preis: DM 2998,00

3D-Animationsprogramm Miranim

- 16,7 Mio. Farben

Preis: DM 17000,00

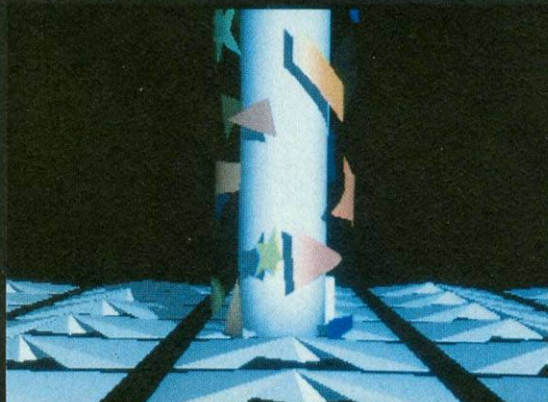


Bild 3: »Polygone-Mapping« mit 16 Mio. Farben erzeugt Raumillusion

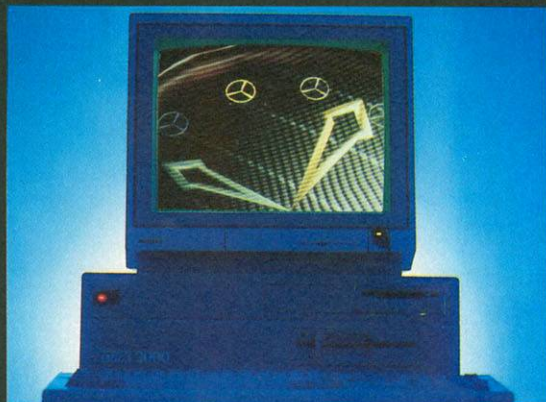


Bild 4: Professionelles Arbeiten mit dem Amiga

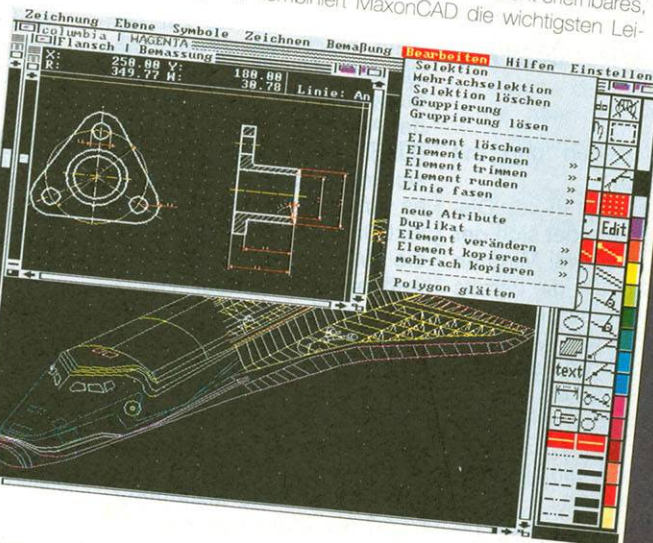
MaxonCAD.

Die neue Dimension in Design und Konstruktion.

MaxonCAD ist das leistungsstarke CAD-Programm für alle Anwender vom Hobby-Designer bis hin zum professionellen Konstrukteur. Seine transparente und flexible Gestaltung macht es offen für jede Art von Anwendung, von der mechanischen Konstruktion bis hin zur Architektur.

Integriert in eine komfortable Benutzeroberfläche, die durch ihre Geradlinigkeit leicht erlernbares, komfortables und effizientes Arbeiten ermöglicht, kombiniert MaxonCAD die wichtigsten Leistungsmerkmale, die für den Anwender von Bedeutung sind:

- Sehr hohe Arbeitsgeschwindigkeit und schneller Bildaufbau.
- Umfangreiche Funktionen für Konstruktion, Nachbearbeitung und Bemessung.
- Leistungsstarke Bibliotheksfunktionen.
- Hochwertige Drucker- und Plotterausgabe sowie Kompatibilität zu bestehenden Format-Standards (DXF, HPGL).



MaxonCAD.

Die Synthese aus Bedienungskomfort, Geschwindigkeit und Leistungsvielfalt.

Geeignet für alle AMIGA mit mindestens 1 MByte Speicher.
Lieferbar in zwei Versionen: MaxonCAD DM 449,-. MaxonCAD Student DM 249,-.

L100

Platinenlayout

Leistungsfähiges Layoutprogramm für doppelseitige Platinen • Platinengröße bis 257*257 oder 346*192 mm (mit 512K noch Euroformat!) • Auflösung 1/100 Zoll • Druckerausgabe bis 400*400 dpi • HPGL-kompatible Plotteransteuerung • Erweiterbare Symbolbibliothek • Blitzschnelles Scrolling durch Super-bitmap-Technik • Einstellbares Raster • Lupe

DM 99,-

unverbindliche Preisempfehlung

Der endgültige Virenschutz

VIRUSCOPE

Schützen Sie sich und Ihre Daten vor digitalen Infektionen! VIRUSCOPE 1.1 erkennt über 70 verschiedene Viren (darunter 8 Linkviren), arbeitet auf Wunsch im Hintergrund und überprüft jeden Datenträger automatisch. Dazu bietet es umfangreiche Bootblocktools, wie z.B. Multiutility mit Screenblanker, Bootblock Menu Maker oder Bootblock-Editor.

DM 59,-

unverbindliche Preisempfehlung

FACE THE MUSIC

Das ultimative Soundprogramm

- Achtstimmiger Stereoklang mit voller Kontrolle über Lautstärke ohne Verminderung der Klangqualität
- Voll multitaskingfähig
- Vielfältige Effektmöglichkeiten
- Stücke können eingegeben, über Tastatur oder über ein MIDI-Keyboar eingespielt werden
- Komfortabler Editor
- Umfangreiche Sammlung digitalisierter Sounds und Musikstücke.

DM 99,-

unverbindliche Preisempfehlung

CHAMÄLEON

Der ST-Emulator ohne Hardware

CHAMÄLEON ist der leistungsfähige ATARI ST-Emulator für alle AMIGA 500, 1000 und 2000, der Ihnen für wenig Geld eine breite Palette von Anwendersoftware für den ATARI ST erschließt • Unterstützt alle Auflösungen • Verschiedene Diskettenformate • Parallelportemulation • Benötigt keinen Hardware-Zusatz • Nahezu volle ST-Geschwindigkeit • Läuft mit jeder TOS-Version

DM 99,-

unverbindliche Preisempfehlung



KICK Pascal

KICK-PASCAL ist ein leistungsfähiges vollintegriertes Entwicklungssystem für den AMIGA • Direktes Compilieren und Linken aus dem komfortablen Fullscreen-Editor • Extrem kurze Übersetzungszeiten (20.000 Zeilen/Minute) • Komplet in Assembler geschrieben • Komfortable Fehlerbehandlung • Stark erweiterter Befehlsumfang • Leichter Zugriff auf alle AMIGA-Funktionen • Alle AMIGA Include-Dateien • Viele umfangreiche Beispiele und Routinen

DM 189,-

unverbindliche Preisempfehlung

Weitere aktuelle AMIGA-Produkte von MAXON:

- GASAL • Graphics And Sound Animation Language DM 99,-
- LAYOUT! • Graphische Gestaltung in hoher Qualität DM 59,-
- KICKTOOLS • Komplexes CLI-Tool mit Zusatzfunktionen DM 99,-
- HD-Backup • Komfortables Festplatten-Backup-Programm DM 99,-
- KICK ASS • Leistungsfähiger 68000-Assembler DM 89,-
- R.C.T. • Requester Construction Set für alle Sprachen DM 129,-
- Junior Prommer • EPROM-Programmiergerät DM 249,-

Gerne senden wir Ihnen nähere Informationen zu unseren Produkten.

Alle Preise sind unverbindliche empfohlene Verkaufspreise

Besuchen Sie uns auf der
AMIGA 90 KÖLN
09.-11. November 1990
Halle 10, Stand 225

MAXON
computer gmbh
Schwalbacher Str. 52b
D-6236 Eschborn
Tel. 06196/481811
FAX 06196/41885

Ingmar Reyer

Simply the best?

Nach dem großen Erfolg kommt mit »BECKERtext II Amiga« ein bewährtes Produkt im neuen Outfit auf den Markt.

BECKERtext II Amiga wird als Software im Buch vermarktet. Wo andere Programme mit großen stabilen Verpackungen protzen, präsentiert sich das Programm im schlichten Folien-Look. Darunter befindet sich das zirka 500 Seiten starke Handbuch und zwei Programmdisketten. Die erste Diskette enthält WorkBench, Beispieltex-te und einige IFF-Bilder. Auf der zweiten Diskette ist das eigentliche Textprogramm, das schon von Textomat her bekannte Tool »BT-SNAP« und ein Harddisk-Installationsprogramm zu finden. Beim ersten Programmstart verlangt »BECKERtext II« eine Namens-Installation. Das bedeutet, der Anwender muß seine Anschrift und die Seriennummer in einem Requester eingeben. Diese Daten werden im Programm vermerkt und dienen dort als Identifikation.

Aller Anfang ist schwer

Um sich vor ungewollten Effekten zu schützen, sollte man sich Sicherheitskopien des

Programms ziehen und mit diesen arbeiten.

Wird »BECKERtext II« von Diskette geladen, meldet sich der Amiga nach gut zwei Minuten wieder. Diejenigen, die bereits das ältere Programm kannten, werden die starken Veränderungen im Outfit und den Menüs sofort bemerken. Nichts sieht mehr so aus, wie beim guten alten »BECKERtext«. Aber was gut war, kann ja auch noch besser werden.

Ein blinkender Cursor lockt direkt nach dem Start des Programms zur Texteingabe. Nach den ersten Buchstaben fällt dem Anwender der recht große, aber gut lesbare Zeichensatz auf. Der Anwender kann vom normalen Edit-Screen auf einen Preview-Screen umschalten, auf dem die gesamte Seite im Überblick sichtbar wird. Will man sich einen der mitgelieferten Beispiel- oder Übungstexte ansehen, fährt man mit der Maus ins Menü und wählt den Menüpunkt »Laden« aus. Das hat zur Folge, daß sich ein großes Dateiauswahlfenster öffnet. Dieses Fenster birgt einige Besonderheiten. Der Anwender kann sich zum Beispiel die Da-

teien, die aufgelistet werden sollen, nach Größe, alphabetisch oder nach Datum sortieren lassen. Er kann Dateien mit besonderen Endungen ausmaskieren und vieles mehr. Besondere Erwähnung verdient die integrierte Hilfsfunktion. Wie in fast allen anderen Fenstern, die zur Auswahl bestimmter Funktionen dienen, stellt auch das Dateiauswahlfenster ein Gadget mit der Aufschrift »Hilfe« zur Verfügung. Per Mausklick öffnet sich ein weiteres Fenster und es wird ein ausführlicher Hilfstext über die Funktionen und die Möglichkeiten des Auswahlfensters sichtbar. Der zu ladende Text kann über verschiedene Varianten aufgerufen werden. Die schnellste ist es, den Textnamen mit einem Doppelklick per Maus zu laden. Wird der Text am Bildschirm angezeigt, sticht sofort ins Auge, daß alle Textattribute sofort sichtbar sind. Für eine weitere angenehme Überraschung sorgt »BECKERtext II«, wenn per Cursor durch den Text gescrollt wird. Obwohl das WYSIWYG-Prinzip konsequent durchgehalten wird, ist die Geschwindigkeit immer zufriedenstellend.

Bilder – das Salz des Textes

Grafikmaterial wie Bilder in einen Text zu integrieren, bieten viele Textverarbeitungsprogramme für den Amiga an. Aber keins kann in dieser Disziplin mit »BECKERtext II« mithalten. IFF-Bilder können an jeder beliebigen Stelle im Text eingefügt werden. Die Grafiken können nachträglich verschoben, gestreckt oder gestaucht werden. Der eindeutige Vorteil gegenüber anderen Programmen liegt darin, daß Grafiken von Text rechts oder links umflossen werden können. Außerdem werden die Grafiken beim Ausdruck nicht in häßliche, grobe Raster zerlegt, wie dies beim Vorgänger der Fall war. Die Bilder werden in höchster Auflösung zu Papier gebracht. Die Druckersteuerung ist ein kleines Meisterwerk. Das Programm stellt fest, welcher Drucker in den Preferences ausgewählt worden ist und stellt nach diesen Informationen den Druckertreiber für »BECKERtext II« selbst-

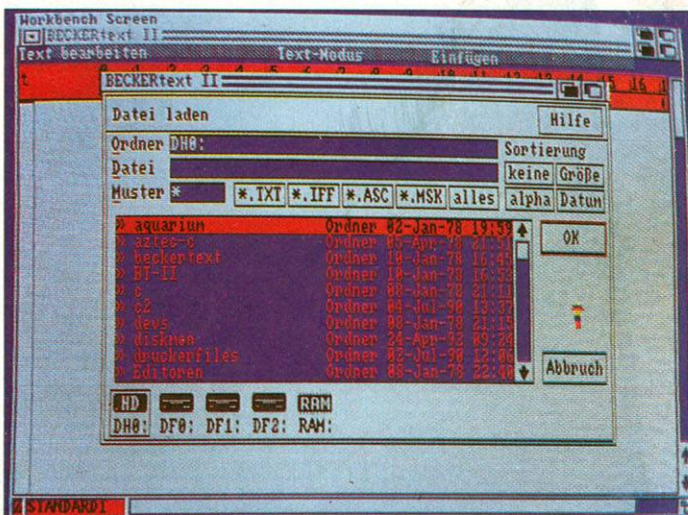
tätig ein. Hinzu kommt noch, daß selbst die Smoothing-Funktion aus dem Preferences mitbenutzt wird. Alle Schriftarten, die mit dem angeschlossenen Drucker möglich sind, können vom Programm aus über ein Auswahlfenster eingestellt werden. Sie werden auch grafisch dargestellt, so daß man sich ein Bild von seiner Zeichensatzwahl machen kann.

Weiterhin fällt positiv auf, daß der Text nach Belieben im Grafik-Modus oder im Textmodus ausgegeben werden kann. Im Grafikmodus kann der Anwender einen beliebigen Zeichensatz von Diskette nachladen (beispielsweise Opal, Ruby oder ähnliches). Im Text-Modus werden nur im Text vorhandene Bilder als Grafik auf dem Drucker ausgegeben. Text bleibt also Text und wird, auch wenn er ein Bild umfließt, als solcher schnell und in bester Qualität ausgedruckt.

Professionelle Anwendungen

Kommt man in den Genuß, eine wissenschaftliche Arbeit verfassen zu müssen, sah es bisher mit geeigneten Hilfsmitteln für den Amiga recht schlecht aus. »BECKERtext II« stellt für solche Aufgaben das beste Werkzeug dar. Kinderleicht können so dringend benötigte Dinge wie Fuß- und Endnoten sowie Indizes in einen Text eingefügt werden. Außerdem ist der integrierte Gliederungs-Editor ein tolles Tool, um den Überblick in seinen Texten zu behalten. Wer viel in seinen Texten rechnet, der ist mit diesem Textprogramm ebenfalls gut bedient. Im Untermenü »Berechnung« steht dem Anwender ein Taschenrechner zur Verfügung. Mit ihm sind die vier Grundrechenarten, sowie Prozentrechnung möglich. Hat man ein im Text benötigtes Ergebnis auf dem Taschenrechner-Display vorliegen, dann braucht man nur das Gadget »in Text« mit der Maus anzuklicken und schon steht die berechnete Zahl im Text. Eine wirklich tolle Funktion.

Durchforstet man die Menüs nach weiteren ausgefallenen Features, dann stößt man sicherlich auch auf die folgende: »ARexx«. Man kann sie sozu-



Das Dateiauswahlfenster zeigt sich äußerst flexibel

sagen als eine leicht zu erlernende Programmiersprache bezeichnen. Ihre Hauptaufgabe ist es, mit anderen Programmen zusammenzuarbeiten. Die Voraussetzung dafür ist aber, daß die Programme über eine AREXX-Schnittstelle verfügen. In »BECKERtext II« ist diese Schnittstelle bereits implementiert. Darüber können von »ARexx« aus, BECKERtext-Befehle an das Hauptprogramm selbst geschickt werden. Diese Befehle wirken dann wie Eingaben über Tastatur oder Maus. »ARexx« stellt in diesem Fall eine Art Fernsteuerung dar.

Rechtschreibung – nicht jeder- manns Sache

»BECKERtext« besaß bereits eine recht ordentliche Rechtschreibprüfung. Die neue Version hat selbst keine Lexikonfunktion integriert. Es steht lediglich ein Software-Interface zum Rechtschreibprüfungsprogramm »DATA BECKERS Rechtschreibprofi Amiga« zur Verfügung. Wer also seine Texte vom Computer korrigiert haben möchte, muß neben dem Textprogramm ein zweites kaufen. Hier wird Data Becker nicht ganz dem Anspruch gerecht, daß »BECKERtext II« ein professionelles Textprogramm sei. Zu einer Textverarbeitung für Profis gehört »serienmäßig« ein gutes Lexikon.

Ein anderes Anwendungsgebiet für Textprogramme beherrscht »BECKERtext II« wieder sehr gut. Es handelt sich dabei um die Serienbrief-Funktion. Damit bietet das Programm die Möglichkeit, mehrere gleichlautende Briefe an verschiedene Empfänger zu senden. Die Anschriften der Empfänger bezieht das Textverarbeitungsprogramm dabei entweder aus zuvor in den laufenden Text eingegebenen Daten, oder – was viel bequemer ist – aus dem Dateiprogramm »Datamat« vom gleichen Softwarehaus.

Flexibilität, ist ein besonderes Merkmal bei »BECKERtext II«. So findet der Anwender in der Menüleiste einen Eintrag, um die Tastaturbelegung zu verändern. Jede Taste kann hier neu belegt werden. Es ist zum Beispiel möglich, mit der Tastenkombination [Ctrl-Alt-A] den

Namen seiner Freundin auf den Bildschirm zu zaubern, oder natürlich auch den seines Freundes. Ebenso können mit diesem Tool aber auch alle Tastaturkommandos dem persönlichen Geschmack angepaßt werden. Etwas übertrieben wurde die Flexibilität bei der Menügestaltung. Die Menüleiste läßt sich nach Lust und Laune verändern. Wer also lieber das Menü »Projekt« ganz rechts haben möchte, anstatt links, der hat mit dieser Funktion Gelegenheit dazu. Weit aus sinnvoller ist das folgende Feature. »BECKERtext II« kann auf drei verschiedene Arten bedient werden. Der Anfänger wird mit der Bedienung per Maus am besten zurechtkommen. Dem Fortgeschrittenen ist anzuraten, die zweite Methode anzuwenden, nämlich die Steuerung des Programms über Befehle. Alle Programmfunktionen können mit Hilfe von Befehlen gesteuert werden. Die letzte Möglichkeit, um Funktionen aufzurufen, besteht darin, Tastenkombinationen zu drücken. Dies ist eine aus der MS-DOS-Welt bekannte und bewährte Art, ein Programm schnellstmöglich zu bedienen.

(Nachzu)schla- gende Argumente

Werfen wir zum Schluß noch ein Blick auf das Handbuch. Es besteht aus vier Teilen und hat einen Umfang von rund 500 Seiten. Der erste Teil beschreibt die Grundlagen des Programms. Es wird auf die Installation auf Festplatten/Disketten eingegangen. In einem gesonderten Einsteigerkurs werden die Grundfunktionen des Programms beschrieben. Der zweite Teil geht mehr auf die Praxis ein. Dort werden alle Menüs und Auswahlfenster ausführlich und gut beschrieben. Leider haben sich aber in diesen Teil auch kleinere Fehler eingeschlichen, die einen Anfänger doch recht stark verwirren können. So wird auf Seite 184 die Markierung von Wörtern und Textzeilen beschrieben. Folgt man den Angaben des Handbuchs, und versucht zum Beispiel mit der Tastenkombination [Shift-F5] ein Wort rechts vom Cursor zu markieren, dann stellt man schnell fest, daß dies keine Re-

aktion hervorruft. Die richtige Taste für diese Funktion ist [F5]. Dies ist leider kein Einzelfall. Auch ein paar andere Tastaturkommandos funktionieren nicht so, wie im Handbuch beschrieben. Die Autoren sollten also möglichst schnell die Fehler, die sich in den Text eingeschlichen haben ausmerzen. Aber nun weiter im Text. Der dritte Teil stellt eine Übersicht über die Befehlswörter zur Verfügung. Es wird sehr ausführlich auf jeden einzelnen Befehl eingegangen. Am Ende dieses Teils wird noch kurz auf die Zusammenarbeit zwischen »BECKERtext II« und der Interpretersprache »ARexx« eingegangen. Ein übersichtlicher Anhang schließt sich an. Hier werden zum Beispiel nützliche Tips über Druckeranpassungen vermittelt und Hinweise gegeben, wie man den Speicher des Rechners optimal ausnutzt.

»BECKERtext II« und A3000

Der erste Eindruck nach dem Test mit dem A3000 läßt die Vermutung aufkommen, daß der entsprechende Praxistest von jemandem vorgenommen wurde, der die Möglichkeiten des 3000ers nicht sehr gut kennt. So stürzt der Rechner aus schier unerklärlichen Gründen während der Arbeit mit dem Programm aus heiterem Himmel ab. Wenn man den Textspeicher löschen möchte, wird ein zweites Fenster aufgemacht. Das Lexikon läßt sich nicht laden.

Daneben gibt es die schöne Möglichkeit Fonts einzustellen. Nur hat man von der automatischen Fensteraktivierung mittels »Autopoint«, die beim A3000 in der WB2.x vorhanden ist, wohl nie etwas gehört? Wie sonst könnte es sein, daß sobald der Auswahl-Requester aktiviert wird, das Fenster plötzlich nur noch zur Hälfte da ist? Mit viel Glück gelingt es, einen Font per Maus anzuklicken, da im Auswahlfenster die Anzeige der verfügbaren Fonts einfach übereinander geschoben werden.

Die abschließende Beurteilung von »BECKERtext II« entpuppt sich als zweischneidiges Schwert. Auf der einen Seite kann das Programm durch seine vielfältigen Funk-

tionen und Features überzeugen. Leichte Bedienung, WYSIWIG, Fuß- und Endnotenverwaltung, um nur einige positive Aspekte zu nennen. Auf der anderen Seite ist aber auch festzuhalten, daß die vielen Optionen, die dieses Programm mitbringt, auch einen großen Nachteil mit sich ziehen: »BECKERtext II« entpuppt sich als wahrer Speicherfresser. Wer versucht, mit 1 MByte RAM und zwei Disk-Laufwerken einen längeren Text (ca. 20 Seiten), der mit ein paar kleinen Grafiken aufgelockert werden soll, zu verfassen, der sieht den Guru leider recht häufig. Es fällt auf, daß das Programm bei gut gefülltem Textspeicher den Guru nicht immer sicher abfangen kann. Daneben sind noch die diversen Probleme anzuführen, die »BECKERtext II« mit dem A3000 hat.

Für 298,- DM erhält der Käufer, sofern er genug Speicher besitzt, ein für den Amiga-Bereich geeignetes Textprogramm. Das Konzept hat sicher einen neuen Standard im Bereich der Amiga-Textverarbeitungen gesetzt, die erwähnten Mängel jedoch sollten von den Programmautoren beseitigt werden.

(vb)



AMIGA DOS

Blitz ☆ licht

Name: BECKERtext II
Hersteller: Data Becker
Preis: 298,- DM
Mindestspeicher: 1 MByte

Positiv:

- leichte Handhabung
- gutes Handbuch
- konsequentes WYSIWIG
- Geschwindigkeit
- Text fließt um Grafiken
- verfügt über AREXX-Schnittstelle

Negativ:

- kleinere Mängel im Handbuch
- kein Lexikon, der Anwender muß sich ein zusätzliches Programm kaufen
- bei Speichermangel erfolgt Absturz

**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.

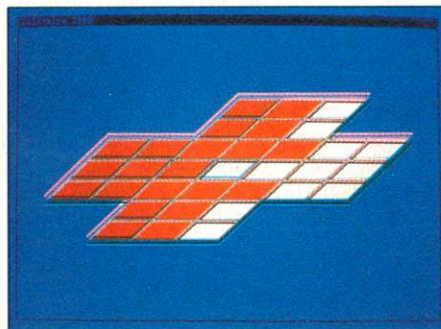
Zermartern Sie sich nicht den Kopf, bis Sie die falsche Zahl im Datafeld gefunden haben. – Alle Dateien auf der Databox zur Amiga DOS sind vom Autor und der Redaktion auf Fehlerfreiheit geprüft und im dazugehörigen System "probegelaufen".

```

(Compiling at look.c) by Matthias Kuhn for AmigaDOS (c)1990
-----
Program.      Look.c
---
Contents.     Anzeigen eines Textfiles oder nur ASCII Zeichen
Author.       Matthias Kuhn (mat)
Address.      (mat) Lützelschmiedstr. 21/1, D-6940 Weinheim
Phone.
Copyright.    AmigaDOS
Language.     ANSI - C
Translator.   Lattice C 5.04
Inputs.       top.lib
AmigaLibs.    Intuition Graphics
History.      V1.0 (mat) 3.Mär.1990
Remarks.

=====
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <exec/types.h>
#include <exec/memory.h>
#include <intuition/intuition.h>
#include <intuition/intuitionbase.h>
#include <intuition/screens.h>
#include <proto/exec.h>
#include <proto/stdio.h>
#include <proto/intuition.h>
    
```

Alle Daten sichtbar machen mit »Look«

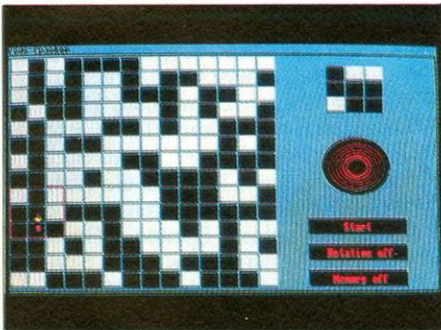


»Solitaire« – ein pffiffiges
Denkspiel für triste
Herbsttage

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

2 Disketten
voller
Programme

BOX



»Code Cracker« –
finden Sie möglichst
schnell das richtige Mu-
ster

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 zuzü-
glichen 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 11/90

X	X	Ein Such- und Denkspiel der Extraklasse präsentiert sich mit »Code Cracker«
X	X	Ob Textdateien oder ausführbare Dateien: »Look« zeigt sie alle an.
X	X	Wie man einen DOS-Handler programmiert, wird anschaulich an dem Programm »Joke-Handler« aufgezeigt.
X	X	Hier finden Sie Beispielprogramme zur Erläuterung der unterschiedlichen Datentypen des Modula-Kurses.
X	X	Intros und Vektorgrafik sind eine feine Sache. Das Programm »Megastars« zeigt Ihnen, wie solch eine Programmierung aussehen könnte.
X	X	Knackig kurze Assembler-Tips werden Ihnen von Willi Bäcker, dem Mitautor solch rasanter Spiele wie »Katakis« und »Turrican«, präsentiert
X	X	Und zusätzlich auf der Diskette: »Solitaire« ist ein recht kniffliges Denkspiel in AmigaBASIC.
X	X	Der Inhalt aller Fish-Disketten von 1 bis 360.

In Cracker-Kreisen sind diese Intros schon längst zu einer Art Briefkasten geworden, in denen die aktuellsten News ausgetauscht werden. Wenn Sie aber nicht länger nur zuschauen wollen, was da so vor sich "hinscrollt", "rastert", "smootht" und "bouncet", haben Sie mit dem Tool »The Megamighty Introdesigner« endlich Gelegenheit, professionelle Intros zu gestalten. Dabei brauchen Sie noch nicht einmal Programmierkenntnisse zu besitzen. Alles, was Sie benötigen, sind ein bißchen Kreativität und die nötigen Anwendungsprogramme: als da wären »Deluxe Paint« oder »Graphicraft« zum Erstellen von Logos oder Backgrounds und die Soundtracker-Versionen 2.0 oder 2.4 (die über jeden Insider zu beschaffen sind). Auch der Original-Soundtracker von Karsten Obarski der Firma EAS kann zur Erstellung von Sounds benutzt werden. Zusätzlich können noch andere Musikstücke in das Intro eingebunden werden, die mit dem »Futurecomposer« (Quelle: AÖY der Firma Digital Marketing) komponiert wurden.

Scroll-Effekte inbegriffen

Nachdem eben die wichtigsten Tools genannt wurden, die zum Introdesigner zusätzlich notwendig sind (sofern Sie Ihrem Intro einen persönlichen Touch verleihen wollen – denn was wäre ein von Ihnen erstelltes Intro mit einem Logo von RSI und einem Tune von

Timo Siebert

The Megamighty Introdesigner

Wer kennt sie nicht, die Vorspanne von Spielen, Anwendungen oder sonstiger Software, in denen der Programmierer "seinen Senf" dazugibt?

Megaforce?), geht es jetzt zum Programm selbst. Ein Intro kann aus folgenden Optionen zusammengesetzt werden: Scrolltext, Sternen-Scrolling (Hintergrundscrollen oder 3-D-Effekt), Backgroundpicture, Screentext, Logo (das auf vielerlei Art und Weise animiert werden kann), Rastermanipulationen, Sprites und Sounds. All diese Möglichkeiten können durch spezielle Untermenüs, die über Funktionstasten abrufbar sind, manipuliert werden. Zur näheren Beschreibung will unsere freiwillige Testperson "Harry der User" einmal einen Vorspann mit dem Intro-Designer kreieren.

Als erstes sollte man sich überlegen, ob man Logos (das sind meist Namen irgendwelcher Crews, grafisch ein bißchen aufgepeppt) in das Intro einbringen will oder nicht, ebenso die Hintergrundbilder. Hat man sich für diese Möglichkeit entschieden, müssen diese natürlich erst einmal gezeichnet werden. Das Programm ist in

der Lage, DPaint- und Graphicraft-Bilder einzubinden. Allerdings dürfen Logos nicht größer als 160x45 Pixel sein. Auch für die musikalische Untermalung ist gesorgt. Wie oben schon erwähnt, erkennt das Programm Soundtracker- und Futuresound-Module, die problemlos geladen und integriert werden können. Schaut man sich das Hauptmenü an, findet man »Sound« als erstes anwählbares Feature. Hier kann festgelegt werden, ob ein Sound gespielt werden soll oder nicht. Wenn ja, kann das Format definiert und die entsprechenden Module geladen werden. Außerdem ist eine Option »Equalizer« enthalten, die die Amplituden der einzelnen Stimmen hinterher im Intro grafisch anzeigt. Diese Option ist aber weiter von keiner großen Bedeutung. Harry entscheidet sich für einen Soundtracker-Tune und für die Equalizer-Funktion. Mit der Taste [Esc] kann man das Untermenü (dies gilt übrigens für alle Untermenüs) wieder verlassen.

Logos ohne Probleme

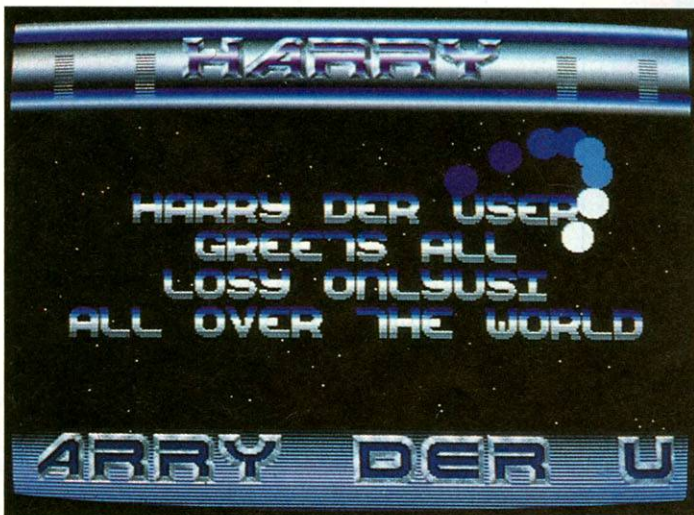
Im Menü »Background« kann man festlegen, ob man ein Bild oder lieber ein Sternen-Scrolling als Hintergrund benutzen möchte. Allerdings kann man nicht von beiden Möglichkeiten gleichzeitig Gebrauch machen. Es gibt zwei Arten von Sternen-Scrolling. Einmal ein "normales" Scrolling von links nach rechts und ein 3-D-Scrolling, was den Effekt hervorruft, als würde man in die Sterne hineinfliegen. Harry wählt nun das normale Sternen-Scrolling.

Beim nächsten Unterpunkt hat man die Möglichkeit, Logos einzubinden. Dies kann dann mit verschiedenen Effekten (Bouncen, "Wabbel-Effekt"...) versehen werden. Harry entscheidet sich hier für ein Logo mit Wabbel-Effekt. Mit der Taste [F4] gelangt man ins Rastermenü. Hier werden die Farben, die hinterher mit der Copperliste gesetzt werden, festgelegt. Zum einen für den Logo-Hintergrund und zum anderen für den Scroll-Text. Die Auswahl erfolgt über die Taste [Space].

Alle Farbdaten können als hexadezimale RGB-Werte eingegeben werden. Das Ergebnis wird im unteren Bildschirmrand gleich angezeigt. Zur besseren Justierung des Coppers lassen sich die Farbwerte mittels Cursortasten nach links oder rechts verschieben (demzufolge verschiebt sich der Copper nach oben beziehungsweise nach unten). Es sind auch schon einige Preset-copper-Listen vorhanden, die dann beliebig verändert werden können. Harry erstellt einen Logobackground mit blauen und grauen "Rasterrohren" und den Scrolltext mit einem graublauen Copper.

Im Screeneditor kann ein Text festgelegt werden, der in der Mitte des Bildschirms erscheinen soll. Hier besteht die Möglichkeit, zwischen fünf Zeichensätzen zu wählen. Harry editiert ein paar Zeilen und bringt im nächsten Menü Sprites ins Spiel. Man hat die Wahl zwischen selbsterstellten und festgelegten Sprites (Balls, Crosses, Triangles). Die Sprites bewegen sich in einer festgelegten Sinusform, die aber durch Strecken und Stauchen geringfügig verändert werden kann. Die Geschwindigkeit läßt sich selbstverständlich genau justieren. Harry benutzt die Ballsprites und eine etwas gestauchte Sinusform.

Um nun dem Intro noch die richtige "Message" zu verpassen, bedient man sich der Taste [F7]. Mit [F1] kann man den Scroll-Text editieren, [F2] löscht den gesamten Scroll-Text. Mit den Cursortasten kann man an den Anfang beziehungsweise an das Ende gelangen. Ferner sind im Scroll-Text Sonderfunktionen aufrufbar. Mittels der Tastenkombinationen [Shift-A], [Shift-B] und [Shift-C] kann die Ge-



Schnell erstellt – das Intro "Harry the User"

schwindigkeit variiert werden, [Shift-D und -E] stoppen den Scroll-Text kurzfristig. Außer den genannten Kombinationen sind fast alle Zeichen verwendbar. Mit den Funktionstasten [F3] bis [F7] können verschiedene Zeichensätze nachgeladen werden. Winzig kleine Zeichensätze, die kein Mensch lesen kann, sind längst "out".

Value for money?

Die verfügbaren Zeichensätze des Scroll-Textes weisen eine Mindestgröße von 36x36 Pixeln auf und werden absolut ruckelfrei bewegt. Harry, der User tippt eine Message in seinen Amiga und wählt Charset 1. Im Hauptmenü kann man mittels der Taste [F8] das fertige Intro betrachten. Die Funktion »Test Intro« kann natürlich auch schon vorher aufgerufen werden, um kleine Unsauberheiten auf der Stelle zu bereinigen. Mit [F9] kann das Intro als lauffähiger Object-Code (jaaa, nix Source, Leute) abgespeichert werden.

Zusätzlich kann man das Intro noch crunchen lassen. Nach rund 10 Minuten (das Zeichnen der Logos und das Komponieren der Musik natürlich nicht dazugerechnet) hat Harry sein Intro fertig und "spread" es erst mal kräftig. Diese relativ kurze Zeit spricht einerseits natürlich für eine hohe Benutzerfreundlichkeit des Programms, andererseits entsteht auch der Eindruck, daß der Intro-Designer damit wahrscheinlich nicht soviel Variationsmöglichkeiten aufweist. Und das nicht zu Unrecht. Intros können mit dem Intromaker noch so verschieden gestaltet werden, irgendwie wird man den Eindruck nicht los, daß hier ein Intro-Editor vorlag. Denn ein mit ihm generiertes Intro besteht meist aus den gleichen Objekten (also Scroller, Sprites...). Man möchte einfach soviel Effekte wie möglich mit hineinbringen, denn davon lebt ein Intro heutzutage. Deshalb werden meist alle Möglichkeiten des Intro-Designers genutzt.

Und trotzdem unterscheidet sich ein Intro des Programms nicht nur durch die Anzahl der Effekte von anderen noch "mit der Hand" programmierten Intros.

Da werden Scroller gedreht und gewendet, Logos "gestretcht", Copperfarben rotiert, alles Sachen, die dem »Megamighty Introdiesigner« fehlen. Denn der Intromaker ist in seinen Möglichkeiten etwas eingeschränkt, man muß seine Phantasie dem Introdiesigner anpassen, nicht umgekehrt (was sicherlich auch ein programmtechnisches Umding wäre). Doch für Freaks, denen ein Intro mit dem Nötigsten und ohne viel Schnickschnack genügt, ist der Introdiesigner eine empfehlenswerte Sache. Auch für Firmen oder Programmierer, die sich lieber auf die zu programmierende Software an sich konzentrieren, ist »The Megamighty Introdiesigner« sicherlich eine willkommene Erleichterung. Schauen Sie sich ihn doch einfach mal an. Der niedrig gehaltene

Preis dürfte für ein Programm dieser Art durchaus ein Anreiz sein.

(vb)

AMIGA DOS

Blitz ☆ licht

Name: The Megamighty Introdiesigner
Quelle: Digital Marketing Krefelder Str. 6
5142 Hückelhoven 2
Preis: 34,95 DM

Positiv:

- schnelle Einarbeitung möglich
- Scroll-Text kann manipuliert werden
- verschiedene Musikformate einlesbar
- günstiger Preis

Negativ:

- zu wenig Zeichensätze
- wenig Variationsmöglichkeiten

Computer & Zubehör



ONLY ONE

Der Umbausatz, der Ihren AMIGA in die richtige Form bringt.

Der Umbausatz bietet Ihrem AMIGA 500 nicht nur ein neues Gehäuse mit abgesetzter Tastatur, wie bei einem A1000 oder A2000, sondern auch erheblich mehr Platz. So bietet er außer für eine Speichererweiterung unter anderem noch für ein zweites Laufwerk oder für eine 3,5" Festplatte genügend Platz. - Und das alles ohne Lötarbeiten. Auch für die Zukunft ist gesorgt, eine Erweiterungsbox mit vielen Features wird es ab ca. Dezember geben.

Ihr A500 wird ein PC

Mit dem **MW 500 System** und dem **PC Power-Board**
Kombiangebot

DM 1098,--

Big Agnus
Speicherchips
Kickstart ROM 1.3
CIA 8520
u.v.m

Tagespreise 3,5" ext. Laufwerk durchgef. Bus
Tagespreise 3,5" int. Laufwerk für A2000 u. A500
DM 59,-- 5,25" ext. Laufwerk durchgef. Bus
DM 69,-- 512 kB Speichererweiterung



MW 500 System

komplett mit Kabelsatz und Tastaturgehäuse

DM 349,--

Warum eine Festplatte mit Gehäuse und nicht

ein Gehäuse mit Festplatte

40 MB autobootend mit A.L.F. 2

und **MW 500 System**

Kombiangebot

DM 1399,--

Miky Wenngatz

08105/24540

Kennwort AD 11/90

Jägerweg 31
8031 Gilching

Fordern Sie gleich unseren kostenlosen

Katalog an!

Lieferbedingungen:

Wir liefern ab Lager Gilching, solange der Vorrat reicht, per UPS Nachnahme zuzüglich Versandkosten. Bei Vorkasse frei Haus.
DDR per Post-Nachnahme. Ausland per Vorkasse zuzüglich DM 40,-- Versandkosten.

Besuchen Sie uns

**in Halle 10,
Stand 908**

**AMIGA
Köln 90**

Robert Marz

Goldene Werkzeuge

Die Golden-Tool-Serie hat sich zur Aufgabe gemacht, Programmierern das Leben zu erleichtern. Der erste Teil ist nun erhältlich.

Um größte Viensicherheit zu gewährleisten, sind Programme und Routinen zunächst nicht lauffähig, sondern müssen von einem Programm namens »Ginstall« umgewandelt und auf dem entsprechenden Zielmedium (Diskette, Festplatte, RAM Disk,...) installiert werden. Dies geschieht komfortabel über eine »Fileselectbox«, auf die auch die meisten ande-

das Amiga-Format konvertieren und umgekehrt. Da jedoch nicht das entsprechende Diskettenformat unterstützt wird, sind diese Programme wohl eher für Modem-Besitzer interessant oder für diejenigen, die mit einem Nullmodem und zwei verschiedenen Computern arbeiten. Programmierer müssen öfters ihre Listings

schlange und unterstützen ein anderes Format. So benutzt »Nprint« den sechsundneunziger Zeilendruck auf Epson-Druckern und »Dprint« den NLQ-Druck auf Epson und Star-Druckern.

Mit dem Program »Getkick« kann das Kickstart aus dem RAM ausgelesen werden. Es kann danach auf Diskette abgespeichert und später auf EPROMs gebrannt werden. »Getkick« patcht das Kickstart,

auswählt, oder man startet es mit Kommandozeile direkt von CLI aus. Damit man »SIFF« und »PICON« auch ausprobieren kann, wird eine zweite Diskette, die randvoll mit PD-IFF-Bildern ist, mitgeliefert. Mittels »SPEED« kann man die Geschwindigkeit eines beliebigen Speichermediums im Schreiben, Lesen und Löschen testen. Dabei werden insgesamt 100 KByte in Blöcken von je 1 KByte geschrieben.

Um Programmierer zu unterstützen und für diejenigen, die mit den Programmen »AST«, »APC«, »PCA« und »STA« nicht zufrieden sind, befinden sich noch die Objectcodes von Routinen auf der Disk, die in eigenen Programmen, nach Absprache mit D.I.T., weiterverwendet werden dürfen.

Die »Golden Tools 1« ist eine Diskette, die es in sich hat, sofern man selbst ein wenig mit dem Amiga programmiert. Die endgültige Version wird in einem stabilen Schuber und einem Ringbuch ausgeliefert, und natürlich ausführlich dokumentiert. D.I.T. bietet einen ausführlichen Kundenservice an und beantwortet auch Fragen direkt am Telefon.



(jb)

ren Programme zugreifen. Das erste Programm nennt sich »PICON«. Der Name leitet sich von Picture-Converter ab. Er konvertiert zum Beispiel IFF-Bilder in von einem Assembler direkt assemblierbare Anweisungen im Format »DC.l«, so daß man die Bilder im fertigen Programm nicht nachladen und umwandeln muß. Die umgewandelten Files können natürlich auch von anderen Sprachen, wie etwa C, genutzt werden. Ebenfalls eine File-Umwandlung vornehmen, aber in einer anderen Art, kann man mit einer Reihe von kleineren Programmen. Diese können damit nämlich Texte vom PC beziehungsweise Atari ST in

ausdrucken. Da die meisten aber nicht über einen hochtechnisierten 24-Nadel-Drucker verfügen, sind auf der Diskette drei Programme enthalten, die sich dieses Problems annehmen. Das Programm »Print« funktioniert ähnlich seinem Namensvetter auf den MS-DOS-Rechnern mit einem Druckerspooher, also einer Warteschlange, und es richtet sich einen eigenen Prozeß ein, so daß der Programmierer ungestört weiterarbeiten kann. Außerdem druckt es geschweifte Klammern und Umlaute gleichzeitig, ein Problem, mit dem meistens C-Programmierer zu tun haben. »Nprint« und »Dprint« arbeiten ähnlich, besitzen jedoch keine Warte-

indem es Fehler aus dem Trackdisk-Device korrigiert und Boot-Befehle ändert, die für den 68020- und 68030-Prozessor bestimmt sind. Auf diese Weise kann der Bootvorgang auf über die Hälfte verkürzt werden.

Das Programm »Qformat«, das bislang nur vom CLI aus gestartet werden kann, stellt eine schnelle und vor allem komfortable Formatieroutine zur Verfügung, mit der eine Diskette nicht nur installiert werden kann, sondern auch gleich eine auf das Nötigste abgespeckte Workbench installiert. Natürlich öffnet »Qformat« auch einen eigenen Prozeß. Mit dem Programm »SIFF« kann man sich IFF-Bilder ansehen, die man vorher ebenfalls aus dem Filerequester

AMIGA DOS

Blitz ☆ licht

Name: Golden Tool Disk 1
Quelle: D.I.T., Musfeldstr. 1a,
4200 Oberhausen 12,
Tel.: 0208/605645
Preis: 89,- DM

Positiv:

- viele Tools
- einfache Verwendung durch eigene Fileselect-box
- guter Service

Negativ:

- nichts Negatives anzumerken



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



Supra Corporation

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



Garry Glendown

Ein neuer Standard?

»Sculpt 4D« und »Turbo Silver« haben einen neuen Konkurrenten bekommen – »3D Professional«.

Dieses Programmpaket schickt sich an, seine Vorgänger auf die Plätze zu verweisen und der neue Standard auf dem Gebiet des Raytracing und der Raytracing-Animation zu werden. Stellt sich nur die Frage: Wird ihm das gelingen, oder ist er nur "ein weiterer Raytracer"? Geliefert wird »3D Professional« in einem wirklich riesigen Paket, das man leicht um die Hälfte hätte verkleinern können. Es enthält neben den sechs Disketten und drei Handbüchern noch eine Videokassette. Eine positive Überraschung oder nur ein "unnützes" Gimmick? Das wollten wir natürlich gleich mal probieren. Noch bevor das erste Handbuch aufgeschlagen oder gar eine Diskette eingelegt wurde, landete die Kassette im Videorecorder.

In einer zweistündigen (englischen) Vorführung wird zuerst eine allgemeine Einführung

zur Bedienung des Programms gebracht, danach werden an vier ausführlichen Beispielen ein Großteil der Funktionen des Programms demonstriert. Dank der guten Gliederung und der genauen Erklärung der einzelnen Schritte gelingt es dabei, den Benutzer sehr gut in die Bedienung einzuweisen. Auch sollte es schon mit einem durchschnittlichen Schulenglisch möglich sein, die gesprochenen Texte zu verstehen, da der Sprecher sehr deutlich spricht und nur einfache Variablen verwendet werden. Und wenn's doch mal nicht klappt – dann schlägt man das Handbuch auf und findet dort den exakten Text noch einmal gedruckt wieder.

Es empfiehlt sich, jeweils einen Teil des Bandes anzusehen und diesen dann, in Verbindung mit dem Benutzerhandbuch, am Rechner zu wiederholen. Dadurch redu-

ziert sich die Menge der zu lernenden Informationen, und man kann auch Teile, die vielleicht nicht so ganz klappen, gleich nochmal am Video "nachschießen".

Man sollte sich aber nicht nur auf das Video verlassen. Ein Nachschlagen im Handbuch ist immer ratsam. Doch eines kann man sicher sagen: Dank des Videos ist dieses (im Gesamtumfang recht komplexe) Programmpaket vor allem in der Anfangsphase, bei der man noch auf schnelle "Erfolgslebnisse" angewiesen ist, besonders gut zu bedienen. Kaum ein anderes Programm, das eine entsprechende Komplexität besitzt, habe ich bisher ähnlich schnell bedienen können. Diese Idee sollten andere Softwarehäuser auch übernehmen...

Ein Wort zu den Handbüchern: Neben dem Benutzerhandbuch (das die Video-Übungen enthält) wird für »3D Professio-

nal« noch ein Reference Manual mitgeliefert, in dem alle Funktionen im einzelnen beschrieben werden. Dies sollte man auch irgendwann einmal komplett durchlesen. Hier werden zusätzlich alle Funktionen, die in der kurzen Einweisung per Video verständlicher werden können, erklärt. Und nicht zuletzt enthält es ebenfalls einen ausführlichen Index, anhand dessen man bei Problemen recht schnell die passende Stelle findet.

Doch kommen wir (endlich) zum eigentlichen Programm. Wer schon mit »Sculpt« gearbeitet hat, dem wird auf den ersten Blick ein deutlicher Unterschied auffallen. Während der Editor von »Sculpt« die Szene in drei Fenstern aus allen Richtungen gleichzeitig darstellt, wurde bei »3D Professional« darauf verzichtet. Statt dessen hat man ein großes Editor-Fenster und somit einen besseren Überblick über die Szene. Am Rande des Fensters findet man zwei Scroll-Gadgets zur Auswahl der Position sowie ein Gadget zur Vergrößerung. Daneben gibt es noch eine Reihe weiterer Funktionen, die per Gadget anwählbar sind. Dazu gehört zum Beispiel das Drehen um die Achsen, Verändern der Größe, Spiegeln und Biegen.

Die Objekte im Editor werden normalerweise mit gefüllten Flächen dargestellt, die schon in etwa das spätere Aussehen widerspiegeln. Bei komplexeren Szenen kann man dies jedoch abschalten und nur mit einer Wire-Frame-Darstellung arbeiten.

Um ein Objekt zu selektieren, kann man es entweder direkt anwählen oder auch einen Rahmen um alle gewünschten Objekte legen. Alle oben aufgeführten Funktionen werden dabei jeweils nur auf die selektierten Objekte angewandt.

Wer glaubt, daß man bei der Verwendung eines solchen Raytracing-Programms die Tastatur getrost in der Ecke liegen lassen kann, ist bei »3D Professional« nicht ganz auf der richtigen Spur. Um verschiedene Optionen einer Funktion aufzurufen, muß man gelegentlich auch mal eine Taste drücken. So wird zum Beispiel ein Objekt in allen drei Richtungen vergrößert oder gestaucht, indem man zusätz-

lich zu einem Gadget noch die Taste [Z] drückt (Das Programm berücksichtigt keine eigenen Keymaps. Bei einer deutschen Tastatur muß dementsprechend das [Y] gedrückt werden.). Dies mag zwar auf den ersten Blick die Sache nur kompliziert erscheinen lassen, doch spart man sich damit zusätzliche Menü-Funktionen, und schon nach kurzer Zeit sind die recht gut gewählten Kombinationen in Fleisch und Blut übergegangen.

Wenn man die Menüs von »3D Professional« durchgeht, fällt auch auf, wieviele vorgefertigte Elemente es schon gibt. Man könnte zwar auf diese Teile verzichten, doch spart man beim Aufbau von Szene durch Verwendung dieser Elemente doch einige Zeit (Im Handbuch wird übrigens für jedes Objekt erklärt, wie man es auf "herkömmliche" Weise unter Verwendung der 3D-Professional-Funktionen erhält).

Während diese Standard-Objekte schon recht praktisch sind, wird's erst beim nächsten Menü richtig interessant – hier kann man seine eigenen Objekte definieren. Den Anfang macht dabei die Funktion »Lathe-Rotationskörper«. Dabei wird die Schnittfläche im Editor eingegeben und später um die Achse rotiert. Dabei können neben den normalen Editier-Funktionen nicht nur gerade Kanten eingegeben werden, sondern auch Kreise und Bezier-Kurven.

Auf die Dauer hilft nur Power...

Der nächste Menüpunkt ruft den Programmteil »Profile« auf. Hierbei wird ein Objekt aus Grundflächen aufgebaut, die untereinander verbunden werden. Dabei kann man im rechten Teil des Bildschirms die Position der einzelnen Ebenen bestimmen und deren Größe erkennen. Mit der nächsten Funktion kann man Objekte erzeugen, die spitz zulaufen. Dabei gibt man auch wieder die Grundfläche vor, »3D Professional« läßt die Seitenflächen dann in einem Punkt zusammenlaufen.

Auch fraktale Landschaften stellen das Programm vor keine Probleme. Auch hier gibt es einen entsprechenden Menüpunkt. Dabei gibt man ver-

schiedene Parameter, zum Beispiel die Anzahl der Flächen und die Höhendifferenzen an. Heraus kommt dann ein Landschaft, die manche darauf spezialisierte Programme kaum so gut hinbekommen.

Während es sich bei dem letzten Punkt um eine oft gewünschte Funktion handelt, ist der nächste Programmpunkt bisher noch nie in einem ähnlichen Programm aufgetaucht: die Erzeugung eines Baumes. Hierbei werden über ein Dutzend Parameter eingestellt, die das Aussehen des Baumes beeinflussen. Je nach Parametereinstellung kommt dann ein mehr oder weniger realistischer Baum heraus. Als weitere Funktionen bietet »3D Professional« noch das Einbinden von Amiga-Fonts sowie das Erstellen von Bodenflächen (die man beispielsweise auch als Wände benutzen kann).

Womit bauen wir heute?

Hat man sich ein Objekt zusammengebaut, muß man normalerweise noch festlegen, wie es denn nun weiter gestaltet ist, das heißt, welche Farbe es hat, ob es glänzt oder matt ist, welche Oberflächenstruktur es hat, und vieles mehr.

Während man bei anderen Raytracern wie »Sculpt« nur recht eingeschränkte Möglichkeiten hat, bietet »3D Professional« hier wieder einmal Funktionen im Überfluß. Für die normale Darstellung beim Editieren wird dabei eine einfache Repräsentation durch eine von 16 Farben vorgenommen, für das endgültige Bild werden für die Farbwerte Farbtöne mit einer Auflösung von 24 Bit vorgenommen (8 Bit pro Farbanteil). Während dies mit einem normalen Amiga nicht darzustellen ist, hat man jedoch die Möglichkeit, die erzeugten Bilder auf andere Rechner zu bringen und dort mit der erhöhten Farbauf Auflösung zu verwenden. Jedoch auch für den Amiga-Bereich ist diese Option interessant, da mit Grafikkarten zu rechnen ist, die auch solch eine Farbvielfalt verkraften (zum Beispiel die Tigrakarte).

Um die Beschaffenheit der Oberfläche zu bestimmen, besitzt »3D Professional« sechs

Werte. Diese definieren unter anderem die Parameter Transparenz, Lichtbrechung (zur Zeit nicht unterstützt), Reflexion und Glanz. Damit hat man die Möglichkeit, die verschiedensten Oberflächen zu simulieren. Um es dem Benutzer ein wenig einfacher zu machen, wurde außerdem ein Satz von neun Voreinstellungen eingebaut, mit dem die gängigsten Oberflächen ausgewählt werden können.

Doch die interessanteste Option dieses Programms sind die verschiedenen Oberflächenstrukturen, die mir in anderen Programmen oft gefehlt haben. Hier hat man die Auswahl zwischen einem Fleckenmuster, Marmor, Holz, Backsteinen, normalen Farbverlauf, einem Wellenmuster sowie einer Hügeloberfläche. Außerdem kann man dem jeweiligen Objekt auch noch eine Lage Schnee auflegen. Durch Verändern der verschiedenen Parameter dieser Effekte kann man sie an die jeweiligen Gegebenheiten anpassen und so hervorragende Effekte erzeugen.

Mit Pinsel und Leinwand...

Nachdem man die Szene soweit fertig hat, kann man daran gehen, das Bild von »3D Professional« berechnen zu lassen. Dabei ist das Programm recht flexibel und eröffnet auch hier eine ganze Reihe Möglichkeiten. Neben den grundsätzlichen Modi »Overscan« und »Interlace« hat man die

Möglichkeit, zwischen den Auflösungen »LORES«, »HIRES«, »HAM« und »A2024« zu wählen. Beim Berechnen der Grafik kann man ebenfalls einige Parameter modifizieren, wobei man neben einfachen Algorithmen auch »Gourad« und »Phong-Schattierung« auswählen kann, die zwar einen höheren Rechenaufwand bedeuten, dafür aber ein erheblich realistischeres Bild liefern. Doch leider haben die eingebauten Algorithmen noch einen Nachteil: Sie liefern keinen Schatten, er muß (laut Handbuch) mittels eines zusätzlichen Objektes eingefügt werden. Dies wird wohl in einer der folgenden Versionen behoben werden, doch wer sich nicht so lang mit Bildern ohne Schatten (oder mit zusätzlichem Arbeitsaufwand) begnügen will, kann einen eigenen Algorithmus einbauen. Dabei wird dem Benutzer im Handbuch eine vollständige Liste der internen Datenstrukturen geliefert, die Übergabe erfolgt mittels einer Exec-Message. Somit also endlich mal ein Programm, an dem man selber herumbasteln kann.

Ist das Bild erst einmal fertig berechnet, geht's ans Ausgeben. Und auch hier ist »3D Professional« mal wieder seinem Namen gerecht geworden. Während andere Programme gerade mal in der Lage sind, ihre Grafiken "ordentlich" auf dem Bildschirm auszugeben, sind die Programmierer hier einen Schritt weitergegangen. Neben der Bildschirmausgabe ist es möglich, das Bild in einem RGB-Format (entweder

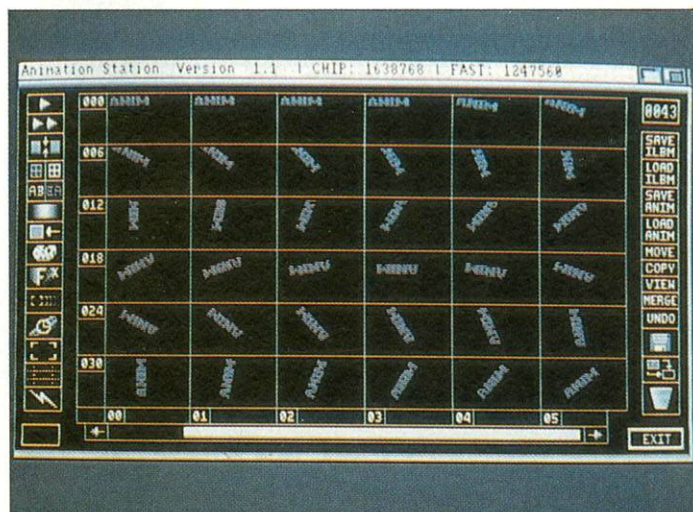


Bild 1. Animation Station – Animationsmanipulation leicht gemacht

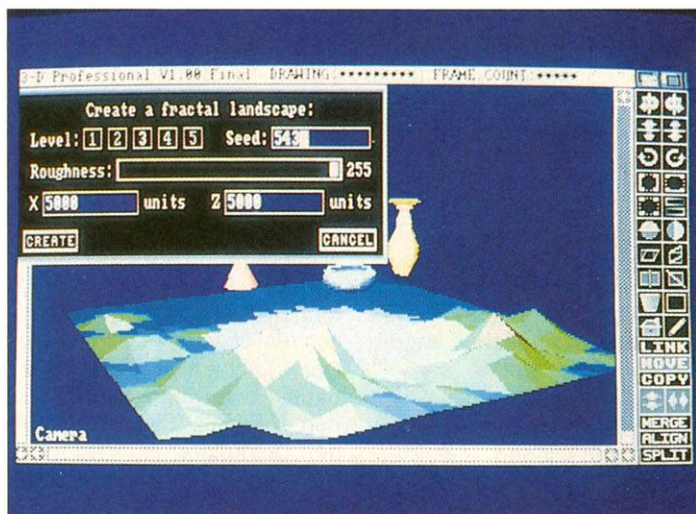


Bild 2. Fraktale Landschaften leichtgemacht

für »3D Professional«, »DBW Render« oder als »RAW« auszugeben, als ILBM-Bild abzuspeichern, Postscript-Dateien zu erzeugen (entweder als eigenständiges PS-Format, oder als Encapsulated, das von manchen Programm weiterverarbeitet wird). Dazu kommt noch eine Option, mit der man Stereo-Bilder berechnen kann. Wichtig für Anwender, die mit »3D Professional« Animationen erstellen wollen, dürfte auch noch die Möglichkeit sein, mit dem Programm direkt einen Videorekorder ansteuern zu können.

Für den fortgeschrittenen Anwender von »3D Professional« gibt es noch eine Menge mehr Möglichkeiten, die Berechnungen des Programms zu manipulieren. Die wahren Qualitäten wird man wohl erst mit der Zeit zu schätzen wissen, doch ist auch das, was man schon Anfangs zu erfahren vermag,

so gut, daß man das Programm kaum noch missen möchte. Auch die Möglichkeit, Objekte von anderen 3D-Programmen einzuladen, sind schon sehr vorteilhaft. Und wenn man noch dazu bedenkt, daß manche Hersteller ihre Datenformate krampfhaft geheimzuhalten versuchen, wird hier ein anderer Weg gegangen. Dem Anwender soll Gelegenheit gegeben werden, rund um das Programm eigene Anwendungen zu bauen und es so noch weiter verbessern zu können. Bleibt eigentlich nur abzuwarten, wann die ersten externen Erweiterungen kommen.

Zugabe, Zugabe!

Und wem das alles noch nicht reicht, für den hält das 3D-Professional-Paket noch ein weiteres As im Ärmel: Animation Station. Mit diesem Paket be-



Bild 3. Gestaltungsmöglichkeiten unter »3D Professional«

kommt man ein Programm an die Hand, mit dem man nicht nur einfache (und kompliziertere) Animationen erstellen kann, sondern auch vorhandene Animationen bearbeiten kann.

Dabei kann das Timing jedes einzelnen Bildes geändert werden, es können neue Bildteile eingefügt werden und die Grafiken verändert werden. Man kann neue Teile in die Animation einfügen, andere Teile ausschneiden, Farbtabelle ändern. Man merkt, daß dieses Programm ebenfalls für die gehobenen Ansprüche programmiert wurde, und nicht einfach eine kleine »Haus- und Hobby-Arbeit« war. Wie auch bei »3D Professional« selber kommt hier auch der Appetit mit dem Essen, die wahren Qualitäten entdeckt man erst, wenn man sie braucht.

Bravo!!!

Ein abschließendes Urteil zu dem Paket »3D Professional« kann eigentlich nur »Superprogramm« lauten. Sowohl die Quantität als auch die Qualität dieses Programms vermag voll zu überzeugen. Allerdings muß gesagt sein, daß man es erst dann richtig ausnutzen kann, wenn man einen etwas besser ausgestatteten Amiga besitzt – zum einen ist es recht anspruchsvoll bei der Belegung des Speichers, zum anderen auch sehr rechenaufwendig. Wenn man die aufwendigeren Render-Algorithmen »Phong« oder »Gourand« verwendet, so ist ein normaler 68000er Amiga mal leicht einen Tag beschäftigt. Auch das

Editieren wird, aufgrund der Berechnungen nach jeder Operation, im normalen Darstellungsmodus schon sehr nervenaufreibend. Ab einer 68020-Karte wird das Programm dann allerdings schon »einsatzfähig«, zumal eine Kombination CPU/FPU 68020/68881 mitgeliefert wird (was sich einige andere Hersteller auch einmal zu Herzen nehmen sollten...).

(vb)

AMIGA DOS

Blitz ☆ licht

Name: 3D Professional
Quelle: Atlantis Software
Dunantstr. 53
5030 Hürth
Tel.: 02233/41081
Preis: 948,- DM

Positiv:

- gute Handbücher
- Videocassette gibt umfangreiche Informationen
- Programm ist leicht zu handhaben
- Oberflächenstruktur
- Vielfalt an Objekteditoren
- viele verschiedene Darstellungsmodi
- unterstützt Objektformate anderer 3D-Raytracer

Negativ

- erst sinnvoll einzusetzen ab 2,5 MB RAM
- Voraussetzung sollte mindestens eine 68020-Karte sein
- keine Schattendarstellung
- Handbuch in Englisch

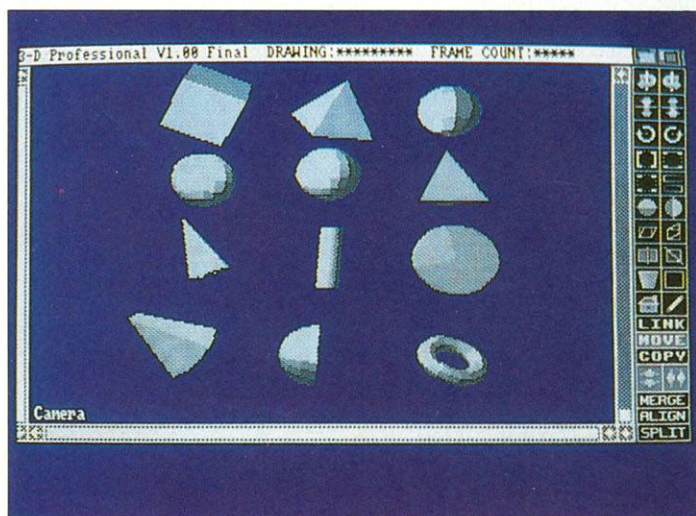


Bild 4. Die Basis-Objekte

Auf geht's zur

AMIGA KÖLN 90

*Unter der
Schirmherrschaft
von Commodore
und dem
Amiga Magazin*

09.11.-11.11.90
Köln Messe,
Halle 10 und 12
(08.11.90 Fachbesuchertag)

Öffnungszeiten:
08.11.90/Fachbesuchertag:
10:00 - 18:00 Uhr

09. 11.-11. 11.90
9:00 - 18:00 Uhr

Eintrittspreise:
Tageskarte 09. 11.-11. 11.90
Schüler/Studenten: DM 10.-
Erwachsene: DM 15.-

Fachbesucherkarte: DM 35.-
(Gültigkeit: alle vier Messetage)

**Pro Messe erwarten Sie
über 100 Aussteller und
30000 AMIGA-Fans!**

**Kommen Sie vorbei,
wir freuen uns auf Sie**

Information:

☆☆☆
☆☆☆ **AMI** ☆☆☆
☆☆☆ **Shows**

Ami Shows Europe GmbH
Zugspitzstraße 2A
8011 Vaterstetten
Fax: 08106-34094

Guido Coenen

"Charley's Tante" auf dem Amiga

Im gleichnamigen Kinofilm trieb das Schicksal den armen Peter Alexander in die Rolle (und Kleider) einer nichtexistenten Person. Auf dem Amiga steckt hinter der Fassade des neuen DTP-Programms »Publishing Partner« ebenfalls ein alter Bekannter.

Niemand anderes als das ausgereifte DTP-Programm »PageStream« in der Version 1.82 verbirgt sich nämlich hinter dem vertrauenerweckenden Namen. Und ebenso, wie bei Peter Alexanders Verkleidung außen nicht draufsteht, daß Peter Alexander drinsteckt, enthält auch die Verpackung keinen Hinweis auf »PageStream«; sogar auf Bildschirmfotos wurde verzichtet. Dabei jedoch von Absicht zu sprechen, heiße, dem deutschen Vertreter Unrecht anzutun; wirbt er doch in halbseitigen Anzeigen gerade damit, daß es sich bei »Publishing Partner« um die deutsche Version von »PageStream« handelt. Wichtiger in diesem Zusammenhang ist der ebenfalls auf der Verpackung fehlende Hinweis, daß das Programm mindestens 1 MByte RAM benötigt; dies sollte dringend kenntlich gemacht werden.

Die Anleitung: unbefriedigend

Das deutsche Handbuch enthält dagegen direkt im ersten Satz einen Hinweis, daß es sich bei »Publishing Partner« um eine PageStream-Version handelt. Weiterhin ist jedoch auch ersichtlich, daß Programm und Anleitung von einer französischen Firma übersetzt wurden. Diejenigen unter Ihnen, die nun sorgenvoll mit dem Kopf schütteln, ahnen es schon: Eine Glanzleistung sieht anders aus als diese Anleitung. Gelegentliche Rechtschreibfehler sind ja noch zu entschuldigen; kriminell wird's erst dann, wenn der Anwender mit einem Mischmasch ver-

schiedener Computersysteme konfrontiert wird und dies sinn gemäß auf den Amiga übertragen muß. Im Handbuch sieht das so aus, daß der Text sich auf den Amiga bezieht, während sämtliche Bildschirmfotos vom Atari ST stammen. Sobald es jedoch um DOS-Angelegenheiten geht, werden (im Handbuch zur Amiga-Fassung!) MS-DOS-Pfade angegeben, komplett mit Backslash und Laufwerksbuchstabe »A.«. So beispielsweise im Kapitel über die Fonts, S. 4.31: »Der Suchpfad einer Times kann z.B. A:\TIMES.FM [...] sein.«

Das Kapitel über Typographie und die Einführung in das DTP haben mir dagegen ausgezeichnet gefallen, hier werden Fachbegriffe verständlich erläutert und dem Anwender die Gestaltungsmöglichkeiten eines Dokumentes nahegelegt. Was diese Kapitel allerdings im Handbuch zu suchen haben, bleibt im Dunkeln.

Schließlich macht alleine der Preis von rund 800,- DM schon deutlich, daß der professionelle Bereich angesprochen werden soll, und dieser Personenkreis benötigt keine DTP-Einführung. Wichtiger wären in diesem Falle Kurzübersichten aller Programmbefehle für den Anwender, der zwar schon weiß, was er mit seinem Dokument machen will, aber wissen möchte, wie er das in »Publishing Partner« anstellt.

Diese Übersichten fehlen leider; der Anwender muß sich seine Informationen mühselig zusammensuchen, und so entstand dieser Bericht auch nach mehreren Stunden stetigen Blättern im Handbuch. Ob es wirklich nicht möglich ist, zu einem 800-DM-Programm eine vernünftige Anleitung mitzuliefern? Im Programm selbst werden weitere Übersetzungsfehler deutlich; hieß es beispielsweise in



»PageStream« noch sinngemäß, man könne mit der Taste »HELP« den Druckvorgang abbrechen, schreibt »Publishing Partner« »Abbruch mit UNDO« auf den Schirm. Gemeint ist die Help-Taste. Bei den eigentlichen DTP-Funktionen unterscheiden sich beide Programme jedoch überhaupt nicht; hier wurde wirklich nur reine Übersetzungsarbeit vorgenommen. Da sicherlich die meisten Leser schon unzählige PageStream-Tests gesehen haben, möchte ich hier nur einen groben Überblick über die Funktionsvielfalt dieses sehr ausgereiften Programms geben.

Das Programm: ausgezeichnet

Der Lieferumfang beinhaltet neben dem Hauptprogramm insgesamt 22 verschiedene Vektor-Fonts, die für allerbeste Druckqualität sorgen. Normalerweise benutzt »Publishing Partner« zum Ausdruck einen der 45 mitgelieferten Druckertreiber, die sogar so alte »Schätzchen« wie den Star Gemini 10-X unterstützen, aber auch für (farbige) Postscript-Ausgabe oder direkt zur Umsetzung in einer Linotype geeignet sind. Obwohl das Programm einen Editor für Texte und Grafikelemente beinhaltet, können Texte und Objekte auch von einer Vielzahl weiterer Programme übernommen werden. Da der eingebaute Texteditor nicht zu den schnellsten und komfortabelsten gehört, empfiehlt es sich, Texte mit einer regulären Textverarbeitung zu erstellen und in »Publishing Partner« zu impor-

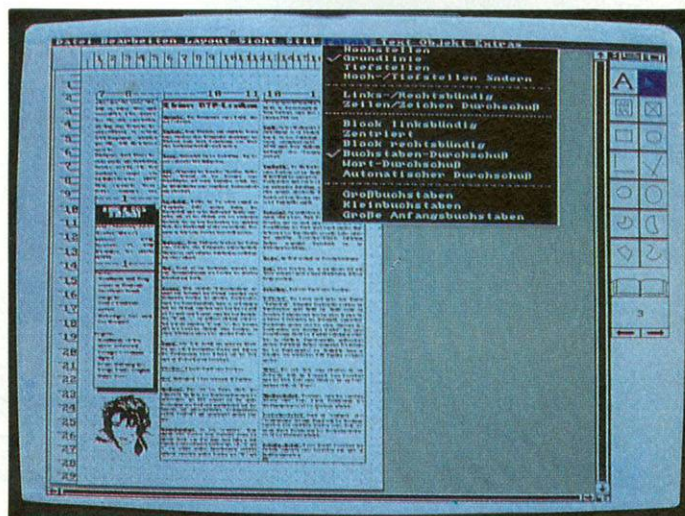


Bild 1. Das Format einer Probeseite läßt sich vielseitig variieren

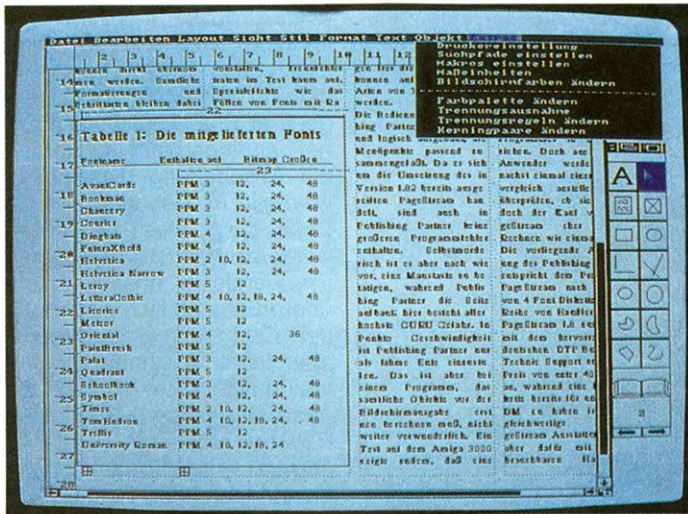


Bild 2. Viele Extras inbegriffen

tieren. Unterstützt wird selbstverständlich das IFF-Format, aber auch Encapsulated Postscript-Dateien sowie Texte von den Programmen »Excellence«, »WordPerfect«, »ProWrite« und einer Reihe weiterer können direkt übernommen werden.

Sämtliche Formatierungen und Schriftarten bleiben dabei voll erhalten. Dagegen ist es nicht möglich, die Texte von nur in Deutschland verbreiteten Programmen wie z.B. »Beckertext« zu übernehmen. Hier muß der Umweg über ASCII-Dateien gegangen werden. Die eingegebenen oder importierten Texte können in vielfältiger Art gestaltet werden; neben 14 verschiedenen Stileigenschaften sind Format

und Font für jeden Buchstaben frei wählbar. Die Fontgröße ist in Schritten ab 0,02 Punkten änderbar. Eine eingebaute deutsche Trennautomatik sorgt dafür, daß bei Blocksatz keine allzugroßen Zwischenräume entstehen. Der Trennvorgang geht relativ flott vonstatten, Trennfehler traten im Test kaum auf.

Spezialeffekte wie das Füllen von Fonts mit Rastern sind ebenfalls in Minutenschnelle realisierbar.

Sowohl die Texte als auch grafische Elemente werden von »Publishing Partner« als »Objekte« behandelt. Diese Objekte sind vielfältig manipulierbar; sie können frei auf der Seite platziert werden und beliebige Größe annehmen. Das Ver-

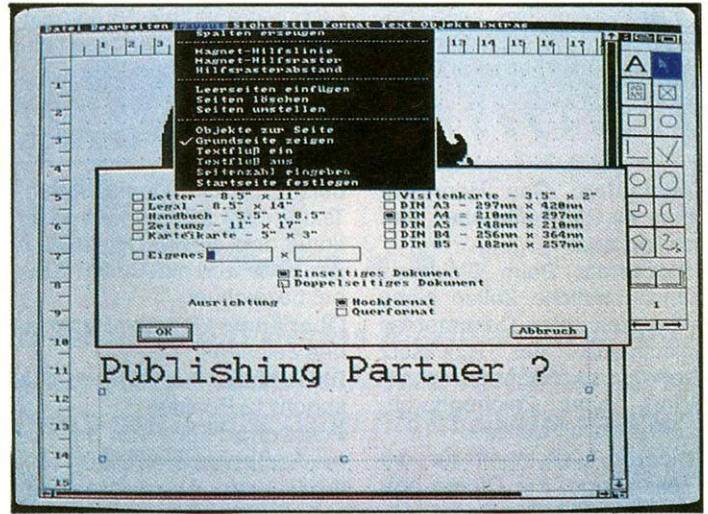


Bild 3. Layout-Erstellung ohne große Probleme

schieben und Kopieren von Objekten geht ebenso flott von der Hand wie die Farbgestaltung, das Einfügen von Füllmustern oder das Ausrichten der Objekte, absolut oder relativ (anhand anderer Objekte). Sämtliche Objekte, also auch Texte, sind in allen drei räumlichen Richtungen frei drehbar, Grafiken können auf verschiedene Arten von Text umflossen werden.

Die Bedienung von »Publishing Partner« ist einfach und logisch aufgebaut, die Menüpunkte passend zusammengefaßt. Da es sich um die Umsetzung des in Version 1.82 bereits ausgereiften »PageStream« handelt, sind auch in »Publishing Partner« keine größeren Programmfehler ent-

halten. Selbstmörderisch ist es aber nach wie vor, eine Maus-taste zu betätigen, während »Publishing Partner« die Seite aufbaut; hier besteht allerhöchste Guru-Gefahr.

In punkto Geschwindigkeit ist das Programm nur als lahme Ente einzustufen. Das ist aber bei einem Programm, das sämtliche Objekte vor der Bildschirm ausgabe erst neu berechnen muß, nicht weiter verwunderlich. Ein Test auf dem Amiga 3000 zeigte zudem, daß eine Turbokarte auch aus lahmen Enten Rennautos machen kann; die gewohnten Wartezeiten traten kaum noch in Erscheinung. Grundsätzlich kann man beim Kauf des »Publishing Partner« nichts verkehrt machen. Es handelt sich um

ids

**IDS Hard + Software GMBH 6921 Epfenbach
Frohnberg 23 Tel 07263/5693 Fax 1739**

AMIGADRIVES

anschlußfertig * amigafarbenes Metallgehäuse
automatische Diskchangererkennung * abschaltbar
Busdurchführung bis DF3 * 5.25" mit 40/80 Track

3.5" EXTERN 159.-
5.25" EXTERN 208.-
3.5" INTERN 129.-

SPEICHER

Amiga 500 512KB 109.-

abschaltbar * Mmegabit Chips
Akkuuhr * intern

Amiga 1000 2MB 599.-

externes Gehäuse * abschaltbar
durchgeführter Bus * mit Testdiskette

Amiga 2000 8MB 588.-

8MB Karte * mit 2MB bestückt
intern * Megabit chips

ein Programm, das auch professionellen Ansprüchen gerecht wird und auf allen Rech-

nern von A500 bis A3000 lauffähig ist, sofern mindestens 1 MByte RAM vorhanden ist.

Eine Festplatte beschleunigt die Arbeit mit dem Programm zusätzlich. Außerdem möchte

ich aber noch etwas zum Preis anmerken. Sicherlich sind 800,- DM kein Pappenstiel, und nur professionelle Anwender (für die »Publishing Partner« auch gedacht ist) werden die Anschaffung dieses Programms in Betracht ziehen. Doch auch diese Anwender werden zunächst einmal einen Preisvergleich anstellen und überprüfen, ob sich nicht doch der Kauf von »PageStream« eher lohnt. Rechnen wir einmal nach: Die vorliegende Ausstattung des »Publishing Partner« entspricht dem Programm »PageStream« nach Zukauf von vier Font-Disketten. Eine Reihe von Händlern bietet »PageStream 1.82« zusammen mit dem hervorragenden deutschen DTP-Buch von Technic Support zu einem Preis von unter 400,- DM an, während eine Font-Diskette bereits für unter 75,- DM zu haben ist. Eine gleichwertige PageStream-Ausstattung, aber dafür mit einem brauchbaren Handbuch, wäre also für unter 700,- DM zu haben. Hier muß letztendlich jeder Anwender für sich entscheiden, ob ihm eine deutsche Benutzerführung in seinem DTP-Programm wirklich gut 100,- DM Mehrkosten wert ist.

(vb)

Kleines DTP-Lexikon

Auslaufen: Das Herausragen einer Grafik über den Seitenrand.

Blocksatz: Beim Blocksatz sind sämtliche Zeilen eines Absatzes oder Dokumentes gleichlang; der Blocksatz kann durch Veränderung von Wort- oder Zeichendurchschuß erzielt werden.

Cicero: Maßeinheit für die Zeilenlänge. Ein Cicero entspricht 4,55 Millimetern.

DTP: Abkürzung des Begriffes »DeskTop Publishing«. Gemeint ist das Erzeugen eines Schriftstückes mit einem Computer, sobald dies die Möglichkeiten einer »regulären« Textverarbeitung übersteigt.

Durchschuß: Nichts, das Sie schon einmal in »Rauchende Colts« gesehen haben. Der Durchschuß wird in der Maßeinheit Punkte angegeben und ist der Abstand von der vorherigen Textzeile bis zu den Textzeichen der nächsten Zeile. Der Durchschuß ist also ein vertikaler Freiraum, in dem keine Zeichen dargestellt werden.

Flattersatz: Beim Flattersatz besitzen die Zeilen eines Absatzes oder Dokumentes unterschiedliche Längen; sie sind entweder linksbündig, rechtsbündig oder zentriert.

Font: Damit ist ein Zeichensatz gemeint, also eine Zusammenfassung von Zeichen der gleichen Schriftart und Größe.

Kerning: Wird ebenfalls »Unterschneidung« genannt und bestimmt den Abstand zwischen zwei bestimmten Zeichen des gleichen Zeichensatzes. Auch bei Proportionalfonts kann es vorkommen, daß der Abstand zwischen zwei Zeichen zu groß ist. So wird zum Beispiel zwischen den Buchstaben »W« und »q« ein größerer Abstand erforderlich sein als zwischen »V« und »q«. Das Kerning gibt den korrekten Abstand jedes Zeichens zu einem beliebigen anderen des gleichen Fonts an.

Layout: Jede Seite besteht aus mehreren Objekten wie

beispielsweise Überschriften, Texten und Bildern. Die Positionierung dieser Objekte auf der Seite wird als (Seiten-)Layout bezeichnet.

Oberlänge: Oberster Punkt eines Zeichens.

Pica: Maßeinheit; 1 Pica entspricht 12 Punkten.

PostScript: Eine von der Firma Adobe Inc. entwickelte Methode der Druckeransteuerung, die insbesondere für DTP gedacht ist. Die große Verbreitung von »PostScript« gewährleistet, daß Dokumente auf unterschiedlichen Konfigurationen erstellt, bearbeitet und ausgedruckt werden können.

Proportionalfont: In den »regulären« Fonts nimmt jedes Zeichen beim Ausdruck genau die gleiche Breite ein. Das kann dazu führen, daß zwischen zwei schmalen Zeichen wie zum Beispiel »i« und »l« ein relativ großer Zwischenraum entsteht, während zwischen großen Zeichen wie »W« und »Z« ein unverhältnismäßig kleiner Zwischenraum bleibt. In Proportionalfonts nehmen die Zeichen beim Ausdruck einen ihrer eigenen Breite entsprechenden Platz ein.

Punkt: In der Mathematik besitzt ein Punkt keine Ausdehnung; er ist lediglich über Koordinaten definiert. In der Satztechnik entspricht ein Punkt einem zweiundsiebzigstel Zoll (also ungefähr 0,3759 mm). Diese Maßeinheit wird auch als »Didot-Punkt« oder »Typographischer Punkt« bezeichnet.

Punktgröße: Die Maßeinheit für die Dimensionen eines Zeichens ist die Punktgröße; diese wird in (Didot-)Punkten angegeben. Die Punktgröße eines Zeichensatzes wird von der höchsten Oberlänge bis zur niedrigsten Unterlänge gemessen. Da ein Zeichen niemals gleichzeitig Ober- und Unterlänge besitzt, ist kein Zeichen eines Fonts so hoch, wie es die Punktgröße angibt.

Rasterfont: Im englischen auch Bitmap-Font genannt, stehen diese Fonts Punkt für Punkt so im Speicher, wie sie

auch ausgedruckt werden. Eine Vergrößerung der Fonts wird durch simples Multiplizieren der Einzelpunkte erreicht, dabei entstehen unschöne »Treppchen«-Effekte. »Publishing Partner« verwendet Rasterfonts für die Bildschirmarstellung.

Serifen: Im Font vorhandene Verschnörkelungen.

Text: Jedes Zeichen, das in irgendeiner Art und Weise gedruckt werden kann (Buchstaben, Ziffern, Sonderzeichen).

Unterlänge: Unterster Punkt eines Zeichens.

Vektorfont: Bei diesen auch unter dem Namen »Outlinefonts« bekannten Zeichensätzen stehen die Einzelzeichen nicht Punkt für Punkt fertig im Speicher. Vielmehr sind hier für jedes Zeichen die relativen Koordinaten der Eckpunkte angegeben; Vektorfonts werden also bei jedem Anzeigen des Zeichensatzes neu berechnet. Vorteil: Bei einer Vergrößerung bleibt die Qualität vollständig erhalten, da sämtliche Zwischenpunkte mitberechnet werden. »Publishing Partner« verwendet Vektorfonts bei der Druckausgabe; die Fontgröße kann in Schritten von mindestens 0,02 Punkten verändert werden.

Witwe: Die erste Zeile eines Absatzes, die am Seitenende steht. Im Gegensatz dazu bezeichnet man die letzte Zeile eines Absatzes, die am Seitenanfang steht, als »Waise«.

Wortdurchschuß: Freiräume zwischen einzelnen Worten einer Zeile. Durch Veränderung des Wortdurchschusses ist der Blocksatz möglich.

Zeichendurchschuß: Auch als »Laufweite« oder »Spationierung« bekannt. Damit sind die Freiräume zwischen den einzelnen Zeichen einer Zeile gemeint. Durch Veränderung des Zeichendurchschusses ist der Blocksatz möglich.

Zeilendurchschuß: Dieser Begriff bezeichnet den Abstand zwischen zwei Textzeilen und wird in Punkten angegeben.

AMIGA DOS

Blitz ☆ licht

Name: Publishing Partner (Version »Master«)

Hersteller: Upgrade Editions, Paris

Quelle: DTM
Poststraße 25
6200 Wiesbaden
Tel. 06121/502050

Preis: 798,- DM

Positiv:

- Handbuch und Programm in Deutsch
- exzellente Druckausgabe
- große Funktionsvielfalt
- vielseitiger Text- und Grafikimport

Negativ:

- Handbuch stellenweise irritierend
- geringe Geschwindigkeit
- keine Nutzung der Amiga-Fonts möglich
- hoher Preis



Es lebe die DFÜ...



Nun ist es endlich soweit. Die AMIGA DOS ist über ein neues Medium zu erreichen – die DFÜ. Noch dazu können Sie über verschiedene Netze mit uns Kontakt aufnehmen. Wenn's also mal ein Problem gibt, und man nicht mehr weiter weiß, dann ruft man einfach – das A(MIGA-DOS)-Team!

Doch was soll die Mailbox bringen? Nun, zuerst einmal möchten wir besser erreichbar sein. Die Hotline ist ja schon mal eine schöne Einrichtung, aber halt nur einmal pro Woche, und dann auch nur 3 Stunden. Dazu kommt, daß man oft am Telefon nicht gleich die passende Antwort parat hat, weil man dazu eventuell den Autor eines Artikels fragen oder verschiedene Unterlagen wälzen muß. Somit kann die Antwort eine kleine Weile auf sich warten lassen.

Mit der Mailbox haben wir jetzt die Möglichkeit, die ganze Woche Fragen entgegenzunehmen und sie, mit Antwort, in lokalen Brettern abzulegen, so daß andere Leute mit ähnlichen Fragen eventuell sofort eine Antwort finden (natürlich werden auf Fragen aus einem Netz die Antworten auch über das Netz geschickt).

Doch wie sieht das in der Praxis aus? Zu diesem Zweck wollen wir erst mal von einem lokalen User ausgehen. Nach dem Einloggen als »Gast« hat man Zugriff auf verschiedene Bretter in der Mailbox. Wichtig sind dabei die Bretter, die sich im Verzeichnis »AmigaDOS« befinden. In dieses Verzeichnis wechselt man mit dem Befehl »brett/amigados«. Nun hat man folgende Bretter zur Auswahl:

Helpline – für Fragen und Tips zu Themen der AMIGA DOS

Gameline – für Fragen und Tips zu Spielen

Bei Problemen mit Programmen oder Fragen zu Texten der AMIGA DOS sollte man diese im Brett »Helpline« stellen. Dazu gibt man zunächst »b help« ein (das Mailbox-Programm erlaubt es, alle Befehle und Brettnamen abzukürzen, hier würde auch schon »b h« ausreichen), danach befindet man sich in dem Brett (was durch den Prompt »AMIGA DOS/Helpline >« zu erkennen ist).

Nun kann man mit »inhalt *« alle, oder mit »inhalt« alle neuen Nachrichten auflisten. Wenn ein Betreff interessant scheint, so kann man die Nachricht mit »lesen <nummer>« lesen, »lesen *« zeigt alle Nachrichten im Brett an.

Selber schreiben kann man eine Nachricht mit »senden«. Danach wird man zunächst nach dem Betreff, anschließend nach der Art der Nachricht gefragt. Hier gibt man normalerweise »t« für eine Text-Nachricht ein. Jetzt befindet man sich in einem Zeilen-Editor, den man mit ».« beenden kann. Abbrechen kann man den Sende-Vorgang mit »q« (eine vollständige Hilfe zum Editor ist mit ».?« abrufbar).

Möchte man auf eine Nachricht antworten, so geht dies mit dem Befehl »antworten«. Dabei kann man entweder die Nummer der Nachricht gleich dazu angeben, oder aber man

gibt sie bei der Abfrage an. Das weitere Vorgehen ist dann wie auch beim Schicken einer Nachricht.

Für Probleme mit Spielen sollte nicht das Helpline-Brett, sondern »Gameline« verwendet werden. In dieses Brett gelangt man mit »b /amigados/game«. Hier werden wir, falls möglich, auch Lösungen zu Adventures und vieles mehr ablegen und Re-Posts von Netz-Brettern vornehmen.

Wer vorhat, die AMIGA-DOS-Mailbox öfters zu besuchen, kann auch mit »antrag« einen User-Antrag stellen. Dies hat den Vorteil, daß man auch Fragen stellen kann, deren Beantwortung ins persönliche Postfach geschickt werden, sofern diese nicht von allgemeinem Interesse sind.

Netze, Netze und kein Ende...

Die AMIGA-DOS-Mailbox ist an das Zerberus-Netz angebunden, ein in Deutschland recht gut verbreitetes Netz (zur Zeit etwa 120 Boxen). Innerhalb dieses Netzes kann man, unter Angabe eines Usernamens sowie der Zieladresse Nachrichten verschicken. Um beispielsweise eine Nachricht an uns zu schicken, kann man eine Nachricht an »sysop@amigados« schreiben, sie wird dann innerhalb von maximal 2 bis 3 Tagen bei uns ankommen.

Wenn Fragen zu bestimmten Artikeln vorliegen, so kann man sich unter Umständen auch direkt an den Autor wenden, eine aktuelle Liste aller in der Mailbox vertretenen Autoren ist in dem Info-Brett abrufbar.

Unsere Redakteure sind ebenfalls persönlich erreichbar. Sie können mit »senden <name> @AMIGADOS.ZER« eine Nachricht absenden. Unter <name> geben Sie einfach den Usernamen des Redakteurs ein. Im Brett »Info« erfahren Sie die Usernamen.

Um uns aus anderen Netzen zu erreichen, gibt es verschiedene Möglichkeiten, wobei man sich mit den jeweiligen SysOps der Gateways (=Boxen, die Mails zwischen zwei verschiedenen Netz-arten umsetzen) in Verbindung setzen sollte: Aus dem Bitnet/Usenet/Subnet kann man sich an »root@fulmin.zer.sub.org« wenden. Für den Austausch zwischen Zerberus und Fido/Maus ist das Gateway ZERMAUS zuständig, das bei der Infinet läuft.

Die Parameter der AMIGA-DOS-Mailbox:

056 51-80 93 45
300/1200/2400 8N1

Online:

Mo.-Fr. 20 Uhr bis 8 Uhr
Sa.-So. 24 Stunden

Viel Spaß im Netz wünscht Ihre
AMIGA-DOS-Redaktion!

(G. Glendown/vb)



DIGI TOWER mit Power

Professionelle Anwendungen soll der DIGI TOWER ermöglichen. Wir haben eine Vorabversion für Sie getestet.

Das neueste Produkt der Firma ALCOMP ist ein Video-Digitizer mit dem Namen DIGI TOWER. Dieser treffende Name bezieht sich auf das Gehäuse des Geräts, das als "Tower" hochkant auf den Arbeitsplatz gestellt wird. Der Digitizer verarbeitet Bildsignale in Echtzeit und Echtfarbe, wobei bis zu 16,7 Millionen Farben möglich sind. Vom Amiga sind intern bis zu 4096 Farben darstellbar, diese jedoch in sämtlichen Auflösungen inklusive High-Resolution. Auch der Extra-Halfbright-Modus wird unterstützt. Die Hardware des »DIGI TOWER« verfügt über ein austauschbares Interface, das zur Zeit für den Amiga und IBM PC (und kompatible) erhältlich ist. Es sind jedoch weitere Interfaces sowie eine Fernbedienung geplant.

Modularer Hardware-Aufbau

Die Hardware zum Umwandeln der analogen Bildsignale

in digitale Werte ist modular aufgebaut. Der DIGI TOWER besteht aus maximal drei dieser Module, den "Wandler-Units", die jeweils 1198,- DM kosten. Jede dieser Wandler-Units ist für einen Farbanteil des Signals zuständig und kann mit einer Speichererweiterung ausgerüstet werden, die ein Digitalisieren in Echtzeit und Echtfarbe ermöglicht. Mit einem einzigen Wandler und dem RGB-Regler ist es bereits möglich, Schwarzweißbilder zu erzeugen. Die RGB-Einheit wird zum Preis von 230,- DM angeboten, die RAM-Erweiterung kostet pro Wandler-Unit 398,- DM. Mit drei Wandler-Units, den entsprechenden RAM-Erweiterungen und der RGB-Unit ist erst ein Digitalisieren in Echtzeit und Echtfarbe möglich.

Jede Wandler-Unit verfügt über Regler, mit denen die Helligkeit, der Kontrast und die Farbsättigung getrennt geregelt werden können. Eine Synchronisierung des Video-signals ist nicht nötig, denn der

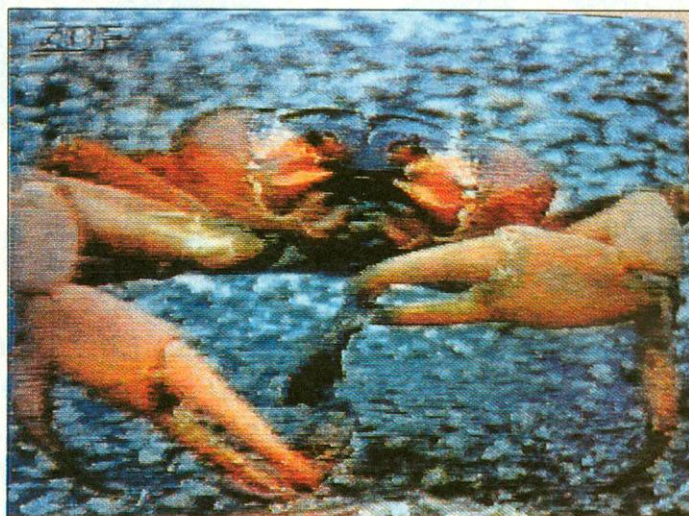


Bild 2. Überzeugende Qualität der Bilder

Digitizer synchronisiert sich automatisch. Des weiteren verfügt DIGI TOWER über einen eingebauten LCD-Farbmonitor mit gestochen scharfem Bild, der als Kontrollmonitor eingesetzt wird, sowie eine eigene Stromversorgung.

Umfangreiche Bildmanipulation durch »AMIGA-vISION«

Die Steuerungssoftware des DIGI TOWER, »AMIGA-vISION« genannt, bietet dem Anwender ein breites Spektrum an Manipulationen vor, während und nach dem Digitalisieren. So können die RGB-Werte des Bildes nochmals einzeln abgeglichen werden. Weiterhin können Teile eines bereits digitalisierten Bildes neu eingelesen oder im Genlock-Verfahren zusammengeblendet werden. Mit dem JAM-Effekt kann ein Bild nach unten hin aufgehellt werden, damit ein zweites Bild im Genlock-Verfahren digitalisiert und gemischt werden kann. Dies sind nur einige wenige Beispiele für Bildmanipulationen, von denen »AMIGA-vISION« eine Menge parat hält.

Ausdruck vom Feinsten!

Für die Ausgabe auf einem Drucker benutzt »AMIGA-vISION« eigens dafür programmierte Druckertreiber, die eine höchstmögliche Qualität bie-

ten. So sind auf 9-Nadel-Druckern 16 Grautöne, auf 24-Nadlern und Laserdruckern sogar bis zu 64 Grautöne möglich. Alle diese Qualitäten fordern selbstverständlich ihren Tribut: »AMIGA vISION« benötigt mindestens 1 MByte Rechnerspeicher.

Aufgrund des hohen Preises eignet sich der DIGI TOWER wohl weniger für den "Heim-Gelegenheitsanwender". Professionelle User bekommen hier jedoch ein Werkzeug, wie es besser nicht sein könnte. Die Bilder auf dieser Seite sprechen eine sehr deutliche Sprache!

(tb)

AMIGA DOS

Blitz ☆ licht

Name: DIGI TOWER

Funktion: Video-Digitizer

Quelle: ALCOMP Computerhardware GmbH
Glescherweg 22
5012 Bedburg
Tel. 02272-2093
Fax. 02272-1580

Preis: 1-Kanal + RGB-Unit
1498,- DM,

Vollausbau 3498,- DM

Positiv:

- alle Auflösungen
- hohe Wiedergabequalität

Negativ:

- hoher Preis

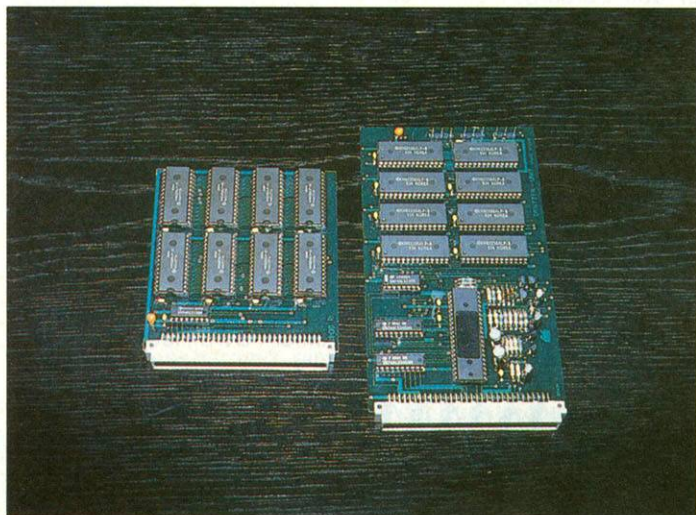
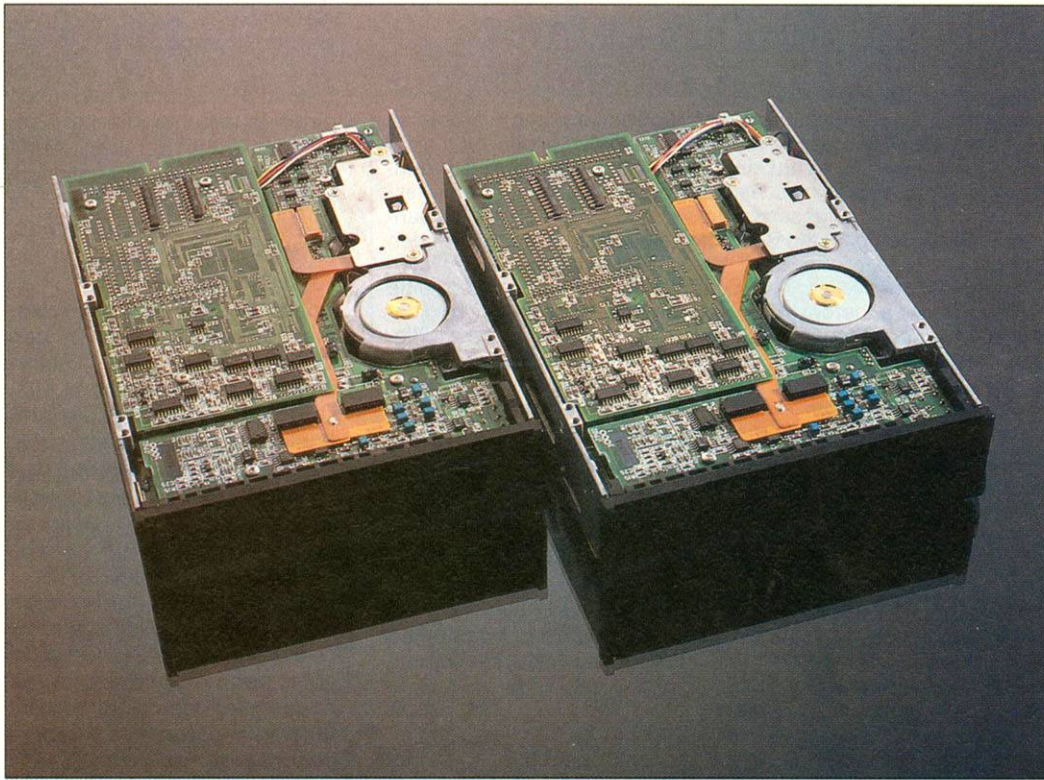


Bild 1. Ein Wandlermodul mit Speicherkarte



Garry Glendown

Streamen, was das Band hält...

“Virus, Virus, gib mir meine Dateien wieder!“. Ein Aufschrei, der wohl das ganze Leid eines Amiga-Besitzers widerspiegelt. Eine zerstörte Festplatten-Partition war das letzte, was man vom Virus sah.

Er hatte zugeschlagen, die 40MB-Platte war (nach Monaten) endlich mal wieder leer. Und das Backup? Ja, in der Anfangszeit hatte man öfters ein Backup gemacht. Aber mit der Zeit wurde diese allwöchentliche Prozedur, jedesmal an die 40 bis 50 Disks einzulegen, eine Minute zu warten, Diskette zu wechseln, ganz einfach zu dumm.

So kam es, daß mit der Zeit auch die letzte Backup-Diskette für neue Fish-Disks “geopfert” wurde. Wie groß war nun der Wunsch, doch weiterhin ein wöchentliches Backup gemacht zu machen. Doch jetzt war es zu spät... Haben Sie sich auch schon einmal so gefühlt? Es muß ja nicht gleich die ganze Platte gewesen sein,

und auch nicht wegen eines Virus, aber die Grundlage ist dieselbe: Man braucht ein Backup. Und während man bei einer kleinen Platte noch mit Disketten arbeiten kann, so wird es ab 60 oder 80 MB doch schon recht problematisch, immer ein halbwegs aktuelles Backup zu haben.

Zurück zu den Wurzeln oder Bändern

Doch es kommt die Rettung in der Not: das Backup-Medium der Unix-Welt, der Streamer, ist nun auch für den Amiga verfügbar.

Wir hatten zum Testen ein Exemplar des A.L.F. FileRunner-

Streamers. Er verfügt über eine Speicherkapazität von 150MB pro Tape, wobei verschiedene Aufzeichnungsformate möglich sind. Geliefert wird das Gerät in einem stabilen Metall-Gehäuse, wobei trotz des großen Innenraums und der guten Belüftung nicht auf einen (leisen!) Lüfter verzichtet wurde, was eine große Lebenserwartung des Netzteils und des Streamers garantieren dürfte.

Die Standfläche ist etwas kleiner als die eines A3000, somit besteht auch kein Problem, ihn auf einen A1000 oder A2000 zu stellen. Sollte jedoch ein anderer Standort gewählt werden, so würde man sich wünschen, daß es auf die Breite des Gerätes (5 1/4 Zoll) reduziert würde.

Für den A.L.F. FileRunner-Streamer wurde bei unserem Testgerät ein Caliper-Modell verwendet, der über eine Frontklappe verfügt. Damit wird er wohl nicht so leicht im Inneren verstauben, wie es bei den gängigen Streamern anderer Hersteller der Fall ist.

Getestet haben wir das neue Backup-Medium an einem A3000, um so zu sehen, ob sowohl Hardware als auch Software mit dem im 3000 eingebauten SCSI-Controller klar kommen. Dabei fiel auf, daß es offensichtlich zumindest eine ältere Version des A3000 gibt, bei dem in Verbindung mit dem Caliper-Streamer Probleme auftreten. Diese sind aber nicht auf die restliche Hardware zurückzuführen, sondern auf das Zusammenspiel zwischen dem Controller und dem Testgerät. Im Test lief das Gerät an einem A3000 einwandfrei, während ein anderer Rechner “gleichen Kalibers” die Festplatte nicht mehr erkannte. In einem solchen Fall muß ein anderer Streamer eingebaut werden (im Test lief der A3000 dann mit einem Wangtek-Modell). Sollten Probleme in dieser Richtung auftreten, so kann man sich direkt an den Anbieter wenden, BSC kann neben dem Caliper- auch Wangtek- und Teac-Streamer liefern.

Die Installation der Software läuft recht einfach. Es werden auf der Diskette mehrere Programme mitgeliefert, die zum einen die Software auf die Festplatte kopieren und danach die »Mountlist«, sowie die »Startup-Sequence« modifizieren. Es muß lediglich festgelegt werden, welches SCSI-Device verwendet wird, sowie unter welchem Namen der Streamer ansprechbar sein soll.

Ansonsten läuft alles, auch unter Kickstart 2.0, automatisch.

Legen wir los

Bei der Software handelt es sich um die bekannte Alf-Backup-Software, die auch schon bei den Alf-Festplattentreibern mitgeliefert wird. Sie ist aus diesen Bereichen schon als sehr zuverlässig bekannt. Beim Sichern hat man die Möglichkeit, zwischen verschiedenen Backup-Arten zu wählen. Sichert man die Platte in einzelnen Files, so kann



Liebe und Zuversicht statt Haß und Resignation

Die Chancen in dem so reich gesegneten Land Brasilien sind für die Menschen extrem ungerecht verteilt. Wenigen unvorstellbar reichen Grundbesitzern stehen Millionen Landlose gegenüber. Sie sind ständig auf der Suche nach einem Platz zum Überleben. Oft endet die Hoffnung auf eine Existenzmöglichkeit in den Slums der Großstädte. Der Teufelskreis von Unwissenheit, Arbeitslosigkeit und Elend läßt hier viele Familien zerbrechen. Dann müssen die Frauen ganz allein die Verantwortung für ihre Kinder tragen. Kein Wunder, wenn Haß und die Bereitschaft zur Gewalt wächst. Ein ökumenisches Zentrum engagierter Christen möchte dieser Entwicklung durch die befreiende Botschaft der Bibel entgegenwirken. In vielen Kursen und gut verständlichem Begleitmaterial leiten sie ihre Gemeindeglieder zum Bibelstudium an. Ganz wichtig ist die ständige Aus- und Weiterbildung von Mitarbeitern, die gemeinsam mit den Armen in den Elendsquartieren der Großstädte leben, ihre Sprache sprechen, ihre Nöte und Verzweiflung kennen... verschüttete Hoffnungen neu wecken können.

BROT FÜR DIE WELT
Spenden helfen bei der Finanzierung des ökumenischen Zentrums für Bibelstudien und seinen vielfältigen Aufgaben.
Postf. 10 1142 · 7000 Stuttgart 10

Brot für die Welt

KONTO · 500 500 500
Postgiro Köln oder Banken und Sparkassen. Bei vielen Kreditinstituten liegen vorgedruckte Spendenzahlscheine aus.

TEST

AMIGA DOS-Info

Thema: Streamer

Mit steigender Plattenkapazität im Workstation- und PC-Bereich wurde es nötig, ein Backup-Medium zu haben, bei dem man nicht nur 500 oder 800 KByte auf einen Datenträger unterbringen konnte. Während Streamer als Backup-Medium zunächst im professionellen Bereich eingesetzt wurden, sind sie in den letzten Jahren auch für den privaten Bereich interessant geworden. Bei den heute recht weit verbreiteten 5 1/4-Zoll-Geräten werden die Daten auf neun Spuren (allerdings nicht alle auf einmal, sondern eine Spur nach der anderen) aufgezeichnet. Das Band selber wird dabei durch einen patentierten Antrieb mit konstanter Geschwindigkeit am Kopf vorbeigeführt. Außerdem werden nicht die Spulen angetrieben, sondern das Band wird von einem Gummireif bewegt und von diesem auch immer fest aufgespult. Dadurch kommt es nur äußerst selten zu einem Bandriß.

Streamer werden nicht nur deswegen benutzt, weil sie eine höhere Datenkapazität haben, sondern auch, weil sie erheblich zuverlässiger sind als andere Datenträger. Selbst längere (ordnungsge-
mäßige!) Lagerung führt kaum zu Fehlern.

Zur Zeit geht der Trend auf dem Unix-Rechnern zu neuen Modellen: den Giga-Tapes. Hierbei wird auf Bändern, die den Video-8-Kassetten entsprechen, bis zu 2 Gigabyte abgespeichert. Da auch diese Laufwerke nach dem SCSI-Standard anzusprechen sind, ist es nur noch eine Frage der Zeit, bis auch sie für den Amiga angeboten werden.

man später einzelne defekte oder gelöschte Files zurückholen. Dieses Backup ist jedoch bei der Sicherung zeitaufwendiger. Im Gegensatz dazu kann man die Platte auch nach Tracks sichern, wobei wahlweise alle oder nur die belegten Blöcke gesichert werden. Diese Art des Backups hat zwar den Vorteil, schneller zu sein, doch muß man bei einem »Restore« die komplette Platte wieder einspielen, wobei der alte Inhalt verlorengeht. Somit eignet sich so ein Backup nur als Sicherheitskopie für den Fall, daß einmal die Platte defekt ist. Jedoch ist hier weiterhin zu beachten, daß die Platte, auf die man das Backup zurückspielt, dieselbe Größe und Parameter haben muß. Andernfalls kann das DOS die restaurierten Daten nicht mehr erkennen.

Wird der File-Modus ausgewählt, so wird die komplette Platte (oder auch nur ein Directory-Baum) nach den Einträgen durchsucht. Danach kann man auswählen, welche der Files oder Direktories man nicht haben möchte. Gut gelöst ist dabei, daß man einen Directory-Baum durch Anklicken des Namens selektieren oder deselektieren kann, und nicht wie zum Beispiel bei Quarterback erst in das Directory wechseln muß. Ist man mit der Auswahl zufrieden, dann kann man das Backup starten. »ALFstreams« sichert nun

nacheinander alle Files auf das Band. Sollte der vorhandene Bandspeicher nicht ausreichen, so wird man am Ende des Bandes aufgefordert, das nächste einzulegen.

Natürlich kann man nicht nur ein Backup auf ein einzelnes Band machen. Werden weitere Backups mit »Erweitertes Sichern« angehängt, so ermöglicht es ein zusätzlich zu dem eigentlichen Backup geführtes History-File, die verschiedenen Backups, die sich auch über mehrere Bänder erstrecken können, schneller aufzufinden.

Beim »Restore« kann man, zumindest im File-Modus, auswählen, welche der gesicherten Dateien man haben will. Diese werden dann recht zielstrebig vom Streamer »angefahren« und in akzeptablen Zeiten restauriert. Überhaupt ist die Geschwindigkeit des Gerätes recht gut.

Und wieder zurück

Auf dem A3000 ergaben sich Werte von etwa 40 KByte pro Sekunde, was etwa um den Faktor 3 schneller ist als ein Disketten-Backup (diese Werte entsprechen auch in etwa denen eines A2000 oder A500). Eine 100-MByte-Partition ist damit in weniger als einer Stunde gesichert – und das ohne etwa 120 Disketten

zu wechseln. Beim »Restore« werden, je nach der Anzahl der Files, ähnliche Werte erreicht. Selbst bei einem Testbackup einer langsamen Platte über ein (in Kürze auf den Markt kommendes) Billig-Netzwerk (Transfer-Rate des Netzes um 30 bis 40 KByte pro Sekunde) ergaben sich noch Werte um 15 KByte.

Sollten beim »Restore« Probleme auftreten, so hilft in den meisten Fällen ein »Retry« oder ein »Retention-Lauf«. Und auch schwerwiegendere Fehler können oft noch, trotz anderslautenden Warnungen im Handbuch (»Bei einem Fehler ... kommt jede Hilfe zu spät«), behoben werden. Alles in allem macht die von BSC gelieferte Hard- und Software einen sehr guten Eindruck. Daß das gesamte System auch auf einem neuen System mit neuer Kickstart läuft, spricht für den Programmierer beziehungsweise Entwickler. »ALFstreams« ist ein empfehlenswertes Produkt, das man meist schon nach dem ersten Backup, spätestens aber nach dem nächsten Plattenfehler nicht mehr hergeben möchte. Der Streamer ist übrigens nicht nur am Amiga 3000 nutzbar. Jeder Amiga, der über eine SCSI-Schnittstelle verfügt, kann mit einem solchen Gerät zusammenarbeiten.

(jb)

AMIGA DOS

Blitz ☆ licht

Name: A.L.F. FileRunner-Streamer

Quelle: BSC Büroautomation GmbH, Lerchenstr. 5, 8000 München 50, Tel.: 089/3544962

Preis: Kompletgerät mit Software, intern: 2190,- DM, Kompletgerät mit Software, Gehäuse, extern: 2590,- DM, ALFstreams Software: 298,- DM, Gehäuse mit Netzteil etc.: etwa 350,- DM

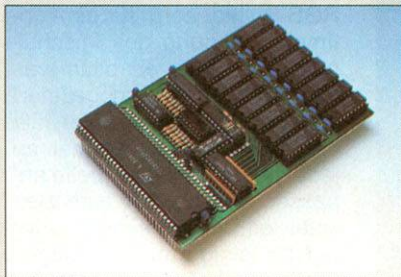
Positiv:

- schnelles und sicheres Backup großer Datenmengen
- gute Verarbeitung
- intern und extern erhältlich

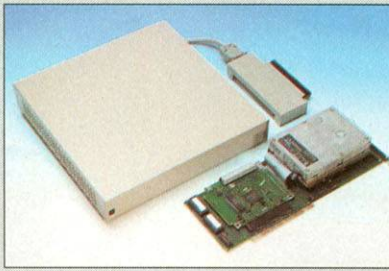
Negativ:

- nichts feststellbar

ROSSMÖLLER...



A 8MB/1000 598 DM
8 MByte Ramkarte; mit 2 MByte bestückt
kinderleicht einzubauen, nicht löten – nur einstecken
abschaltbar, ohne Waitstates



3 1/2" Disk-Drive extern 239 DM
5 1/4" Disk-Drive extern, 40/80 Tracks 279 DM
Unsere Floppy-Stationen sind ausschließlich mit Qualitäts-
laufwerken der Firmen NEC und TEAC ausgestattet. Alle
Laufwerke sind abschaltbar; der Bus ist durchgeführt
zum Anschluß weiterer Drives.

Filecards für AMIGA® 2000 und externe Festplatten für AMIGA® 500
mit Metallgehäuse und eigenem Netzteil. Selbstverständlich sind
unsere Festplatten autobootend, ALF-formatiert und ALF-installiert!

Amiga 500/1000		Amiga 2000	
20 MB	998 DM	20 MB 40ms	898 DM
30 MB	1089 DM	30 MB 40ms	998 DM
40 MB 28ms	1298 DM	49 MB 40ms	1398 DM
50 MB 40ms	1398 DM	40 MB 19ms SCSI	1498 DM
60 MB 28ms	1698 DM	60 MB 28ms SCSI	1598 DM
A1000 – Aufpreis 120 DM		83 MB 28ms SCSI	1798 DM

PREISHIT

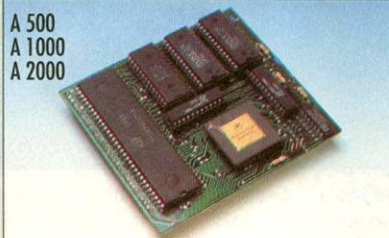
A512/500 99 DM

512 kByte Speichererweiterung mit Uhr. Höchste
Ganggenauigkeit durch OKI Uhren-IC mit integriertem
Quarz. Benötigt durch MBit-Technologie ca. 80%
weniger Strom! Abschaltbar, autokonfigurierend.

A2MB/500 449 DM

Die 2 MByte Ramkarte mit akkugepufferter
Echtzeituhr läuft sowohl mit dem alten FAT AGNUS als
auch mit dem neuen BIG AGNUS (umschaltbar).
MBit-Rams für niedrigsten Stromverbrauch. Sie hat
0-Waitstates, ist autokonfigurierend und abschaltbar.

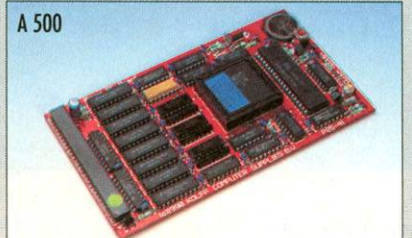
Testurteil „AMIGA DOS“ 2/90: „Sehr gutes
Zusatzgerät, keine Negativpunkte“.



TORNADO 498 DM

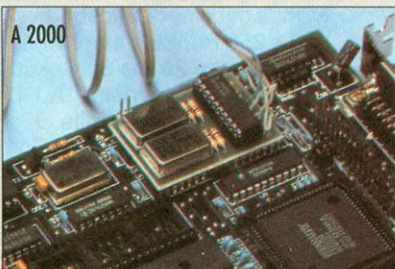
Jetzt geht für Ihren AMIGA® 500/1000/2000
die Post ab:

TORNADO macht ihn doppelt so schnell!
Und mit Arithmetik-Prozessor (optional) ist
Ihr AMIGA® sogar bis zu 20x schneller!



Power PC-Board 798 DM

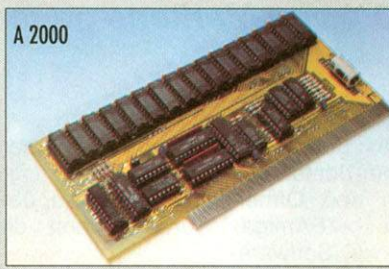
Der Geheimtipp: Ihr AMIGA® 500 IBM-kompatibel!
Alle AMIGA® Ein- und Ausgänge werden unterstützt:
Maus, Joystick, int.+ext. (auch 5,25 Zoll) Laufwerke,
par./ser. Schnittstelle, 1 MByte Ram und Uhr auch für
den AMIGA® 500 ansprechbar, NORTON SI-Faktor 3,3



TURBO-XT 199 DM

macht die Commodore XT-Karte doppelt so schnell
(XT-Karte nicht im Lieferumfang)

XT-RAM 768 k 298 DM
erweitert Ihre XT-Karte ON BOARD auf 768 KByte!



A 8MB/2000 549 DM

8 MByte Ramkarte, 2 MByte bestückt;
abschaltbar autokonfigurierend, 0 Waitstates,
4 MBit-Technologie.

Jetzt auch 68020/30-Kompatibel (A 2630)

Für AMIGA® 500/1000/2000

BTX-Interface für MULTITERM-Software	138,00 DM
BTX-Interface für COMMODORE-Software	138,00 DM
VESUV Eprom-Programmiergerät für die Eproms 2716-27512, 27513, 27011	199,00 DM
"HAPPY" 3/89 Gesamturteil "SEHR GUT"	
DigiSmooth Grafiktablett f. A500/1000	748,00 DM
Auflösung 1000 Pixel/Zoll; f. A2000	698,00 DM
Medusa, Atari ST-Emulator f. A500/2000	498,00 DM
Soundsampler, Profi-Modell	89,00 DM
MIDI-Interface für höchste Ansprüche	94,00 DM
Lightpen für Sculpt 3D/4D, DPaint etc.	79,95 DM
3-fach Kickstart-Umschaltplatine	59,95 DM
TRACK-DISPLAY 2000 für 2 HD's + 2 FD's	168,00 DM

AMIGA
Köln 90

Halle 10, Stand 1023

ROSSMÖLLER
COMPUTER TECHNIK

Rossmöller Handshake GmbH · Neuer Markt 21 · 5309 Meckenheim · Tel. 02225/2061-62-63

...WAS SONST!

Innovativ, ausgereift, für preisbewußte Anwender – das ist Rossmöller Computer Technik. HiTec-Qualität made in
Germany (1 Jahr Garantie), die mit jedem Geldbeutel kompatibel ist. Mit perfektem Service: So stehen Ihnen an
der Telefon-Hotline von Montag bis Freitag, 16-17 Uhr, die Entwickler unserer AMIGA-Produkte Rede und
Antwort. So nehmen wir Ihre telefonischen Bestellungen an 365 Tagen im Jahr; rund um die Uhr unter der
Nummer 0203/5195130, persönlich entgegen! Na, überzeugt? Dann fordern Sie unseren Gratiskatalog an!



Patric Hofman

Video-Digitizer VD 4 Amiga

Obwohl alle Video-Digitizer grundsätzlich die gleiche Arbeit verrichten, gibt es immer wieder Unterschiede. Wir haben uns diesmal den VD 4 von Merkens genauer angesehen.

Auf den ersten Blick präsentiert sich der Digitizer eher unscheinbar, nämlich in einem schlichten amigafarben-gespritzten Kunststoffgehäuse mit den Abmessungen 240x160x35 mm. Auf der Front befinden sich zwei Cinch-Buchsen für Video beziehungsweise Chroma-Eingang, außerdem ein Umschalter für Drucker oder Digitizer-Betrieb und eine LED für die Betriebsanzeige. Zum Lieferumfang gehört ein externes Netzteil mit einer Ausgangsspannung von 15 Volt und ein Flachkabel für den Anschluß an den Parallelport des Amiga. Der Parallelport ist glücklicherweise durchgeführt, so daß man den Drucker am Digitizer anschließen kann. Dies spricht für die Entwickler bei Merkens EDV, da andere Hardware-Entwickler solche "Kleinigkeiten"

offensichtlich für überflüssig halten. Nach dem Booten von der Workbench und Öffnen des Directories »VDAmiga« entdeckt man zwei Software-Versionen. Eine Schwarzweißversion mit der Bezeichnung »VD4Amiga« und eine Farbversion mit Namen »VD4Color«. Nach dem obligatorischen Doppelklick auf das Icon von »VD4Color« erscheint nach einer kurzen Ladezeit ein Filerequester mit dem Titel »Load Options«. Durch Anklicken des Gadgets »OK« werden die Voreinstellungen geladen. Nach kurzer Zeit erscheint ein neuer Bildschirm mit Bedienerleiste am unteren Bildschirmrand. Über diese können die wichtigsten Programmfunktionen wie Digitalisieren, Diskettenzugriff, Einstellungen und anderes erreicht werden. Außerdem sind sämtliche Programmfunk-

tionen auch über Pulldown-Menüs oder Tastatur ansprechbar. Wer über »AREXX« verfügt, kann die Software

auch darüber steuern. »VD4Color« arbeitet mit sämtlichen Auflösungen, einschließlich Overscan. Der interne RGB-Splitter sorgt für eine optimale Trennung des Mischsignals, das von Farbkameras oder Videorecordern geliefert wird. Über Schieberegler läßt sich der Digitizer optimal an jede denkbare Videoquelle anpassen. So genügt zum Beispiel ein Videorecorder, der nur ein einigermaßen ruhiges Standbild liefert, um nahezu identische Bilder zu erstellen. Allerdings muß man schon einiges an Zeit investieren, wenn man bestmögliche Ergebnisse erzielen will. Vor allem das Anpassen an die Videoquelle gestaltet sich unter Umständen zu einem langwierigen Geduldsspiel, das aber durch immer bessere Bilder belohnt wird und schließlich zu dem bekannten "Aha-Effekt" führt. Als sehr positiv muß auf jeden Fall die hohe Geschwindigkeit des Digitizers erwähnt werden. Im S/W-Non-Interlace-Format dauert der Digitalisierungsvorgang 20 ms, im S/W-Interlace 40 ms, im Color-Format mit allen Farbtiefen (Non-Interlace) 60 ms, mit Interlace 120 ms. Die Rechenzeit für die einzelnen Colorbilder hängt vom eingesetzten Rechnertyp ab. Beim Einsatz eines "normalen" A2000 ergibt sich im HAM-Modus mit einer Auflösung von 320 x 256 Punkten eine Digitalisierungszeit von 20 bis 35 Sekunden, wobei doppelte Auflösung auch eine doppelte Digitalisierungszeit bedeutet. In S/W wird dagegen in Echtzeit digitalisiert. Bei der derzeitigen Version 3.53 der Farb-Software



Bild 1. Zwei Beispiele, die die Qualitäten des "VD 4" zeigen...



Bild 2. ...und sich mit wenig Aufwand realisieren lassen

sind noch nicht alle Möglichkeiten nutzbar. Geplant sind unter anderem eine "Floyd-Steinberg-Korrektur", wie auch eine "Gamma-Korrektur", beide Korrekturen ermöglichen Änderungen des Kontrastes und der Bildschärfe. Außerdem wird wohl in Zukunft das flimmern beim Betrieb im Interlace-Modus der Vergangenheit

angehören. Beim Abspeichern der Grafik hat man zwei Möglichkeiten: außer dem üblichen IFF-Format können die Daten im Rohformat ohne die sonst üblichen "Datenköpfe" abgespeichert werden, was als reine Datensicherung auf Diskette gut geeignet, da platzsparend ist. Für die Programmierer unter den Lesern dürfte

es interessant sein zu erfahren, daß sich auf der Diskette ein Verzeichnis »Routines« befindet, mit den Linkfiles und den »Include-Headers« zum Steuern des Digitizers beziehungsweise zum Konvertieren der Bildinformationen zu den amigaspezifischen Bildformaten.

Das Konzept des VD 4 Amiga wurde sowohl bei der Hardware als auch bei der Software gut durchdacht und umgesetzt.

Der Preis von zirka 1300 DM ist sicherlich kein Pappenstiel, aber man bekommt auch einiges dafür, nämlich außer einem wirklich hervorragenden Digitizer auch noch einen kostenlosen Update-Service, wenn gravierende Änderungen vorgenommen wurden. Bedingt durch die ständige Weiterentwicklung, fällt leider aus Kostengründen die Bedienungsanleitung etwas mager aus und wird durch »ReadMe-Files« auf der Diskette ergänzt. Sollte man einmal nicht weiterkommen kann man sich pro-

blemlos mit Merkens-EDV telefonisch in Verbindung setzen, was sehr hilfreich sein kann. Übrigens ist auch ein Anschluß für einen S-VHS-Recorder vorhanden. Mit dem VD 4 Amiga erwirbt man auf jeden Fall ein Gerät, das technisch alle Voraussetzungen erfüllt, um naturgetreue Bilder zu digitalisieren. (jb)

AMIGA DOS

Blitz ☆ licht

Name: VD 4 Amiga
Video-Digitizer

Quelle: Merkens EDV, Fuchstanzstraße 6a, 6231 Schwalbach, Tel. (06196)/3026

Preis: etwa 1300 DM

Positiv:

- gute Qualität der eingelese-
nen Bilder
- hohe Geschwindigkeit
- interner RGB-Splitter

Negativ:

- mageres Handbuch

• Grafiken in allen Amiga-Grafikmodi • TFMX-Sound • Multiplechoice-Kommunikationen • 6 Begleiter mit Funktion • Bedienung über Icons • Action-Spiele • diverse Sound-Effekte • Speicher-Möglichkeiten des Spielstandes

Fatal Heritage

DAS NEUE SUPERADVENTURE

Perfekte Grafiken
Topsound
intelligente Texte
spannende Handlung



An einem Tag wie jedem anderen bricht plötzlich das große Abenteuer über Dich herein.

Super: Eine südamerikanische Erbschaft; plötzlich und unerwartet. Aber: Das Testament verschwindet auf unerklärliche Weise.

Auf der weltweiten Jagd danach brauchst Du zuverlässige Freunde, denn Du bist nicht als Einziger hinter der Erbschaft, einer tropischen Plantage mit einem riesigen Goldschatz, her. Der Weg dahin ist mit gefährlichen Zwischenfällen gepflastert:

Permanent tödlich sitzt Dir "Das Kartell" mit seinen Helfershelfern im Nacken, total fatale Tücken lauern im kleinsten Detail, brutal heimtückische Bestien bringen Dich an den Rand des Abgrundes (oder auch einen Schritt weiter). Bezaubernd verführerisch wirst Du dubiosen Fahrten ausgesetzt, Terroristen machen Dir das Leben schwer, einige Begleiter machen sich

bei der ersten Gelegenheit aus dem Staube – andere helfen genau da, wo sie unentbehrlich sind.

Oft trägt der Schein und das grauenvolle Verhängnis ist nahe – better think twice – damit Du am Ende aus der "Fatal Heritage" als strahlender Erbe hervorgehst.

Nur DM 89,95

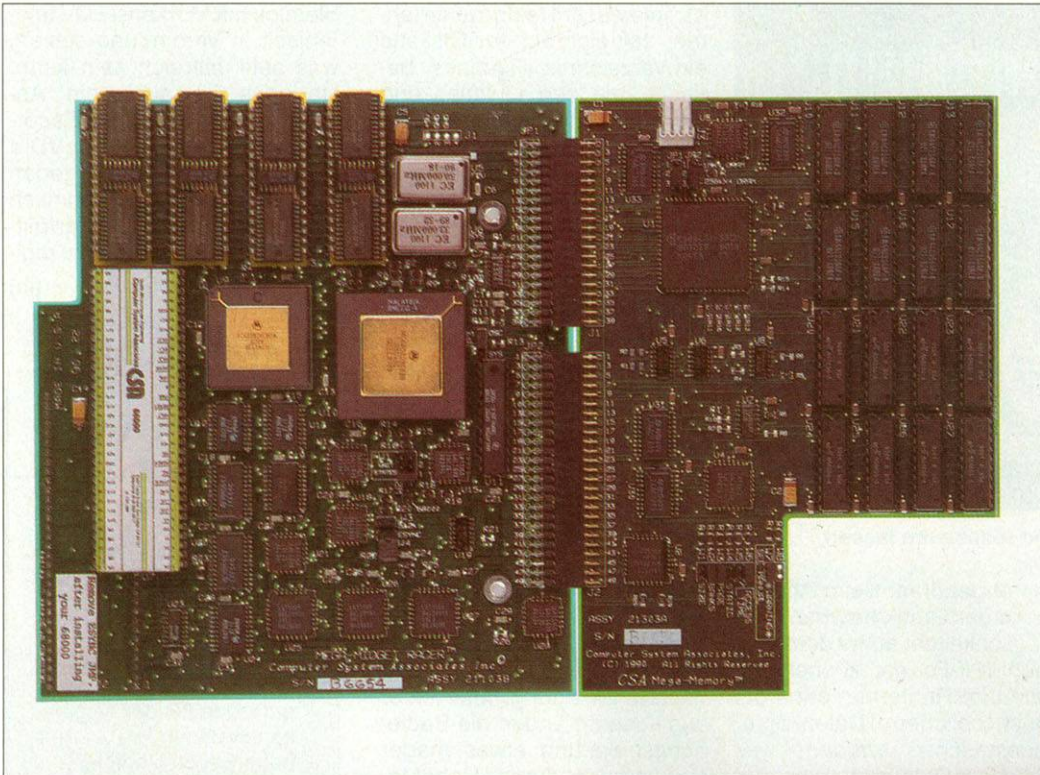
Für Amiga 500 • 1000 • 2000 • 2500 • 3000

Made by **EGD** & DELTA KONZEPT

Händleranfragen erwünscht

Delta Konzept GmbH • 4000 Düsseldorf 11 • Postfach 190 104 • Telefon 021 59/8 1075

COUPON
Anschritt: Delta Konzept GmbH, Postfach 190 104, 4000 Düsseldorf 11
Ich wage es und bestelle Spiel/e zum Preis von jeweils
89,95 DM per Nachnahme.
Name: _____
Eigene Anschrift: _____



Der Mega-Racer

Der 68030-Prozessor macht als großer Bruder des 68000 immer öfter von sich reden, aber bisher nur auf dem Amiga 2000. Mit dem Mega Midget soll dies anders werden.

Wer bisherige Turbo-Karten, basierend auf der Einheit CPU 68030 und FPU (Floating Point Unit) 68881/68882 einbauen wollte, hatte nur die Auswahl unter den Steckplatzkarten für "Amiga Großen", Besitzer eines A500 schauten da in die Röhre. Nun ist der Einsatz eines Turbo-Boards nicht gerade billig, aber sollte man deshalb den kleinen Amiga-Bruder außen vorlassen? Die Entwickler der Firma CSA aus dem amerikanischen San Diego waren sich wohl einig, daß dies kein Zustand war und entwickelten ein neues Turbo-Board, zwar mit den bekannten Komponenten bestückt, jedoch neu, was den Anschluß an den Rechner betrifft.

Das Turbo-Board für alle Amigas

Das Mega-Midget entpuppt sich als High-Tech-Gerät mit

sehr vielen hochintegrierten SMD-Bauteilen ("Surface Mounted Devices" oberflächenmontierte Bauteile; sind platzsparender und verkleinern die Träger, also die Platinen erheblich). Das Mega-Midget mit seiner viel schnelleren Prozessor-Kombination ersetzt den langsamen 68000 und übernimmt alle Funktionen. Und dieser Thematik gemäß wird das Board auch eingesetzt: Der 68000 wird aus seinem Thron (Sockel) entfernt und das Board an seiner statt eingesetzt. Zu diesem Zweck befindet sich ein Extra-Sockel auf der Platine, der in den Sockel des 68000 paßt. Das Board wird dann mittels zweier Abstandhalter auf der Hauptplatine befestigt. Der Vorteil dieser Konstruktion liegt klar auf der Hand, das Mega-Midget ist für den Amiga 500 wie auch für den Amiga 2000 einsetzbar.

Der eingebaute 68030 arbeitet asynchron und kann damit

eine Taktfrequenz von 20 bis 33 Mhz "fahren", die optional einzubauende FPU 68882 wird mit bis zu 50 MHz getaktet, wenn man ihr einen Extra-Quarz spendiert, ansonsten wird der Takt des Prozessors übernommen. Implementiert auf der Platine wurden ein CPU/FPU-Bus-Interface, ein SRAM-Interface um Zugriffe auf 32 Bit Fast-RAM zu ermöglichen und ein Expansion-Bus-Interface. Zusätzlich befinden sich noch ein Schaltkreis zur Kompatibilitätsbindung an den 68000 und eine Kontrollogik für den Cache-Speicher, in dem der Prozessor Befehls-Sequenzen zwischenspeichert, um schneller auf sie zurückgreifen zu können.

Um Programmen, die sich mit der Prozessorkombination 68030/68882 nicht so recht vertragen wollen, doch noch Chancen zu geben, wurde ein 68000 auf der Platine integriert, der mittels Software-

Umschaltung (auf der mitgelieferten Diskette) den Part des Alleskönners übernehmen kann – mit der gewohnten Geschwindigkeit versteht sich. Hier fragt man sich allerdings, was für eine Aufgabe der herausgehebelte, eigene 68000 übernehmen soll, und ob es nicht sinnvoller gewesen wäre, ihn auf der neuen Platine einzusetzen. Es gäbe zwei Gründe dafür:

- 1.) Die Platine wird etwas preiswerter, auch wenn ein "normaler" 68000 nur noch knapp 20 DM kostet.
- 2.) Wer das Mega-Midget einbaut, hat auch die Erfahrung, einem Prozessor einen neuen Platz zuzuweisen.

Als Kritik war das übrigens gerade nicht gedacht, nur als Überlegung, die beim Einbau der Platte auftauchte.

Nicht nur die Schnelligkeit der Prozessoren ist Trumpf beim Mega-Midget, sondern auch die Schnelligkeit der Datenübertragung. Das Mega-Midget verfügt über ein volles 32-Bit-Bus-System, das heißt, Adress- und Datenbus können also Daten im Format »Long-Word« (vier Byte) verarbeiten. Der 68030 verfügt zu diesem Zweck über ein eigenes 512 KByte großes High-Speed-RAM, in welches das ROM-Kernal (die System-Programme des Amiga) kopiert und ausgelesen werden können. Dieses kann auch unter Ausschluß der 68030-Unit geschehen, so daß fremde Software ungehindert Zugriff zum SRAM hat. Auch hierdurch sollen Kompatibilitätsprobleme weitestgehend vermieden werden.

Einbau in einen A500

Bei unserem Test mußte als "Opfer" ein A500 herhalten, dem man nach Öffnen des Gehäuses das Herz, also den Prozessor entriß. Das Einsetzen der Mega-Platine ist zwar recht schnell durchzuführen, erwartet vom Käufer aber ein gewisses Maß an Zutrauen zu sich selbst, denn Platinen dieses "Kalibers" sind bei Falschbehandlung schneller defekt, als man denkt. Zwei Platinen-Abstandhalter werden in vorgebohrte Löcher gesteckt; sie bieten der aufgesetzten Mega-Platine den nötigen Halt. Die Mega-Platine besteht eigent-

lich aus zwei Platinen, der eigentlichen Prozessor-Platine und einer über einen Stecksockel angeschlossenen RAM-Platine mit dem benötigten SRAM. Stabilität ist gerade hier beim A500 ein Wort, was unbedingt großgeschrieben werden muß, man sollte sich also sehr nach der mitgelieferten Einbauanleitung richten. Als nächstes ist es nötig, die FPU 68882, so vorhanden, an die richtige Taktfrequenz anzupassen. Dazu befindet sich ein Jumper (Steckbrücke) auf der Platine, der die FPU entweder mit dem Systemtakt des 68030 gleichschaltet oder den Taktimpuls eines externen Quarzes nutzt, in diesem Falle ist eine Taktfrequenz bis 50 Mhz möglich.

Eine im A500 eingebaute interne Speichererweiterung wirkt sich beim Einbau problematisch aus, sofern der Chip GARY auf einem Extra-Sockel sitzt und mit seinem "Hochsitz" die Präzision der Mega-Boards empfindlich stört. Durch die Baumaße des A500 ergeben sich nicht nur diese Probleme, auch das Oberteil des Gehäuses drückt ganz empfindlich an vielen Ecken und Kanten. Wer sich einen solchen Einbau in Eigenregie nicht zutraut, sollte unbedingt einen Fachhändler zu Rate ziehen.

Ist das Board dann doch an seinem Platz, und alles andere ebenfalls verstaut, kann der Start durch das Einschalten beginnen. Auf der mitgelieferten Diskette befinden sich diverse Utility-Programme, darunter einen Software-Schalter, der über einen Doppelklick den Wechsel zwischen 68000 und 68030 vornimmt. Bei Prozessorwechsel wird ein Reset erzwungen, da Programme, die a) nur mit dem 68000 laufen und b) gebootet werden müssen, sich hierbei am besten handhaben lassen. Außerdem befinden sich Tools zum Auslesen und Initialisieren des Cache-Speichers und zum Konfigurationstest, sowie das PD-Programm »Mandelbrot« aus der Fish-Reihe, mit dem fraktale Grafiken erzeugt werden können. Das Programm zeigt die Unterschiede in der Rechengeschwindigkeit zwischen dem "einsamen 68000er" und der Kombination CPU/FPU 68030/68882. Der Mega Midget Racer mit sei-



Ein Speedtest zeigt, wer hier das Sagen hat...

nem Mega Memory, so die offiziellen Bezeichnungen, ist an sich eine großartige Idee. Die Möglichkeit, mit einer 68030-Karte (fast) alle Amigas beglücken zu können, ist eigentlich Gold wert.

Lohnt sich der Kauf?

Etwas Sorgen dürfte die Größe der Platine und ihr Platzbedarf im A500 bereiten. Wer im Besitz einer internen Speichererweiterung ist, die Änderungen am System benötigt, wie zum Beispiel eine Neuselektierung des Gary-Chips oder Eingriffe am Prozessor-Sockel erfordert (wenn zum Beispiel 1 MByte Chip-RAM genutzt werden sollen), sollte den Einbau vom Fachhändler vornehmen lassen. Die Software ist zwar ausreichend, hier hätten jedoch ein paar Speed- oder Konfigurationstests, die Infos über den gerade aktiven Prozessor abgeben, noch auf Diskette gepaßt, wobei diverse PD-Programme bestimmt gut geeignet sind.

Ansonsten macht das Mega-Midget einen sehr guten Eindruck, auch was die Verarbeitung betrifft. Die Wärmeentwicklung der Chips ist allerdings hoch, so daß man es auf jeden Fall vermeiden sollte, die Lüftungsschlitze des Amiga zuzustellen.

68030-Power im Amiga ist Dank des Mega-Midget nun auch für jeden Amiga-Besitzer erreichbar geworden, auch wenn der Preis nicht gerade gering ist. Mit zunehmender Verbreitung allerdings dürfte

wohl auch hier irgendwann eine Änderung in Sicht sein. Wer angesichts solcher Peripherie für den A 500 immer noch von einem Spiel-Computer spricht, schaut allerdings an den Tatsachen vorbei.

Anzumerken ist noch eines: Unser Testgerät war eines der ersten Geräte überhaupt, somit sind die Mängel, sofern sie als solche benennbar sind, bei späteren Versionen des Mega-Midgets wahrscheinlich ausgeräumt, zumindest sollten sie von den Entwicklern schon erkannt worden sein.

(jb)

AMIGA DOS

Blitz ☆ licht

Name: Mega-Midget-Racer und Mega-Memory

Quelle: Pulsar, Erlanger
Str. 8-10, 5000 Köln 91,
Tel.: 0221/873359-873657-874 3-21

Preis: Mega-Midget-Racer mit 68030 und FPU ca. 1400 DM, Mega-Memory ohne Chips 740,- DM

Positiv:

- Einbau in A500 und A2000 möglich
- schnelles 32 Bit SRAM anschließbar
- FPU gesondert getaktet (bis 50 Mhz, optional)

Negativ:

- Einbau beim A500 nicht unproblematisch

CSV-HIGHLIGHTS

Commodore	
Speichererweiterung Commodore 1050 (256 kB)	39,-
Commodore Farbmonitor 1084 Stereo	589,-
Commodore Amiga 500	789,-
Amiga 500 + Farbmonitor 1084 S	1369,-
Speichererweiterung auf 1 MB mit Uhr	149,-
Commodore Amiga 2000	1669,-
Amiga 2000 + Farbmonitor 1084 S	2249,-
Amiga 2000 + AT-Karte mit 5,25"-Laufwerk + Festplatte WD Filecard 40 MB (29 ms)	3679,-
Amiga 3000 (16 MHz, 40-MB-Festplatte)	5799,-
3000 (25 MHz, 40-MB-Festplatte)	6789,-
3,5"-Zweitlaufwerk Amiga 2000 (Commodore)	199,-
PCXT-Karte mit 5 1/4"-Laufwerk	599,-
AT-Karte mit 5 1/4"-Laufwerk	1299,-
SCSI Controller Commodore A 2090 A	579,-
20-MB-Festplatte für Amiga 2000 mit SCSI Controller Comm. 2090 A (autobootend)	899,-
40-MB-Festplatte mit Controller 2090 A	1299,-
47-MB-Filecard autobootfähig, formatiert	1149,-
20-MB-Filecard (Seagate, 40 ms) für A 2000 mit PC-Karte oder A 1000/Sidecar	579,-
40-MB-Filecard (Western Digital, 29 ms)	799,-
50-MB-Filecard (Seagate, 40 ms)	949,-
2-MB-RAM-Erweiterungskarte für A 2000, aufrüstbar bis 8 MB	649,-
Atari	
Festplatte Vortex HD 40 Plus	899,-
Festplatte Atari Megaflo 30	869,-
1040 STFM + Monochrommonitor SM 124	1099,-
1040 STFM + SM 124 + Megaflo 30	1949,-
Atari STE + Monochrommonitor SM 124	1399,-
Atari Computer Mega ST 1 mit Maus + Monochrommonitor SM 124	1469,-
Mega ST 1 + SM 124 + Megaflo 30 MB	2319,-
Atari Mega ST 2 + Monochrommon. SM 124	2169,-
Atari Mega ST 2 + SM 124 + Megaflo 30	2999,-
Atari Mega ST 4 + Monochrommon. SM 124	3149,-
Atari Mega ST 4 + SM 124 + Megaflo 30	3999,-
Epsondrucker (dt. Handbücher)	
LX 400	399,-
LQ 400 (24-Nadel-Drucker)	599,-
LQ 550 (24-Nadel-Drucker)	749,-
Tintenstrahlendrucker LX 800 (9 Düsen, NLQ, max. 240 Zeichen/Sekunde)	449,-
Stardrucker (dt. Handbücher)	
LC 10 Color Farbdrucker mit Centronics	569,-
LC 24-10 mit Centronicsinterface	649,-
XB 24-10 mit Centronicsinterface	1279,-
NEC-Drucker (dt. Handbücher)	
Farboption P6 + P7 + 249,- EZB P 7 Plus	399,-
NEC P 60	1479,-
Colorit für NEC P 60 oder P 70	179,-
NEC Drucker P 2 Plus	649,-
NEC Farbmonitor Multisync 3 D	1369,-
NEU: Panasonicdrucker KXP 1124	649,-
IBM-kompatibler AT (12 MHz, 1 MB)	888,-
Druckerlabel 5 m lang für Amiga, ST	29,-
EGA-Karte (80x80) + 14-EGA-Farbmonitor	649,-
VGA-Karte 8 Bit 256 kB	169,-
VGA-Karte Optima 16 Bit, 512 kB	279,-
VGA-Karte Trident 16 Bit, 512 kB	249,-
Multisynch Farbmon. (0,28 mm, 1024x768)	949,-
VGA-Farbmonitor (0,28 mm, 1024x768)	729,-

Versandkostenpauschale: Inland DM 12,-, Ausland DM 40,- je Paket. Lieferung nur gegen NN oder Vorauskasse. Ausland nur Vorauskasse. Preise gültig ab 15.10.90.

CSV RIEGERT GmbH

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



**AMIGA 3000, 2500,
2000, 1000 und 500**

Was fällt *Ihrem* AMIGA Händler ein zu :

**FFS, OSA, BTX,
GVP, DTV, AREXX,
DTP, NIL, CON,
RAD, OFS, CLI und
WYSIWIG ???**

Wir wissen, wovon wir reden...



**Schiffgraben 19
3000 Hannover 1
Tel.: 0511/326736
Fax: 0511/3632210**

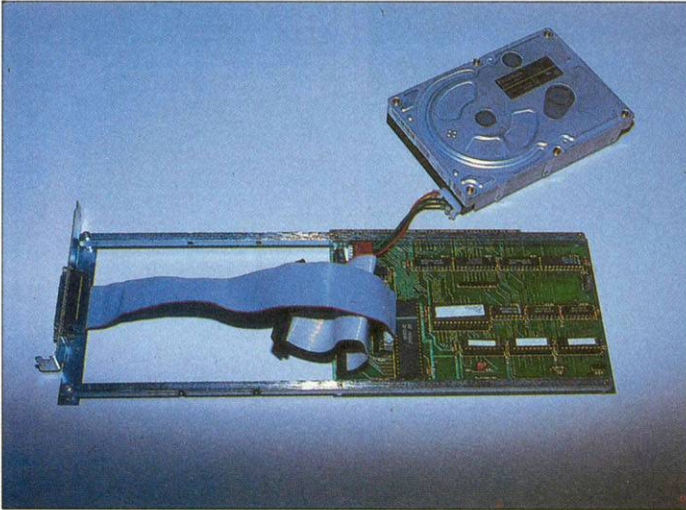


Bild 1. Die 1-Zoll-Harddisk

Kleiner Speicherriese

Kleine Platte, große Leistung? Die (räumlich) kleinste Festplatte von FSE will neue Maßstäbe setzen.

Die neueste "Kreation" der Firma FSE ist eine Festplatte für den Amiga 2000. Die Bauhöhe dieser Platte beträgt ein Zoll (2,54 cm). Trotz dieser geringen Maße verfügt die Platte über erstaunliche Leistungsdaten. So beträgt die mittlere Zugriffszeit 17 ms, unter Zuhilfenahme des 64 KByte großen Cache-Puffers sogar bis zu 11 ms. Die ermittelten Werte sind aufse-

henerregend. Unter »FastFile-System« und Kickstart 1.3 betrug die Datenübertragungsrate über 700 KByte/s, unter Kickstart 2.0 sogar bis zu 826 KByte/s. Die Platte verdankt ihre hohe Leistung zum Teil auch dem Interface, das die Daten parallel mit 16 Bit überträgt. Die Treibersoftware ist zum größten Teil ROM-resident, so daß weniger RAM verbraucht wird als bei herkömm-



Bild 2. Die Testdaten sprechen für sich!

lichen Treibern. Als Treiber dient wieder die FSE-Entwicklung »BOIL3«, die mit einer neuen Benutzeroberfläche versehen wurde. Mit dieser Oberfläche ist es nun möglich, die Harddisk fast komplett mit der Maus einzurichten.

wendungen an, wo große Datenmengen schnell von einem Rechner zum nächsten übertragen werden müssen. Für A500 und A1000 ist das Produkt ab November als AT-Busplatte verfügbar. (tb)

Garantiert gut

Überzeugt von ihrem Produkt ist die Firma FSE allemal, denn sie gewährt zwei Jahre Vollgarantie.

Wir sind uns sicher, daß diese Platte zu einer der begehrtesten Harddisks für den Amiga 2000 wird.

Neben dieser überragenden Festplatte bietet FSE ein Wechselplattensystem an, das inklusive einer 44-MByte-Cartridge 1678,- DM kosten soll. Jede weitere Cartridge kostet 200,- DM. Dieses Wechselplattensystem bietet sich vor allem bei kommerziellen An-

AMIGA DOS

Blitz ☆ licht

Name: Trinology Profile
1-Zoll
Funktion: Festplatte
Preis: 52 MByte 1148,- DM
105 MByte 1648,- DM
Quelle: Frank Strauß Elektronik, Schmiedstraße 11,
6750 Kaiserslautern, Tel:
0631/67096, Fax: 0631/60697

Positiv
– sicherer Betrieb
– sehr leise

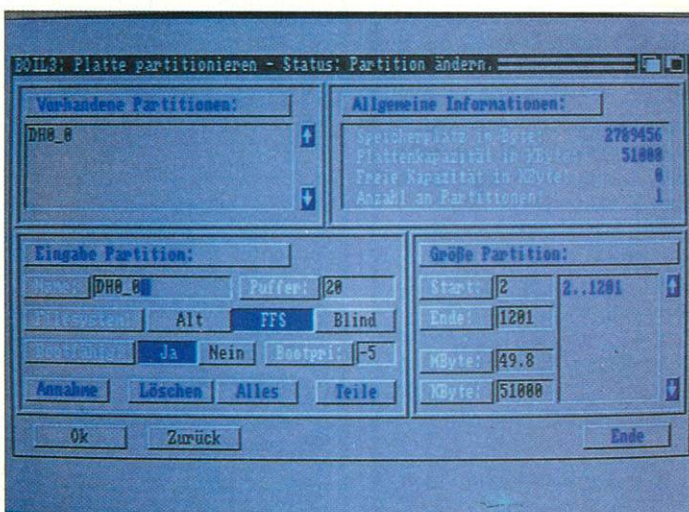


Bild 3. »BOIL3« mit neuer Oberfläche



Bild 4. Wechselplatte mit 44-MByte-Cartridge

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,-

neu

Strategiespiele in GFA-BASIC

Darstellung und Realisierung von Strategiespielen auf dem Computer und Entwicklung „intelligenter Computergegner“. Als Beispiel dienen vier Programme.

DM 99,-

**AMIGA
Köln 90**

Wir stellen aus
09.-11. November 1990
Halle 10.2, Stand 1005

GFA Systemtechnik GmbH
Heerdter Sandberg 30
D-4000 Düsseldorf 11

Tel. 02 11/55 04-0 · Fax 02 11/55 04 44

**GFA
SYSTEMTECHNIK**



Amiga, ich befehle dir...

“Lieber Amiga! Drucke mir bitte alle Ausgaben des Monats Mai aus...” – Wäre das schön, wenn unser elektronischer Wunderkasten jetzt täte, was gewünscht wurde. Aber leider...

Wenn man heute so vor seinem Amiga sitzt und per Maus die verschiedensten Programme aufruft, dann macht man sich gar keine Gedanken darüber, daß dieses “Zweimal-Mausklick-dann-starte-Programm” im Grunde genommen selbst ein Programm ist. Woher “weiß” aber der Amiga, was er tun soll, wenn Sie ihn mit der Maus traktieren? Wer schon Programmiererfahrung besitzt, für den wird das, was jetzt kommt, kalter Kaffee sein. Für Anfänger jedoch soll jetzt die Verständnisfrage beantwortet werden.

Schaut man unter das Gehäuse des Rechners, findet man nicht etwa ein Gehirn, oh nein, vielmehr kleine schwarze Kästchen mit vielen silbernen Beinen dran. Die sehen zwar lustig aus, aber denken können sie trotzdem nicht.

Woran liegt es also, daß der Amiga das macht, was sein Herrchen (Frauchen soll es in-

zwischen laut unbestätigten Gerüchten auch geben) von ihm will?

Im Schuldeutsch bedeutet das Wort “Programm” nichts weiter als eine Reihenfolge von Anweisungen, die nacheinander abgearbeitet werden. Recht haben sie damit, die Herren Lehrer. Am besten kann man sich dies an einem kleinen Beispiel vor Augen halten.

Nehmen wir an, Sie wohnen auf dem Lande, wo Ihnen Zeitung und Milch noch morgens vor die Tür gebracht werden. Nun könnte ihre morgendliche Tätigkeit wie folgt aussehen:

- 1) von der Küche zur Haustür,
 - 2) Zeitung dem Briefkasten entnehmen,
 - 3) von der Haustür in die Küche.
- Für die Milch dann der zweite Durchlauf:

- 1) von der Küche zur Haustür,
- 2) Milch vom Fußboden aufnehmen,
- 3) von der Haustür in die Küche.

Wenn Sie das in Wirklichkeit tun, sind Sie zwar reif für den

Psychiater, haben aber zwei Einzelaktionen durchgeführt. Dabei könnten Sie zwei Unteraktionen, nämlich Zeitung entnehmen und Milch aufnehmen hintereinander setzen und brauchten dann nur einmal zur Haustür und zurückzulaufen (wie es jeder normale Mensch sowieso täte).

Wir haben damit zwei Einzelaktionen zusammengefaßt zu einem Ablauf oder einem Programm.

Beim Amiga sieht das allerdings noch komplizierter aus. Wenn Sie mit der Maus ein Programm anklicken, geschieht folgendes:

- 1) Mausbewegung auslesen,
- 2) Mauszeiger setzen,
- 3) ist die linke Maustaste zweimal gedrückt? Wenn ja, dann zu Punkt “5”;
- 4) Wenn nicht, dann zu Punkt “1”;
- 5) schauen, welches Icon unter dem Mauszeiger liegt. Ist das Programm, dessen Icon geöffnet wurde, lauffähig?

Wenn ja, dann zu Punkt “7”;

6) Wenn nein, dann Fehlermeldung ausgeben und zu Punkt “1” zurück.

7) Wenn ja, Programm starten. Ob das wohl alles war? Nicht anzunehmen, denn da passiert noch mehr, von dem wir im Augenblick keine Ahnung haben. Und das alles passiert, wenn wir eben mal mit der Maus auf dem Pad langrutschen und die Taste zweimal drücken. Ist schon toll, oder? Was den meisten Amiga-Besitzern fehlt, ist das Wissen, wie man solche Befehlssequenzen, genannt Programme, erstellt. Programmiersprachen gibt es viele, angefangen beim mitgelieferten AmigaBASIC bis hin zu C oder Assembler. Doch welche Sprache ist die richtige, welche zu schwer für Einsteiger, welche bietet die meisten Möglichkeiten?

Als Einsteiger in den Zug nach Babylon fühlt man sich da oft verraten und verkauft. Fast jeder hat seine Lieblingssprache, keiner will von seiner Meinung ein winziges Stückchen abweichen und Andersdenkende werden gleich zum Schweigen gebracht. Was also tun, ratloser Ratsuchender?

Wir von der AMIGA DOS haben uns, wie wir meinen, was ganz besonderes ausgedacht. Wir sagen Ihnen nicht, welche Sprache besser ist, wir lassen es sagen. Und zwar von Profis, die mit ihrer Lieblingssprache Tag für Tag zurechtkommen müssen. Und so werden Sie auf den folgenden Seiten lesen, warum diese Sprache mehr kann als die anderen oder eine dritte einfacher zu erlernen ist. Was wir uns davon versprechen, ist klar: Sie bekommen geballte Informationen zu den Sprachen, ohne daß Sie verwirrt werden. Suchen Sie sich Ihren Favoriten unter den Computersprachen aus, lernen Sie die Vorzüge und Nachteile kennen, und Sie werden sehen, daß viele Ihrer Fragen sich beantworten lassen.

Um noch mal auf unsere Einleitung zurückzukommen: Bevor unsere Wunderkiste so einen Satz versteht, werden wohl noch ein paar Jährchen vergehen, aber bis dahin gibt es sowieso den wandelnden Amiga mit Kopf, Armen und Beinen und einem eingebauten Fach für Beruhigungspillen...

(jb)

Die Sprache Modula2, wie man sie heute kennt, ist eine von Prof. Dr. Nicolas Wirth, dem Schöpfer von Pascal, entwickelte Sprache. Modula, der Name steht für "MODular LAnguage", wurde zur Systemprogrammierung und zum Compilerbau konzipiert. Für den Amiga sind drei Compilersysteme erhältlich: TDI Modula II, Benchmark Modula2 und M2Amiga von A+L AG. In Deutschland ist der schweizerische M2Amiga am weitesten verbreitet, deshalb beziehe ich mich, wenn ich vom Compiler spreche, auf dieses System.

Modula2 ging aus Pascal und Modula, einer Sprache, die zur Programmierung von parallelen Prozessen entwickelt wurde, hervor. Modula2 übernimmt die Vorteile der strukturierten Programmierung von Pascal, beseitigt aber die Unzulänglichkeiten dieser alten Sprache. Zusätzlich bietet das Sprachsystem maschinennahe Sprachkonstrukte, wie sie von C bekannt sind.

Programme in Modula2 sind modular aufgebaut, das heißt, sie bestehen aus mehreren Modulen, die durch exakt definierte Schnittstellen in Verbindung treten. Dadurch ist es zum Beispiel sehr gut möglich, daß mehrere Programmierer unabhängig an einem Projekt arbeiten; es muß nur Übereinstimmung über die Schnittstellen (»DEFINITION MODULE«) bestehen. Da man alle Module getrennt testen kann, ist eine große Zuverlässigkeit garantiert. Der Aufruf jeder Funktion, ihr Rückgabewert und ihre Parameter sind genau definiert, dadurch ist ein unsichtlicher falscher Aufruf unmöglich.

Modula2 bietet variable Parameter, so daß eine String-Einlesefunktion, im Gegensatz zu C, ohne Zeiger beziehungsweise Adressen auskommt. Durch "opaque" (unsichtbare) Typen, offene Felder und die Einteilung in Module, die eine fast beliebige Wahl des Sichtbarkeitsbereichs eines Bezeichners ermöglicht, ist ein Programm in Modula2 sehr übersichtlich. Die Sprache zwingt den Programmierer, saubere Module zu schreiben. So ist im Gegensatz zu durchschnittlichen Programmen in C eine leichte Wartung möglich. Durch die strenge Typen-

Kai Bolay

Modular zum Erfolg

Wie in einem Baukasten-Prinzip setzen sich bei Modula alle Programmteile zu einem Ganzen zusammen.

kontrolle und den massiven Einsatz von Kontroll-Code, der natürlich auch abgeschaltet werden kann, ist Modula2 sehr sicher.

Durch den klaren Aufbau der Sprache, so sind zum Beispiel Zeiger an einen Typ gebunden, ist Modula2 sehr gut für Anfänger geeignet. Des weiteren kommt dem Anfänger entgegen, daß das System M2Amiga in deutscher Sprache erhältlich ist, was die Einarbeitung erleichtert. Außerdem gibt es für Profis und Anfänger eine sehr große Modul-Sammlung auf Public-Domain-Disketten, die AMOK-Reihe (AMOK = Amiga Modula & Oberon Klub). Sie enthält Module zu allen Themen (zum Beispiel IFF, Sound, Graphics, Intuition, DOS, TrackDisk, und so weiter) die natürlich zum Einbau in eigene Programme gedacht und sehr gut zu Lernzwecken zu gebrauchen sind. Auch Utilities und Spiele, natürlich mit Quelltexten, sind vorhanden.

Daß sich C zum Programmieren des Amiga besser als andere Sprachen eignet, weil das Amiga-Betriebssystem in C geschrieben wurde, stimmt nicht. M2Amiga bietet ein komplettes Interface zu den Libraries und Devices des Amigas. Libraries werden automatisch geöffnet, ihre Funktionen ohne Zwischencode aufgerufen. Das Compilersystem kann von Workbench und CLI gleichermaßen benutzt werden. Programme des M2Amiga laufen in der Regel auch von Workbench und CLI, ohne Zutun des Programmierers. Auch die BCPL-Pointer, werden vorbildlich unterstützt: Sie werden automatisch richtig behandelt. Ein sehr gutes Laufzeitsystem fängt bei M2Amiga fast alle

Abstürze ab und beendet ein Programm, bei Zurückgabe aller Ressourcen (Speicher, Windows, etc.), ordnungsgemäß.

Doch einige Nachteile von Modula2 sollte man nicht verschweigen: Leider ist Modula bis jetzt nicht standardisiert, was sich negativ auf die Portierbarkeit und damit auf die Verbreitung in der Industrie auswirkt. Aber ein Standard ist in Sicht! Auch sehen manche Leute es als einen Nachteil an, daß man in Modula2 keine "Wollmilchsau-Prozeduren" wie »printf()« zur Verfügung hat, da

die Sprache keine variable Anzahl von Parametern zuläßt. Gemischte Ausgabe von Zahlen und Strings muß man deshalb mit mehreren Prozeduraufrufen programmieren: Es ist etwas umständlicher, aber viel übersichtlicher, von der Implementation her sauberer und leichter lesbar.

Als Argument gegen Modula2 hört man öfters, die Programme seien groß und langsam. Doch dies liegt nicht in der Natur der Sprache, sondern am dazugehörigen Compiler, dessen Code sich noch stark optimieren läßt. Auch das große Laufzeitsystem tut das Seine. Ein weiterer Punkt ist der nicht optimierende Linker, es wird immer das ganze Modul eingebunden, auch wenn nur eine Prozedur importiert wird. Zwar gibt es jetzt das Tool M2Optimize«, das nachträglich (!) überflüssigen Code entfernt, doch dies ist keine befriedigende Lösung. Das Objektformat des M2Amiga ist auch ein Ärgernis, da es nicht kompatibel zu »ALink« oder »BLink« ist (nachträglich erreichbar mit »M2ObjectConverter«). Dadurch wird das Einbinden von Assembler unnötig

Der Autor

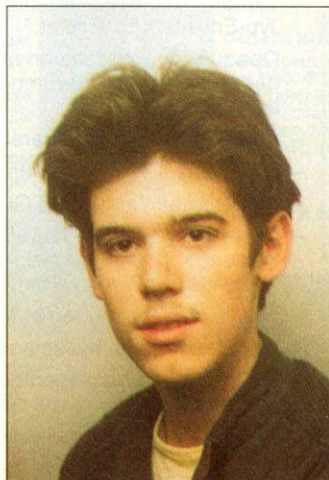
Name: Bolay

Vorname: Kai

Geboren: 24.4.73 in Stuttgart

Schulbesuch: vier Jahre Grundschule, bis jetzt sieben Jahre Gymnasium, ab September 12. Klasse Albert-Schweizer-Gymnasium Leonberg

Hobbies: Schach, Tennis, Lesen, Computer (!)



Computer: vor sieben Jahren C64 in BASIC, später Assembler, vor zwei Jahren: Amiga 2000 zuerst in Amiga BASIC, bald C, dann aus Überzeugung Modula2, jetzt auch Oberon (seit »Fridis«...), Voll-Mitglied im AMOK (Amiga Modula & Oberon Klub) Stuttgart
AMOK ist seit kurzem eingetragener Commodore-Entwickler!

Anlage: Amiga 2000, Highscreen KP 548 (Monitor), Supra Modem 2400, NEC Pinwriter P6, 2*3,5-Zoll-Diskettenlaufwerk, 5,25-Zoll-20MB-Harddisk, Speichererweiterung 2MB, Kickstart 1.3/1.2 auf Umschaltplatine, Kickstart 2.0 als File mit Hilfe von »KickIt«.

kompliziert (nur mit »INLINE«). Auch das Laufzeitsystem »Arts« hat seine Macken:

Es patcht die Funktion »OpenWindow()« und die Process-Struktur, wodurch Programme in M2Amiga zum Beispiel nicht im »WShell« laufen. Zwar kann der Compiler resident gehalten werden (nicht mit »ARP«), doch dies nicht ohne Fehler.

M2Amiga kann auch keine reentranten Programme erzeugen. Und im Angesicht von Kickstart 2.0: Der Compiler läßt sich nicht unter OS 2.0 starten, da er in der Version 3.3 resident libraries benutzt, die unter OS 2.0 nicht mehr existieren.

Wie man sieht, sind die meisten Nachteile, die man im all-

gemeinen Modula2 anlastet, auf den weit verbreiteten Compiler M2Amiga zurückzuführen. Die Probleme sind der A+L-AG bekannt und werden hoffentlich bald beseitigt. Wenn dies der Fall ist, ist Modula2 erst recht eine der besten Sprachen, mit der Sie Ihren Amiga programmieren können. Sie ist für Profis und

Anfänger, für den Amiga und andere Computer bestens geeignet.

Abschließend noch ein Zitat des Schöpfers von Modula2, Professor Dr. Wirth: »C ist mit Syntax verzuckerter Assembler. Und wenn man die kryptischen Notationen ansieht, wird es einem fast übel...«

(jb)

Oberon ist der direkte Nachfolger von Modula2 und übernimmt die bekannten Vorteile dieser Sprache, verzichtet aber auf wenig gebrauchte oder ineffiziente Sprachbestandteile, um den Compilerbau zu vereinfachen. Neu hinzugekommen ist die Hierarchie bei den numerischen Typen, die einige Komplikationen von Modula2 ausräumt, und die Möglichkeit der Record-Erweiterung. Mit ihr ist es zum Beispiel möglich, ein Listenverwaltungsmodul zu schreiben, das mit dem Typ »List« arbeitet. Wenn man jetzt eine Liste aufbauen will, erweitert man »List« einfach um die gewünschten Information, man kann aber trotzdem mit den Prozeduren aus dem Listenmodul weiterarbeiten. Für den Amiga gibt es nur ein Compilersystem für Oberon: Amiga-Oberon. Es ist im September bei A+L AG erschienen. Eine Demo-Version gibt es auf der PD-Diskette »AMOK #36«. Oberon bietet viele ungewöhnliche und nützliche Features: Um Betriebssystem-Strukturen vorzubelegen, bietet der Compiler strukturierte Konstanten an. Ein komplettes und komfortables Interface zum Betriebssystem ist selbstverständlich.

Dank des kleinen, aber trotzdem sehr guten, Laufzeitsystems, Registeroptimierung, Small-Code, -Data und optimierendem Linken sind Programme in dieser Sprache unglaublich kurz und schnell. Der Compiler erzeugt natürlich Object-Files, die zu »ALink«/»BLink« kompatibel sind und unterstützt Small-Code/-Data. Routinen aus anderen Sprachen (zum Beispiel C oder Assembler) kann man daher sehr leicht einbinden. Alle Program-

Kai Bolay

Besser geht's mit Oberon

Das Weiterentwicklungen durchaus ihren Platz im Sprachendschungel einnehmen können, zeigt die hier vorgestellte Sprache.

- Variante Records werden nicht mehr unterstützt und durch die Typ-Erweiterung ersetzt.
- Opaque Typen existieren nicht mehr als solche, es kann für jedes einzelne Record-Element bestimmt werden, ob es exportiert werden soll.
- Aufzählung und Unterbereichstypen wurden aufgegeben.
- Die Typen »CARDINAL« und »ADDRESS« gibt es nicht mehr.
- Arrays haben als untere Grenze jetzt immer 0.
- Importe sind nur noch qualifiziert möglich, erlauben aber Abkürzungen (zum Beispiel: »IMPORT I: Intuition« erlaubt zu schreiben: »I.WindowPtr«).
- Lokale Module wurden aufgegeben.
- Das Hauptmodul hat keinen Sonderstatus mehr.
- Definition und Implementation sind jetzt in einem Modul, exportierte Bezeichner werden mit »*« markiert.
- Der Befehl »WITH« hat eine andere Bedeutung, die ursprüngliche gibt es nicht mehr.
- Den Befehl »FOR« gibt es nicht mehr (durch »WHILE« zu ersetzen!).
- Coroutinen sind nicht mehr als Standardmodul zu verwenden.
- Typ-Erweiterung ist neu in Oberon: Records können erweitert werden.
- Der Typ »LONGREAL« enthält die Typen »REAL«, »LONGINT«, »INTEGER« und »SHORTINT«. Dies bedeutet in dem Fall, daß der Typ »LONGREAL« alle anderen Typen enthält; das heißt, man kann also einem »LONGREAL« einen »INTEGER« zuweisen; aber man kann nicht beispielsweise einem »INTEGER« einen »LONGINT« zuweisen, da dieser nicht in ihm enthalten ist.
- Neue assemblernahe Prozeduren »ASH«, »LSH« und »ROT«

Tabelle. Unterschiede zu Modula2

me in Oberon, die mit dem Small-Data-Modell compiliert wurden, können ohne Probleme resident gehalten werden, da sie reentrant sind! Übrigens ist das komplette Compilersystem mit SmallData compiliert worden und daher reentrant. Deshalb ist es auch kein Problem, Compiler, Editor, und so weiter resident zu halten.

Dem Compiler wird ein Library-Linker mitgeliefert, mit dem es ganz einfach möglich ist, Amiga-shared-Libraries zu programmieren! BCPL-Pointer sind kein Problem, sie werden automatisch konvertiert. Wie man sieht, wird das Betriebssystem in vorbildlicher Weise unterstützt. Mit Kickstart 2.0 gibt es natürlich keine Probleme.

Es ist möglich, direkt aus dem Editor Programme zu compilieren, zu linken und zu starten, das heißt, man muß den Editor zur Programmentwicklung nicht immer verlassen. Natürlich werden die Fehlermeldungen im Quelltext angezeigt. Das gesamte System arbeitet sehr schnell und sicher. Der in Modula2 erfahrene Programmierer hat keine Probleme, auf Oberon umzusteigen, aber auch der Anfänger hat es leicht: Neben dem klaren Aufbau der Sprache hilft ihm die Modulsammlung der AMOK-PD-Reihe.

Alles in allem ist Oberon ein würdiger Nachfolger für Modula2 und hat dank des Compilers »Amiga-Oberon« beste Chancen, eine große Fan-Gemeinde zu begeistern. Oberon ist dank seiner Vielseitigkeit und der von Modula her schon bekannten Vorzüge auf dem besten Wege, ein Spitzenreiter bei den Sprachen zu werden.

(jb)

AMIGA 90

Halle 10.2
Stand 1114



Golem SCSI II *State of the Art*

Markt und Technik Urteil "sehr gut" mit der höchsten Bewertung die je eine Festplatte erhalten hat
Die schnellste Festplatte für Ihren Amiga ● Datentransferrate bis 870 KB/sec. ● autom. Prozessor-Erkennung ● kein kritischer DMA Zugriff ● 16 Bit SCSI Controller ● durchgeführter SCSI Bus ● bis 8 SCSI Geräte anschließbar ● Quantum Qualitätslaufwerke ● als Filecard für A 2000 ● als externe Lösung für A 500/A 1000 ● mit eigenem Netzteil und Lüfter SCSI Interface im Rechnerdesign ● alle SCSI Festplatten incl. Golem Backup Software

Filecards A 2000		
Golem S40	40MB	1199.-DM
Golem S80	80MB	1599.-DM
Golem S105	105MB	1799.-DM
Golem S170	170MB	2499.-DM
SCSI II Controller		
extern A500/A1000		
Golem SE40	40MB	1399.-DM
Golem SE80	80MB	1799.-DM
Golem SE105	105MB	1999.-DM
Golem SE170	170MB	2899.-DM
SCSI II Controller im Gehäuse für A500/1000		599.-DM



GOLEM Eprommer

Der Leistungsfähige ● brennt 27512 (64KB) in 15 Sekunden ● intelligenter Algorithmus integrierter Monitor ● brennt alle gängigen Eproms bis zum Megabit ● Eprommer incl. komfortable Brennersoftware

249.-- DM



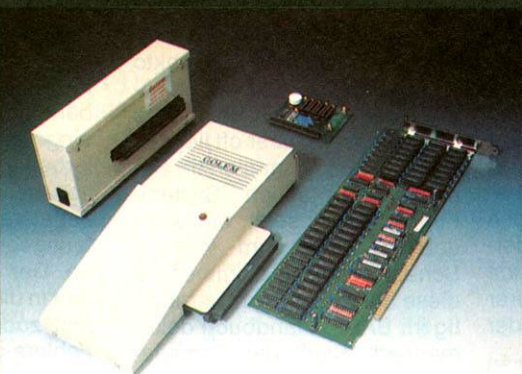
GOLEM TOWER

Der Amiga 2000 im Towergehäuse, räumt Ihren Schreibtisch auf und hat jede Menge Platz zur Aufnahme von Festplatten bis 1 Gigabyte und 3 Laufwerken und und und...

Das äußerst attraktive Gehäusedesign gibt Ihrem Amiga eine besondere Note.
Leergehäuse incl. Kabelsatz

548.-- DM

Auf Wunsch bauen wir Ihnen persönlich Golem Tower mit allen gewünschten Erweiterungskarten, Zusatzlaufwerken und Festplatten. Rufen Sie uns an!



GOLEM Ram Erweiterungen

Voll autokonfigurierende Speichererweiterungen für alle Amiga.

A) **externe Ram Box für Amiga 500/1000** 549.-DM
● im jeweiligen Rechnerdesign ● abschaltbar ● Busdurchführung ● erweiterbar bis 8 MB.

B) **8 Megabyte Steckkarte für Amiga 2000** 549.-DM
● 2MB bestückt, 8MB gesockelt ● echtes Fast-Ram nach GOLEM Standard.

C) **512KB Einsteckkarte für Amiga 500** 129.-DM
● mit gepufferter Uhr und Abschalte ● in stromsparender Megabit Technologie



GOLEM HD 3000A

Autobootende Festplatten für alle Amiga ● extern für A 1000 und A 500 ● Filecard für A 2000 ● autoboot ab Kick 1.3 ● automount aller Partitionen autoknfig auch unter Kick 1.2 ● stabiles Gehäuse m. Lüfter u. Netzteil ● Datentransfer bis 400K/sec. bei vollem Multitasking ● Qualitätslaufwerke ● Auto-Fehlererkennung bringt immer volle Plattenkapazität

20MB	948.-DM	30MB	1048.-DM
40MB	1448.-DM		
Filecard A 2000			
20MB	699.-DM	30MB	799.-DM
40MB	949.-DM		



GOLEM Sound II

189.-- DM

Ein Audio Digitizer für höchste Ansprüche ● DIN u. Chinch Anschlüsse ● Stereo u. Mono Sampling ● optisches Aussteuerungsdisplay ● kompatibel zu aller gängigen Software

GOLEM Sound Mashine

129.-DM

Professionelle Sampler-Software für Stereo- und Monobetrieb ● 100% Deutsch ● Echtzeitecho ● Sound-Editing

Sound Packet Digitizer und Software 269.-DM



GOLEM Drives

Amiga-farbenes Metallgehäuse ● Busdurchführung bis DF3: ● Ein/Aus Schalter PC Karten und Sidecar kompatibel ● Trackdisplay ● 5,25" mit 40/80 Track Schalter NEC Laufwerke ● Ein GOLEM Qualitätsprodukt

A) 3,5 Zoll Trackdisplay	239.-DM
B) 3,5 Zoll	199.-DM
C) 5,25 Zoll Trackdisplay	289.-DM
D) 5,25 Zoll	259.-DM
E) A 2000 intern incl. Einbausatz	179.-DM

Trackdisplay A 2000

einsteckbares Display für die Laufwerke DF0: u. DF1: zur Anzeige der aktuellen Kopf- und Stepperposition

89.-- DM

AKTUELL

AMIGA 90

Halle 10.2
Stand 1114

BASIC als sogenannte Interpretersprache ist geradezu geeignet, um erste Programmierschritte zu unternehmen. Zwar hat die aktuelle Version bereits etwas "Schimmel angesetzt", aber jeder Einsteiger kann sofort mit ersten Gehversuchen beginnen. Das deutsche Handbuch zu AmigaBASIC ist als Einstieg ideal. Der Nachteil: Commodore hat das Handbuch konzipiert. Dadurch sind alle Möglichkeiten, die uns in AmigaBASIC offenstehen (wie schon beim 64er), nur recht dünn beschrieben.

Durch Pull-down-Menüs werden Lade-, Speicher-, Editier-, Start-, Stop- und Debug-Funktionen aufgerufen oder alternativ durch die Tastatur. Einfacher und leicht verständlicher Aufbau der Befehle und Schreibweise ermöglichen dem Einsteiger schnelles Anfreunden mit AmigaBASIC. Wer sauber und strukturiert programmiert, der verwendet nicht die schnöden GOTOS oder GOSUBS wie sonst in Basic-Dialekten üblich, sondern versucht es mit While-Wend

Siegfried Rings

Eine Lanze für BASIC

"BASIC", werden viele die Nase rümpfen, "ist nur etwas für Anfänger." Dies ist wohl weit gefehlt.

oder If-Else. Die Subroutinen in AmigaBASIC ähneln den PROCEDURES von Pascal. Durch die TAB- und WIDTH-Funktion kann man leicht (wie in Fortran möglich) Tabellen erstellen. Zeilennummern benötigt man keine, was dem Überblick eines Programms erheblich zu Gute kommt. Der größte Vorteil von AmigaBASIC liegt im Preis. Es ist kostenlos auf jedem Amiga vorhanden. Gravierende Nachteile sind jedoch auch zu vermeiden. Ein dermaßen lahmer Editor hemmt den eifrigen Programmierer und läßt ihn auffluchen ob der

niedrigen Scroll-Geschwindigkeit des Cursors. Abhilfe schafft hier ein normaler Text-Editor.

Selbstverständlich ist AmigaBASIC Multitasking-fähig, es kann also neben anderen Programmen laufen. In puncto Geschwindigkeit ist AmigaBASIC mit seinem Befehlssatz recht langsam, aber oft liegt es auch am Programmierstil. Sobald man jedoch System-Funktionen einsetzt, wird AmigaBASIC fast zum Renner. Schade nur, das Commodore diese überhaupt nicht vernünftig im BASIC-Handbuch dokumentiert. Nach der Compilation der Programme kann man dann einen Geschwindigkeitsrausch auf dem Bildschirm erleben.

Empfehlenswert ist der Kauf eines Compilers. Zwei recht hochkompatible bieten sich an: Hisoft und Absoft. Zudem sieht man es einem guten Compiler nicht an, ob es in BASIC oder C geschrieben ist. AmigaBASIC hat auch seine Grenzen. Manchmal muß man schon kleine Sub-Routinen schreiben, um einen Befehl zu simulieren, den andere Programmiersprachen als Standard besitzen. Trotzdem finde ich, daß der Befehlssatz groß genug ist, um viele Probleme direkt und schnell zu lösen. Natürlich kann man die Hardware direkt durch PEEKs und POKEs beeinflussen. Was ich selbst vermisste, sind Befehle zur Bit-Manipulation/Abfrage wie bei Assembler. Nirgendwo anders als in BASIC kann man schnell eine Benutzeroberfläche programmieren und verändern. Auch Mausabfragen bezüglich der Position sind eine starke Sache. Screens und Windows werden in einem Befehl deklariert und müssen nicht erst durch lange Strukturen (in C) oder langen Befehls-

listen beschrieben werden. Dateiverwaltungen kann man unter 20 Zeilen schreiben; selbst Diskettenmanipulationen und Kopierprogramme lassen in BASIC verwirklichen. Die Stringverwaltung ist vorbildlich und kinderleicht zu handhaben. Man kann Bilderl Bildbereiche laden und speichern, Bildbereiche schnell scrollen (durch Benutzung des Blitters), eine Copperliste schreiben und verändern, Soundtracker- und digitalisierte Musik abspielen. Natürlich ist diese Art der Anwendung eigentlich Assembler-Freaks vorbehalten (wegen der Ablaufgeschwindigkeit), aber es ist auch in BASIC möglich, ruckelfrei gemalte Großbuchstaben scrollen zu lassen. Zeitkritische Routinen sollten natürlich in Assembler geschrieben und dann in AmigaBASIC eingebunden werden. Die Fehlerbehandlung ist abgesehen von der Geschwindigkeit des Fensteraufbaus und der Textausgabe eine gelungene Routine. Der Interpreter springt direkt in die fehlerhafte Zeile und gibt zudem noch die Art des Fehlers an. Programmiertechnisch können viele Fehler durch eine Routine »On Error gosub Marke« abgefangen werden. Weder in C oder Assembler finden Sie vergleichbare Fehler Routinen, es erscheint höchstens ein Ticket nach Indien (ich meine die niedlichen roten mit den lustigen Zahlen; Gruß von der Exec-Base).

Trotz der vielleicht hier aufgezeigten Mängel stellt AmigaBASIC für mich die beste Programmiersprache dar. Einfache Handhabung, leichtes Erlernen und verständliche Befehle sind ein gutes Argument. Mit dieser Sprache ist das Vorurteil der Spaghetti-Code-Programmierung endlich vom Tisch, denn strukturierte und übersichtliche Programme sind möglich. Für die Zukunft würde ich mir wünschen, daß Microsoft das Programm überarbeitet. Beschleunigung der Ablaufgeschwindigkeit und des Fensteraufbaus sind hier die Stichworte. Oder Microsoft sollte AmigaBASIC inklusive Source-Code als Public-Domain freigeben – die eingefleischten Cracks werden's schon richten.

(vb)



Der Autor

Name: Siegfried Rings

Geboren: Jahrgang 1966

Werdegang: 1966 dachten meine Eltern sicherlich nicht an BASIC, als ich zur Welt kam. Nach Beendigung der Realschule und ersten Gehversuchen mit Taschenrechnern erlernte ich den Beruf des Elektro-Installateurs. Mittlerweile studiere ich Elektrotechnik und programmiere nebenher für Firmen Anwendersoftware sowie Intros für Public-Domain-Disketten.

EDV-Erfahrung: Fasziniert von Strom, Spannung, Null

und Eins entschied ich mich 1984 entgegen dem Rat von Bekannten zum Kauf eines C 64 und nicht zu einem etwas besseren Automobil. Nach kurzer Spielzeit entdeckte ich die Programmierleidenschaft in mir und begann die ersten Programme im spärlichen C64-Basic zu erstellen. Als während der Bundeswehrzeit ein Kamerad mich nächtelang mit seinem damals noch über 3000.-DM teuren Amiga 1000 faszinierte, beschloß ich im März 1987, kurz vor dem Preispurzel bei Commodore, den 64er gewinnbringend zu verschleudern und einen Amiga 1000 zu kaufen. Nach Durchwühlen des CLI-Handbuches fing ich an, strukturiert in AmigaBASIC zu programmieren. Kurze Abstecher in C und heute noch in Assembler ermöglichten mir, etwas mehr Einblick in die Programmier-techniken und Hardware des Amigas zu nehmen. Von Raubkopien halte ich soviel wie von teuren Autos, denn man nutzt sie fast nie und kommt auch nicht viel schneller ans Ziel. Ich arbeite mit und empfehle jedem Public-Domain statt Raubkopien.

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-Spietips 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 Spietips – gleich welcher Art – verfügen, würden wir uns freuen, von Ihnen Post zu bekommen.

Und hier nun die Adresse, an die Sie Ihre Spietips schicken können:

**DMV-Verlag
Redaktion AMIGA DOS
Stichwort: Spietips
Postfach 250
3440 Eschwege**

PS: Unter allen Einsendungen verlosen wir jeden Monat fünf Softwaretitel für die besten Spietips.

Der Speicherprofi

FSE

Floppylaufwerke
Festplatten
Rams

Speichererweiterungen

A500, 512KB, Uhr 128.-
A500, 512KB, max 2MB 398.-
A2000, 2MB, max 8MB 598.-
dto. 4 MB bestückt 898.-

TEAC Diskettenlaufwerke

Vollkompatibel, anschlußfertig, abschaltbar, Busdurchführung, bei 5.25" Stationen 40/80 Tr. schaltbar, 12 Monate Garantie
3.5": 179.- 5.25": 198.-
NEC 1036, intern für A2000: 129.-



SyQuest 44 MB Wechselplatte

Kompl. für A 2000, 2 J. Garantie, 20 ms, SCSI, mit Medium 1678.-
Medium SQ 400, 44 MB 198.-

66 MB 1148.-

28 ms, NEC, AutoBoot, AutoPark, 450 KB/S, OMTI, dg. Bus, extern, für A500/1000, 1 Jahr Garantie

**AMIGA
Köln 90**

*Wir stellen aus
09.-11. November 1990
Halle 10, Stand 623-724*

Frank Strauß Elektronik
Schmiedstraße 11
6750 Kaiserslautern
0631/67096-99 Fax 60697
Händleranfragen erwünscht



Quantum

Quantum Festplatten

sehr leise, zuverlässig, schnell
Interleave 1:1, 2 Jahre Garantie
64 KB Cache, 19 ms, 750 KB/S

16 Bit Harddisk für A500/1000

AutoBoot, 16 Bit Technik, kurze Bootzeit, 19 ms, 750 KB/S, leise, beachten Sie bitte u.a. Testauszüge, 2 Jahre Garantie, Quantum
42 MB 1248.- 84 MB 1648.-

Festplatten SCSI, A2000

AutoBoot, SCSI, 17 ms, 850 KB/S, 2 Jahre Garantie, Quantum, BOIL3
52 MB 1148.- 105 MB 1648.-
210 MB Rodime, 18ms 2548.-

Kickstart 9/90

Den Entwicklern gebührt wirklich ein sehr großes Lob, einfacher und verständlicher geht es kaum noch . . . Die FSE - Platte muß man in Sachen Leistung ganz oben ansiedeln.
Prädikat : Sehr Gut

AMIGA DOS 9/90

Die AT-Bus Festplatte von FSE stellt ein gut durchdachtes Produkt, das sich technologisch auf dem neuesten Stand befindet. Gute Konzeption, solide Verarbeitung und durchdachte Software sind hier zu einem fairen Preis erhältlich.

AMIGA 8/90

Festplatte und Lüfter arbeiten geräuscharm . . . Die Verarbeitung des Festplattengehäuses und die Montage von Netzteil, Lüfter und Festplatte sind einwandfrei . . . zählt die CHA-40Q5 zur Gruppe der schnellsten Festplatten für den A500.

Als Pluspunkt für diese Sprache wird häufig das Argument "Portabilität" gebracht, was mir in der Regel ein müdes Lächeln entlockt – die Amiga-C-Compiler sind sogar untereinander inkompatibel. Wie sieht es da erst mit der Portierung von Programmen anderer Rechner aus. Auch wenn ein Compiler das Programm selbst fehlerfrei übersetzt – der Gedanke, ein MS-DOS-Programm im status quo auf dem Amiga zu sehen, dürfte für die meisten User ein wahrer Alptraum sein. Denn die Faszination »Amiga« wird gerade durch seine vielfältigen Ressourcen wie die Intuition-Oberfläche geweckt. Und wenn ich die benutze (was nur zu empfehlen ist), dann läuft das Programm nur auf dem Amiga, ob es nun in C, Assembler, Modula oder Keilschrift geschrieben ist.

Nein, vom Amiga-Standpunkt aus gesehen, ist C erst einmal nicht portabel. Aber warum benutze ich dann C? Der Erfinder der meistgehaßten Programmiersprachen der Welt, Niklas Wirth, sagte einmal: "C ist ein verzuckerter Assembler". Recht hat er, und das ist genau der Grund, der für C spricht: die ultimative Leistungsfähigkeit von reinem Assembler gepaart mit den Möglichkeiten einer Hochsprache. C ist sehr maschinennah, ein guter Programmierer hat beim Schreiben von C-Code immer in Gedanken den vom Compiler erzeugten Assembler-Code vor Augen. Ohne mindestens grundlegende Kenntnisse von Assembler ist das

Oliver Wagner

C – eine Sprache nur für Könner?

Warum benutze ich C? Nun, warum sollte ich es nicht benutzen?

Schreiben von effektivem C-Code kaum möglich.

Optimaler Code

Moderne C-Compiler (auf dem Amiga ist Lattice-C in dieser Hinsicht immer noch ungeschlagen) erzeugen so gut optimierten Code, der einem unoptimierten, von Mensch geschriebenen Assemblercode problemlos überlegen ist. Ein Bekannter von mir veröffentlichte einmal ein kurzes Assemblerprogrammchen zum Abschalten des Tiefpassfilters mit korrektem Workbench-Startup in nur noch 92 Bytes und wettete eine Tüte Gummibärchen, daß es sich um die kürzest mögliche Version handeln würde. Nun, die betreffende Tüte hat ihren Besitzer bereits des öfteren gewechselt. Das Rennen machte zwar wiederum ein Assembler-Programm, allerdings lag ein Ver-

such in Lattice mit 88 Bytes schon besser als das Original. Wüßten Sie – so aus dem Kopf, daß die 6-Byte-lange Sequenz

```
MOVE.L # $20000, D2
```

durch das nur 4-Byte-lange

```
MOVEQ # $2, D2
SWAP D2
```

ersetzbar ist? – Wahrscheinlich nicht. Aber der Codegenerator des Compilers weiß es und kann solche Tricks (von denen es gerade beim 680x0-Prozessor Unmengen gibt) immer, wenn es möglich ist, anbringen.

Falls es wirklich nicht anders geht, kann ich immer noch problemlos Assembler mit C-Code verbinden: Die Schnittstellen sind bekannt und dokumentiert. Ich weiß, wo auf dem Stack oder in welchem Register ich welchen Parameter zu erwarten habe.

Und noch was: Gibt es einen neuen Programmiertrick, einen neuen Prozessor, einen neuen Coprozessor: Man muß ihn nur einmal dem Compiler beibringen und schon sind alle damit erzeugten Programme durch einfaches Neukompilieren ohne jeden Aufwand immer "state of the art", während der Assemblerprogrammierer sich wieder stundenlang durch seine Byte-Wüsten quält. Praktisches Beispiel: Mit Lattice-C ist es problemlos nur durch das Setzen eines Flags möglich, Code optimiert für 68000, für 68010, für 68020, für 68030, für 68881 zu erzeugen.

Das ist die eine Seite von C. Die andere zeigt sich in allen Vorteilen, die eine Hochsprache bietet: Funktionsaufrufe, komplexe Datentypen, Schlei-

fen, Bedingungen, Fließkommaarithmetik...

Dadurch wird das Programmieren wunderbar leicht: Für das, was ich in eine Zeile C-Code erreichen kann

```
if (user->iq==20 || user->sex==1) goto xit;
```

braucht der typische Assemblerprogrammierer das siebenfache:

```
move.l user,a0
moveq #20,d0
cmp.l d0,4(a0)
beq.s xit
moveq #1,d0
cmp.l d0,8(a0)
beq.s xit
```

Das Interessante daran: Im Vergleich schaffen Assembler und C-Programmierer bei der Lösung des gleichen Problems innerhalb eines Zeitraums den gleichen Umfang von Quelltext. Nur hat der C-Programmierer bei gleicher Zeilenzahl im Endeffekt mehr erreicht.

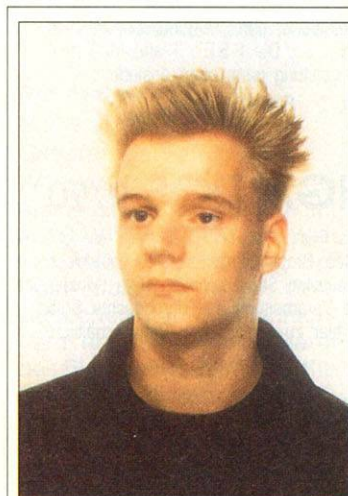
Und was ist später leichter zu merken, zu durchschauen, zu debuggen?

Natürlich war das Beispiel recht einfach – wie sieht es aus mit Fließkommaarithmetik in Assembler? Oder komplexe Strukturreferenzen? Man stelle sich das folgende im Assembler-äquivalent vor:

```
IntuitionBase->ActiveScreen
->RastPort.BitMap->Planes[0]
```

Denn eins ist zu bedenken: Das Amiga-Betriebssystem basiert auf solchen Strukturen und deren Verkettung. Denn es ist selber in C geschrieben, mit C hat man die besten Karten im Umgang mit der Kickstart. Auch die Dokumentation des Betriebssystems ist für C ausgelegt, Programmierer anderer Sprachen müssen sich mit Informationen aus zweiter Hand begnügen – meiner Erfahrung nach meist fehlerhaften. C ist und bleibt die Sprache für Leute, die ernsthafte Projekte auf dem Amiga und für den Amiga verwirklichen wollen. Die Codegenerierung heutiger Compiler läßt im Vergleich zu Assembler keine Wünsche mehr offen – in Umfang und Geschwindigkeit. Der Komfort, die Entwicklungsgeschwindigkeit eines Programms ist um ein Vielfaches höher und die Möglichkeiten zur Fehlersuche um ein Vielfaches besser. Warum sollte man es sich also schwer machen, wenn es auch einfach geht?

(vb)



Der Autor

Name: Oliver Wagner

Geboren: auch, am 21.1.1971

Werdegang: Ich arbeite seit 1983 mit Computern und bin seit 1986 stolzer Besitzer eines Amiga. Momentan erwartet mich ein Informatikstudium. Nebenbei betreibe ich eine kleine Firma und arbeite ab und zu als freier Programmierer und Autor.

EDV-Erfahrung: Während ich ZX-Spectrum, C-64 und Atari ST hauptsächlich in Assembler programmierte, entdeckte ich auf dem Amiga mein Herz für C. Projekte wie der Synthesizer-Sound-Editor »CZED« oder der Ansi-Editor »TextPaint« entstanden vollständig in dieser Sprache.

Drucker
Citizen Swift 24 **768,-**
4-Farb-Option dazu **168,-**

RAM-Karte 1-2-4-6-8 MB
für A2000 mit **588,-**
2 MB bestückt

Floppy-Drive 3,5 Zoll
1,44 MB/720 kB **158,-**
für AT-Karte

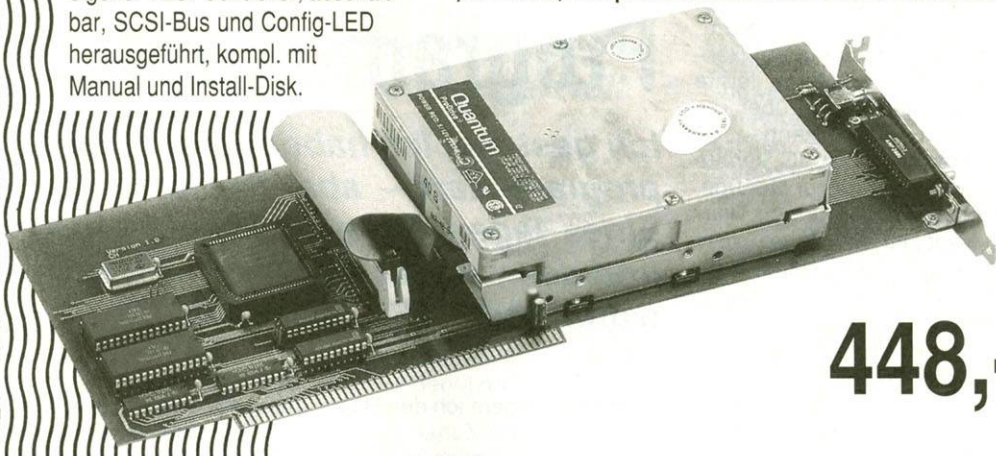
Evolution-Controller Filecard
Zum Einstecken in A 2000,
Autoboot unter KS 1.2/ 1.3/ 2.0,
Kooperation mit Streamer-Soft-
ware durch SCSI-Direkt-Einsprung,
direkte 16-bit-Übertragung ohne
DMA für Datenübertragungsraten
bis weit über 1 MB/sek., zukunfts-
kompatibel Amiga-OS V 2.0,
eigener VLSI-Controller, abschalt-
bar, SCSI-Bus und Config-LED
herausgeführt, kompl. mit
Manual und Install-Disk.

SCSI-Festplatten 3,5 "

mit Evolution-
Controller*

Fujitsu M 2614 S, 180 MB	2098,-	2498,-
Fujitsu M 2613 S, 135 MB	1898,-	2298,-
Quantum Prodrive Q 80 S, 80 MB	1498,-	1898,-
Seagate ST 1096 N, 28 ms, 80 MB	948,-	1348,-
Seagate ST 157 N-1, 28 ms, 48 MB	798,-	1198,-
Seagate ST 138 N-1, 28 ms, 32 MB	698,-	1098,-

***) montiert, komplett formatiert und sofort einsatzbereit**



448,-

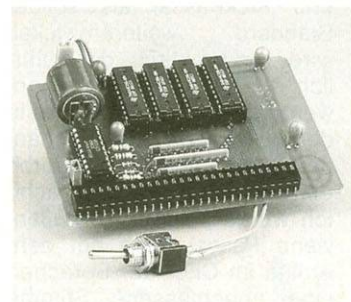
MASOBOSHI

Informationssysteme GmbH
Joachimstr. 16, 4630 Bochum
Telefon 0234/30 81 51
Telefax 0234/30 86 35

Speichererweiterung
für Amiga 500 intern, 512 KB
RAM, 4 Megabit-Chips, abschalt-
bar, inkl. akkugepufferter Uhr und
Ein-/Ausschalter mit Kabel, auto-
konfigurierend, läuft unter jedem
Kickstart und jedem Agnus-Chip,
alle RAM-Chips gesockelt.

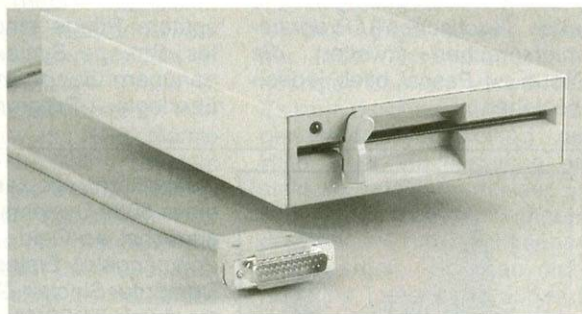
Neu: Serienmäßig mit schalt-
barem Schreibschutz für die Uhr,
kein Virus und keine abstürzenden
Programme können die Uhr mehr
löschen!

99,-



Floppy Drive 3,5 Zoll intern
Commodore A 500/A 2000
Original-Laufwerk, df 0-df 1
schaltbar, Front amiga-farben,
inkl. Staubschutzklappe und
Einbaumaterial.

148,-



Floppy Drive 3,5 Zoll extern
Chinon Markenfloppy, abschaltbar,
Bus durchgeführt, Front und Metallge-
häuse amiga-farben, automatisches
Diskchangesignal, superslimline
(25 mm hoch), 880 KB Speicher-
kapazität, komplett
anschlußfertig
mit Kabel

169,-



Floppy Drive 5,25 Zoll extern
Chinon Markenfloppy, abschalt-
bar, voll kompatibel zu MS-DOS,
Front und Metallgehäuse amiga-
farben, automatisches Disk-
changesignal, Bus durchgeführt,
880 KB Speicherkapazität, voll
kompatibel zu 3,5 Zoll Floppies,
komplett anschlußfertig mit Kabel

199,-

Bootselector
für DF 1 oder DF 2, 15,95.

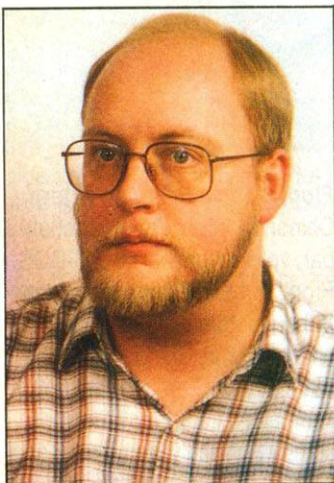
Bestellannahme und Ladenverkauf Mo-Fr 9.00-17.00
Auf alle Produkte 6 Monate Garantie

Es gelten unsere allgemeinen Geschäftsbedingungen. Händleranfragen willkommen.

Eigentlich war der erste Kontakt mit Pascal Mitte der 80er Jahre eher ein Zufall – mein erstes „kommerzielles“ Programm, ein Zeichensatz-Designer für den Joyce, war eine Koproduktion. Darin hatte ich den maschinennahen (Z80) Assembler-Teil übernommen, während ein Bekannter das „Drumherum“ in Pascal beisteuerte. Im Rahmen dieses Projektes lernte ich gewissermaßen „über die Schulter“ Pascal kennen – und schätzen. BASIC kannte ich damals schon zur Genüge, die ersten Erfahrungen wurden Ende der 70er Jahre in der Schule auf einem Wang-Computer gemacht, der satte 8 KByte RAM besaß, seine Daten auf Musik-Kassette aufzeichnete und mit allen seinen Komponenten einen größeren Schreibtisch komplett belegte.

Warum Pascal?

Im Laufe der Zeit wurde die EDV-Erfahrung durch Benutzung der verschiedensten Rechner und durch Programme in fast allen gängigen (und auch „exotischen“) Programmiersprachen erweitert, die Treue zu Pascal blieb jedoch bestehen...



Der Autor

Name: Michael Anton

Geboren: ja, am 13.12.1962

Beruflicher Werdegang:

Abitur, Studium der Germanistik und Anglistik zu Stuttgart, ständiger freier Mitarbeiter beim DMV-Verlag, derzeit tätig als Zivildienstleistender.

EDV-Erfahrung: Ende der 70er Jahre ersten Kontakt auf

Michael Anton

Bekenntnisse eines Pascal-Programmierers

Ich gestehe, ich habe früher in Basic programmiert – aber das ist lange her. Eines Tages lernte ich Pascal kennen, und das war der Beginn einer langen, wunderbaren Freundschaft...

Was macht eigentlich den Reiz von Pascal aus? Zunächst einmal seine Konzeption als Lehrsprache. Übersichtliche Blockkonstruktionen und sinnvolle Bezeichner (im Kontrast seien die logischen Operatoren in C genannt) erleichtern nicht nur den Einstieg in die Programmierung, sondern auch die spätere Pflege eines Projektes. Strenge Syntax- und Typenüberprüfung zwingt zu überlegter Programmierung,

mal kurz eine Variable oder einen Sprung einzuschieben, nur um zu sehen, ob das Programm dann läuft, ist nicht immer drin. Es mag zwar nicht jedermanns Sache sein, sich von seinem Compiler bevormunden zu lassen, aber der strenge Formalismus von Pascal hat auch seine guten Seiten: Ein Großteil der potentiellen Laufzeitfehler wird schon sehr früh erkannt, so daß Pascal-Programme etwas betriebssicherer sind.

Sicherlich spielt auch die Ästhetik eine kleine Rolle: Ein ordentlich strukturiertes Pascal-Programm „sieht einfach gut aus“, auch fügen sich die eigenen Routinen „nahtlos“ in die Sprache ein. Aber das ist natürlich nicht alles. Pascal ist eigentlich ein recht guter Kompromiß zwischen Erlernbarkeit (wie BASIC), Lesbarkeit (im Gegensatz zu C) und Strukturiertheit (im Gegensatz zu älteren BASIC-Varianten). Vergleichbares bietet nur noch Modula, dort fällt jedoch zusätzliche Tipparbeit an, die auch nicht jedermanns Geschmack ist.

Pascal auf dem Amiga

Der Sprachumfang des Standard-Pascal ist sicher nicht gerade berauschend, aber die meisten Implementationen haben diesen Standard erheblich erweitert und bieten auch die Ausnutzung systemspezifischer Möglichkeiten. Mein Motto daher: Pascal, wo immer

es verfügbar ist... Als eingefleischter „Pascalero“ war ich natürlich enttäuscht, daß zur Zeit meines Umstiegs noch kein brauchbares Pascal für den Amiga verfügbar war, von PCQ-Pascal aus der Public Domain einmal abgesehen, welches zumindest als Notlösung dienlich war. Nachdem die Firma Maxon mit KickPascal nun den Sprung ins kalte Wasser gewagt hat, sieht die Sache schon besser aus.

Weitgehende Kompatibilität zu Standard- und Turbo-Pascal, eine integrierte Oberfläche, die dauernden Wechsel von Editor zu Compiler zu Linker überflüssig macht; dazu ein schneller und effektiver Compiler – das ist KickPascal. Fast schon der Stoff, aus dem die (Pascal)-Programmiererträume sind...

In seiner jetzigen Form ist Pascal auf dem Amiga noch nicht ganz perfekt. KickPascal mangelt es noch an einigen Verbesserungen wie etwa dem modularen Unit-Konzept der neueren Versionen von Turbo Pascal, der Einbindung von anderen Programmiersprachen oder der etwas besseren Ausnutzung der Ressourcen des Amiga. Im Vergleich zu Turbo Pascal unter CP/M und MS-DOS entspricht der derzeitige Stand von KickPascal etwa der Version 3.0 von Turbo Pascal. Wenn man sich jedoch vor Augen hält, welchen Erfolg diese Version hatte und wie sie sich bis zu Turbo Pascal 5.5 unter MS-DOS weiterentwickelt hat, so stehen die Karten für Pascal auf dem Amiga gar nicht so schlecht.

Der Erfolg ist jedoch nur dann garantiert, wenn sich genügend Interessenten für diese Sprache zusammenfinden und KickPascal als solider Standard weiterentwickelt wird. Ein Umstand, der natürlich nur dann eintreten kann, wenn ein gewisses Interesse in der Anwenderschaft vorhanden ist, die eine Pflege durch den Hersteller möglich macht. Ich würde es sehr begrüßen, wenn Pascal auch auf dem Amiga im Chor der Sprachen eine angemessene Stimme zufiele – und dieses durch einen entsprechenden Kurs in der AMIGA DOS unterstützen. Wie stehen die Chancen dafür? – Lassen Sie es uns wissen...

(jb)

Jede Sprache nimmt für sich in Anspruch, die beste zu sein. Somit bleibt dem Programmierer nur die Möglichkeit, sich zu informieren. Auch ich habe mich vor einiger Zeit für eine Programmiersprache entscheiden müssen, die für Themenbereich, Computer und mich den größten Nutzen bringt. Es hat sich in der Praxis erwiesen, daß für mich in Bezug auf die Systemprogrammierung mit dem Amiga nur Assembler die richtige Sprache sein kann. Sie ist schnell, kurz und günstig auf Systemkomponenten bis hin zu den einzelnen Bausteinen verwertbar.

Viele kritisieren Assembler, weil die "mnemonische" Schreibweise unter Umständen unübersichtlich sein kann und sich dadurch eventuell extreme Fehler in ein Programm einschleichen. Dieses Manko war mir bekannt, daher habe ich mir einen sehr disziplinierten Programmierstil angeeignet. So bin ich in der Lage, meine Programme auch nach Jahren noch zu bearbeiten. Man darf in keinem Fall nur die Sprache allein betrachten. Schreibt man ein Tabellenkalkulationsprogramm, so ist sicherlich beim Amiga eine Kombination von C und Assembler zu empfehlen. Globale Berechnungen können somit recht einfach in C durchgeführt und schnelle Programmteile in Assembler programmiert werden. Um das Programmieren zu erlernen, ist Assembler allerdings nicht sehr gut geeignet. In einem solchen Fall würde man sich sicherlich Pascal oder Modula zuwenden. Wenn man aller-

Bert Schwarm

Inside Amiga

Warum Assembler? Wenn man sich eine solche Frage stellt, sollte man mehrere Blickpunkte in Bezug auf die Sprachverwirrung der Computerwelt berücksichtigen.

ding das Programmieren erlernt hat, führt in vielen Fällen an Assembler kein Weg vorbei. In Bezug auf den Amiga sei erwähnt, daß das Betriebssystem vom Grundsatz her in allen systemnahen Programmiersprachen sehr gut gehandhabt werden kann. Man sollte sich nur an die Konventionen des Herstellers halten. An dieser Stelle sei noch ein Gerücht erwähnt, das die Programmierwelt erschüttert hat.

Es soll tatsächlich möglich sein, in C schnelleren und kompakteren Operationscode (Verarbeitungscode) zu erstellen als in Assembler. Dem kann ich getrost widersprechen. Ein Compiler arbeitet meist nach einer Methode, die den Stack (Stapelspeicher) während des Programmlaufs sehr stark beansprucht. Mit einem in Assembler geschriebenen Programm brauche ich den Stack (Stapelspeicher) nicht unbedingt zu benutzen. So kann man sich leicht vorstellen, daß das C-Programm mehr Taktzyklen (Verarbeitungseinheiten des

Prozessors) verbraucht als ein Assemblerprogramm, das genau die gleiche Aufgabe übernimmt.

Es gibt noch einen Grund, der für die Benutzung von Assembler spricht. Das wäre beim Amiga das Multitasking, welches es erlaubt, mehrere Programme beziehungsweise Arbeitsabläufe des Prozessors zu spalten und somit den Eindruck entstehen zu lassen, zwei oder mehrere Programme arbeiteten gleichzeitig. In Wahrheit wird der Prozessor in bestimmten Zeitabständen unterbrochen. Dann arbeitet er bis zur nächsten Unterbrechung an einem anderen Programmteil weiter. Das Ganze wird durch das Betriebssystem (»EXEC«) überwacht. »EXEC« ist natürlich selbst ein Programm und benötigt ebenfalls den Prozessor. Das Resultat ist ein sich selbst verwaltendes Programmverwaltungssystem. Jetzt wird deutlich, warum der Amiga bei manchen Programmen in der normalen Rechnerkonfiguration langsam wirkt. Der Prozessor arbeitet nach wie vor mit der gleichen Taktfrequenz, er wird allerdings ganz gebraucht. Schreibt man seine Programme in Assembler, arbeiten diese bei entsprechender Programmierung mit einem Minimum von Befehlen. Die CPU kann somit verbleibende Taktzyklen für andere Programme verwenden. Wichtig ist hier die konsequente Nutzung dieser Sprache.

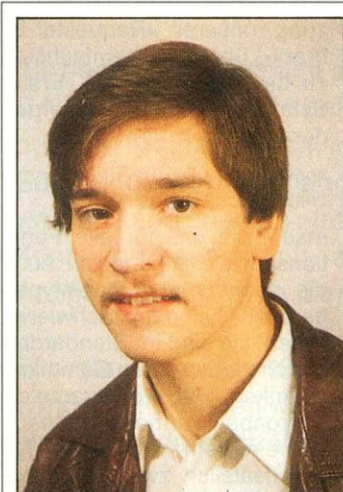
Wenn ich ein Programm für den Amiga schreibe, beachte ich in erster Linie das gesamte Umfeld für den Benutzer. Ein Beispiel: Es soll ein File-Requester geschrieben werden. Dieser soll bei der Erstellung des jeweiligen Verzeichnisses

möglichst schnell Dateien und ähnliches anzeigen. Man muß also das entsprechende Speichermedium in sehr kurzer Zeit nach den Verzeichnisinformationen durchsuchen. Zudem ist eine betriebssystemnahe Programmierung unumgänglich. Weiterhin ist das AmigaDOS in BCPL erstellt worden. Der Compiler dieser Sprache hat die Eigenschaft, langwortorientiert zu arbeiten und Adressen in ein für sich typisches Format zu wandeln (Länge 4 Byte oder 2 Word). Folglich ist das eigene Programm gegebenenfalls auf diese Umgebung des AmigaDOS anzupassen. Infolgedessen ist eine Sprache zu wählen, die nicht nur schnell ist, sondern auch effektive Systemanpassung erlaubt. Programmtechnisch ist eine Lösung in verschiedenen Sprachen denkbar. Die Effizienz von Assembler ist in diesem Fall wohl am besten.

Die vorhergehenden Beispiele bezogen sich nur auf die Nutzung des Rechners als Arbeitsmittel. Betrachtet man die vielfältigen Grafikmöglichkeiten des Amiga, so sind diese nur unter Assembler bis in höchste Ebenen verwertbar. Es kann durchaus auch ein sogenanntes Softscrolling mit Hilfe einer Hochsprache erstellt werden, dann müssen allerdings diverse "Zuckungen" der Laufschrift in Kauf genommen werden. Will man nicht nur eine Laufschrift, sondern parallel dazu noch Musik und Farbvielfalt, so ist das natürlich unter den obengenannten Bedingungen nur mit Assembler ohne merkliche Störungen des Programmlaufs zu bewältigen.

Von Natur aus ist, wie oben erwähnt, eine Hochsprache "langsamer" als Assembler und kann wahrscheinlich nur in Großrechenanlagen ohne Einbußen verwendet werden. Damit wird deutlich, daß Assembler zumindest im Kleincomputerbereich nach wie vor genutzt werden sollte, um Programme zu schreiben, die für den Benutzer eine größtmögliche Verwertbarkeit seines Computers beinhalten. In unserer Firma wird prinzipiell in Assembler programmiert, und ich bin davon überzeugt, mit diesem Konzept auch weiterhin Erfolg zu haben.

(jb)



Der Autor

Name: Schwarm
Vorname: Bert

Geboren: 19.04.1963 in Oberhausen

Kurzer Lebenslauf: Lehre als Mechaniker, schließlich Fachhochschulreife und anschließendes Studium in Bergbautechnik, Wechsel in den Informatikbereich, Autodidakt in puncto Programmiersprachen. Jetzt Geschäftsführer der Firma D.I.T., die Produkte wie »Virusdetektor«, »Vaccinator« und »Golden Tools« herausbringt. Kommende Projekte werden Hardware-Erweiterungen für die Amigas sein.

Franz-Josef Reichert

Vielseitige Requester für alle Fälle

Auf zum Endspurt! In den letzten drei Teilen haben wir Ihnen Schritt für Schritt gezeigt, wie man sich eine Amiga-spezifische »shared-library« zusammenbaut.

Heute wollen wir anhand praktischer Anwendungsbeispiele verdeutlichen, wie unsere Library-Routinen aus jeder Programmiersprache aufgerufen werden können. Dabei wollen wir auf die am meisten verbreiteten Sprachen eingehen, die auf dem Amiga Anwendung finden: AmigaBASIC, C und Assembler.

BASIC für alle

Beginnen wir bei der Sprache, über die fast alle den Einstieg in die Amiga-Programmierung gefunden haben: AmigaBASIC! BASIC hat daher im Hobby- und Heimanwenderbereich eine sehr weite Verbreitung erfahren. Die Argumentation, BASIC-Programme seien immer sehr viel langsamer als solche in anderen Hochsprachen, trifft auch nur bedingt zu: Wer alle Tricks kennt und sicher beherrscht, wird auch in BASIC Programme entwickeln können, bei denen in Bezug auf Verarbeitungsgeschwindigkeit und Bedienungskomfort kaum Abstriche gemacht werden müssen. Einer dieser Tricks ist die Benutzung von Maschinensprachebibliotheken, wie unsere »Requester-Library« ja eine ist. Die genauen Aufrufkonventionen wollen wir im folgenden Abschnitt besprechen.

Zunächst muß dem Interpreter mitgeteilt werden, welche Library überhaupt benutzt werden soll. Dazu dient die Anweisung »LIBRARY« im AmigaBASIC-Dialekt:

```
LIBRARY "requester.library"
```

bewirkt also, daß unsere »Requester-Library« geladen wird. Damit ist aber noch nicht vereinbart, an welcher Stelle im Speicher die Funktionen zu

finden sind, wie sie heißen und mit welchen Parametern sie aufgerufen werden müssen. Der Interpreter benötigt dazu Dateien mit der Endung »bmap«, die wir zunächst für unsere Library erzeugen müssen. Grundlage dazu ist eine Datei mit der Endung »fd« (steht für "Function-Definition"); die Datei »requester.lib.fd« wurde bereits im letzten Teil als Listing 4 abgedruckt. Weitere Dateien mit der Endung »fd« für die System-Libraries finden sich ebenfalls auf der Diskette »Extras«, im Verzeichnis »FD1.3«. Glücklicherweise brauchen wir uns um das genaue Format keine großen Gedanken zu machen. Die Umwandlung erledigt für uns ein BASIC-Programm mit Namen »ConvertFD«, das sich im Verzeichnis »BasicDemos« befindet. Starten Sie also das Programm, und folgen Sie den Anweisungen. Damit die »bmap«-Datei vom Interpreter gelesen werden kann, muß Sie sich im aktuellen Directory oder dem Verzeichnis »LIBS:« der Startdiskette (»Workbench«) befinden. Sollten Sie öfters mit wechselnden Verzeichnissen arbeiten, ist sie am besten in letzterem aufgehoben. Kopieren Sie bitte gleich noch die Dateien »dos.bmap« und »exec.bmap« dazu, sie werden auch zur korrekten Funktion unseres Beispielprogramms benötigt (siehe [1], Kapitel 6-12 und Anhang F).

Das Datenformat der Rückgabewerte von Unterprogrammen ist beim AmigaBASIC voreinstellungsgemäß eine Zahl einfacher Genauigkeit. Library-Funktionen liefern jedoch in der Regel lange Ganzzahlwerte (32-Bit-Integer, Suffix "&") zurück, dies muß zunächst vor Aufruf der Funktionen vereinbart werden. Die Anweisungen

```
DECLARE FUNCTION BuildFileReq
uester&() LIBRARY
DECLARE FUNCTION BuildBoolReq
uester&() LIBRARY
DECLARE FUNCTION BuildColorPa
lette&() LIBRARY
```

erledigen dies für uns. Damit haben wir alle Vorbereitungen abgeschlossen und können die Funktionen aufrufen. Bitte beachten Sie, daß alle Aufrufparameter ebenfalls als lange Ganzzahlen übergeben werden müssen. Ebenso müssen alle Zeichenketten nach den Konventionen von C mit einem Null-Byte abgeschlossen sein, also etwa so:

```
a$ = "Dies ist ein String"+CHR$(0)
```

Auf einfache Zeichenketten werden immer nur Zeiger übergeben, einen solchen liefert die Funktion »SADD(a\$)«. Sollen mehrere Zeichenketten übergeben werden, so ist der Parameterwert ein Zeiger auf ein Array, dessen Elemente die Adressen der eigentlichen Zeichenketten enthalten. Dieser für BASIC eher ungewöhnliche Datentyp wird folgendermaßen konstruiert:

```
a$ = "String 1"+CHR$(0):z&(0)
= SADD(a$)
b$ = "String 2"+CHR$(0):z&(1)
= SADD(b$)
c$ = "String 3"+CHR$(0):z&(2)
= SADD(c$)
```

Das Array »z&(0..2)« enthält nun die Adressen der Zeichenketten »a\$«, »b\$« und »c\$«. Um die Adresse des Arrays zu übergeben, genügt es »z&()« zu schreiben.

Weitere für BASIC ungewöhnliche Übergabeparameter, die für unsere Library-Funktionen benötigt werden, sind die Zeiger auf eine Window- und Screen-Struktur. Erstere liefert problemlos die Funktion »win& = WINDOW(7)« zurück, für letztere müssen wir ein bißchen "peeken":

```
scr& = PEEKL(win& + 46)
```

liest den erforderlichen Wert aus der Window-Struktur für uns aus.

Mit diesem "Spezialwissen" ausgestattet, sollte nun jeder in der Lage sein, die Arbeitsweise unseres Beispielprogramms »requesterdemo.bas« (Listing 1) nachzuvollziehen. Zunächst wird ein neuer Screen mit drei Fenstern geöffnet. In jedem Fenster kann die Demonstration einer Funktion durch einfaches Anklicken ausgelöst werden. Nach Beendigung und Rückkehr aus der Library-Routine bekommen Sie das Resultat angezeigt, wertvolle Tips zur Aufbereitung und Weiterverwertung der Resultate sowie Datenkonstruktion und Parameterübergabe finden Sie im Listing. Beenden können Sie das Programm über Anwahl von »Stop« aus der Menüleiste, wobei natürlich zunächst eine Sicherheitsabfrage erfolgt.

C für Kenner

Kaum eine Sprache auf dem Amiga hat im professionellen Bereich eine so breite Akzeptanz und ein so hohes Entwicklungsniveau erfahren wie C. Nicht umsonst, denn C ist die "Muttersprache" des Amigas. Weite Teile seines Betriebssystemes wurden in C entwickelt, die Originaldokumentation sowie die meisten Beispielprogramme aus dem Hause Commodore und dem riesigen Public-Domain-Pool sind auf die Bedürfnisse der Programmierer, die die Sprache C benutzen, zugeschnitten. So wundert es auch nicht, daß der Zugang zum Library-Konzept (und damit auch die Benutzung unserer »Requester-Library«) über C am einfachsten zu bewerkstelligen ist. Erster Schritt ist immer das Öffnen der Library:

```
RequesterBase = OpenLibrary("requester.library",0L)
```

macht sie uns für alle Funktionsaufrufe zugänglich. Auch die verwendeten Datentypen bereiten keinerlei Schwierigkeiten, sie sind im Standardrepertoire jedes guten Compilers verankert. Für die Parameterübergabe können je nach verwendetem Compiler-Typ grundsätzlich zwei Wege beschritten werden:

– Der erste führt über den Stack und die Stub-Routinen (`requester_stubs.asm`, Listing 3 im letzten Heft). Zunächst werden alle Parameter auf dem Stack abgelegt, wonach in den Stub-Routinen die Library-Basis geladen wird und die Parameter wieder vom Stack in die entsprechenden Prozessorregister befördert werden. Nach Aufruf und Durchlauf der Library-Routine werden alle Registerinhalte restauriert und zum Hauptprogramm zurückgekehrt. Für den Compiler sieht es also so aus, als würde eine gewöhnliche externe Funktion aufgerufen werden. Die Stub-Routinen werden daher, sollten sie aus Versehen nicht angegeben werden, auch erst vom Linker reklamiert.

– Bei der zweiten Möglichkeit werden die Parameter und die Library-Basis direkt in die entsprechenden Prozessorregister eingeladen, ohne den Umweg über den Stack zu nehmen. Dazu müssen dem Compiler aber zunächst die Aufrufkonventionen bekanntgegeben werden. Dies geschieht durch die Anweisung `»#pragma«` im Header-File `requester.h` (Listing 4 aus AMIGA DOS 8/90). Leider funktioniert dies (noch) nicht für die Funktion `»BuildBoolRequester()«`, da zu viele Parameter erwartet werden. Für diese muß also noch der erste Weg beschritten werden, möglicherweise ändert sich dies bis

zum nächsten Update des Lattice-C-Compilers. Zur Erzeugung der Anweisungen vom Typ `»#pragma«` kann wieder unser File `».fd«` verwendet werden, die Konvertierung besorgt hier das Hilfsprogramm `»FD2-Pragma«` aus dem Lattice-Entwicklungspaket ([2], G20-6 ff.). Beide Methoden lassen sich problemlos mischen. Soll die erste verwendet werden, so ist einfach die Definition für `»#pragma«` wegzulassen, was auch durch Definition des Symbols `»NO_PRAGMAS«` erfolgen kann. Der komplette Aufruf für unser Beispiel `requesterdemo.c` (Listing 2) lautet also:

```
lc -v -w -csf -Lend+requester
_stubs.o requesterdemo
```

Ansonsten verhält sich das Programm genau wie das BASIC-Beispiel. Als Besonderheit hat das erste Fenster hier einen Close-Gadget zum Verlassen des Programmes. Außerdem wird gezeigt, wie man mehrere Fenster über einen gemeinsamen Intuition-Port abfragen kann ([3], Seite 167 ff.).

Assembler für Spezialisten

Meist aus Geschwindigkeitsgründen wird Maschinensprache gewählt. Ihre Benutzung fordert hohe Disziplin und erhebliches Fachwissen vom Programmierer, läßt dafür aber auch kein Vorhaben an möglichen Einschränkungen der

Entwicklungsumgebung scheitern. Alles ist mit Assembler machbar, und damit ist natürlich auch die Möglichkeit gegeben, beliebig viele und schwere Fehler in ein Programm einzubauen. Der Preis der Freiheit ist die gar nicht erst stattfindende Fehlerkontrolle (mit Ausnahme syntaktischer Prüfung) durch das Übersetzerprogramm, der Lohn dafür sehr kompakter und schneller Code. Demzufolge sollte in Maschinensprache wirklich nur dort programmiert werden, wo die Ansprüche über die Leistungsfähigkeit eines Hochsprachen-Entwicklungssystems hinausgehen. Unser Beispielprogramm soll nur grundsätzliche Methoden aufzeigen, geschwindigkeitsmäßig vermag es hier zwar nicht aufzutreffen, dafür beträgt aber die Codelänge nur etwa die Hälfte des Äquivalentes in C. Die Aufrufkonventionen beschränken sich auf das Allernötigste. Die Einsprung-Offsets für die Library-Funktionen werden im Include-File `»requester.i«` (Listing 1 der letzten Folge) mit dem Präfix `»_LVO«` festgelegt und können über die Makros

```
CALLLIB _LVOBuildFileRequester
```

oder

```
LINKLIB _LVOBuildFileRequester, RequesterBase(a4)
```

(aus `»exec/libraries.i«`) zur Anwendung gelangen. Die Registerwerte sind entsprechend der Definitionen im `».fd«`-File

zu laden. Das ablauffähige Programm wird aus dem Quellcode `»requesterdemo.asm«` (Listing 3) über folgende Aufrufe erzeugt:

```
asm -iINCLUDE: requesterdemo.
asm blink from requesterdemo.
o to requesterdemo lib lib:am
iga.lib sc sd nd
```

Wie auch in den vorangehenden Beispielen werden alle drei Funktionen unserer Library demonstriert.

Damit sind wir am Ende unseres Werkstatt-Artikels angelangt. Als Ergebnis halten wir eine funktionierende, universelle Library in Händen, zur Realisierung eines anspruchsvollen Benutzer-Interfaces. Die Anwendung der Routinen für eigene Programme steht damit allen Lesern offen!

(jb)

Literatur:

- [1] Commodore Büromaschinen GmbH: Amiga-Basic, Frankfurt, 1986
- [2] Lattice, Inc.: Lattice Amiga C Compiler User's Manual, Version 5.00, Volume I & II; Lombard, IL 1988
- [3] Commodore-Amiga, Inc.: Amiga ROM Kernel Reference Manual: Libraries & Devices, Revised and Updated, 1. Auflage, Addison-Wesley, 1989, ISBN 0-201-18187-8.

★ ★ ★ 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 Ausgabe 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!

Listings



```

1: /*=====
2: ***
3: *** RequesterDemo.asm
4: ***
5: *** by Franz-Josef Reichert
6: ***
7: /*=====
8:
9: * asm -iinclude: requesterdemo.asm
10: * blink from requesterdemo.o to requesterdemo lib
11: * lib:amiga.lib sc sd nd
12:
13: INCLUDE "exec/types.i"
14: INCLUDE "exec/memory.i"
15: INCLUDE "intuition/intuition.i"
16: INCLUDE "intuition/screens.i"
17: INCLUDE "libraries/dosextens.i"
18: INCLUDE "requester.i"
19:
20: W PALETTE EQU 0
21: W FILEREQ EQU 1
22: W BOOLREQ EQU 2
23: DEPTH EQU 2
24: MAXWINDOWS EQU 3
25: MAXCOLORS EQU 32
26: IDCMPFLAGS EQU MOUSEBUTTONS!CLOSEWINDOW
27: WINDOWFLAGS EQU WINDOWDRAG!WINDOWDEPTH!SMART_REFRESH!
IMMEZEROZERO!NOCAREREFRESH
28: MAXLINE EQU 104
29: FONT_HEIGHT EQU TOPAZ_EIGHTY
30:
31: STRUCTURE STACK_FRAME,0
32: APTR msg
33: APTR msgwindow
34: APTR fib
35: BPTR fl
36: ULONG iclass
37: UWORD icode
38: UWORD red
39: UWORD green
40: UWORD blue
41: LABEL SF_SIZEOF
42:
43: XDEF _SysBase
44:
45: XREF AbsExecBase, LinkerDB, CreatePort
46: XREF DeletePort, _sprintf
47:
48: XREF LVOFindTask, LVOOpenLibrary
49: XREF LVOCloseLibrary, LVOWaitPort, LVOGetMsg
50: XREF LVOReplyMsg, LVOForbid, LVOPermit
51: XREF LVOAllocMem, LVOFreeMem
52:
53: XREF LVOOpenScreen, LVOCloseScreen
54: XREF LVOOpenWindow, LVOCloseWindow
55: XREF LVOModifyIDCMP
56:
57: XREF LVOSetApen, LVOSetRast, LVOMove
58: XREF LVOText, LVOScrollRast
59:
60: XREF LVOLock, LVOUnLock, LVOExamine
61:
62: OPENLIB MACRO
63: lea.l \1Name(pc),a1
64: moveq #LIBRARY_VERSION,d0
65: CALLLIB LVOOpenLibrary
66: move.l d0,\1Base(a4)
67: beq \2
68: ENDM
69:
70: CLOSELIB MACRO
71: move.l \1Base(a4),a1
72: CALLLIB LVOCloseLibrary
73: ENDM
74:
75: ; -----
76: ; --- CODE SECTION ---
77: ; -----
78:
79: SECTION requesterdemo, CODE
80:
81: _startup
82: movem.l d5/a4-a6, -(sp)
83: lea.l LinkerDB, a4
84: move.l AbsExecBase.w, a6
85: move.l a6, SysBase(a4)
86: OPENLIB DOS, 1$
87: moveq #0, d5
88: sub.l a1, a1
89: CALLLIB LVOFindTask
90: move.l d0, a5
91: tst.l pr_CLI(a5)
92: bne.s 2$
93: lea.l pr_MsgPort(a5), a0
94: CALLLIB LVOWaitPort
95: lea.l pr_MsgPort(a5), a0
96: CALLLIB LVOGetMsg
97: move.l d0, d5
98: 2$ jsr main(pc)
99: tst.l d5
100: beq.s 3$
101: CALLLIB LVOForbid
102: move.l d5, a1
103: CALLLIB LVOReplyMsg
104: 3$ CLOSELIB DOS
105: 1$ movem.l (sp)+, d5/a4-a6
106: moveq #0, d0
107: rts
108:
109: _main
110: movem.l d2-d7/a2-a6, -(sp)
111: link a5, #-SF_SIZEOF

```

Listing: RequesterDEMO.ASM



```

112: move.l _SysBase(a4), a6
113:
114: OPENLIB Gfx, 1$
115: OPENLIB Intuition, 2$
116: OPENLIB Requester, 3$
117:
118: moveq #0, d0
119: move.l d0, -(sp)
120: move.l d0, -(sp)
121: jsr CreatePort(pc)
122: addq.l #8, sp
123: move.l d0, IDCMPPort(a4)
124: beq 4$
125:
126: jsr OpenAll(pc)
127: tst.l d0
128: beq 5$
129:
130: lea.l w(a4), a0
131: move.l W_BOOLREQ*4(a0), msgwindow(a5)
132: jsr Handle_BoolReq(pc)
133:
134: 6$ move.l IDCMPPort(a4), a0
135: CALLLIB LVOWaitPort
136: 7$ move.l IDCMPPort(a4), a0
137: CALLLIB LVOGetMsg
138: move.l d0, imsg(a5)
139: beq.s 6$
140: move.l d0, a1
141: im IDCMPWindow(a1), msgwindow(a5)
142: move.w im_Code(a1), icode(a5)
143: move.l im_Class(a1), iclass(a5)
144: CALLLIB LVOReplyMsg
145:
146: move.l iclass(a5), d0
147: cmpi.l #MOUSEBUTTONS, d0
148: beq.s 8$
149: cmpi.l #CLOSEWINDOW, d0
150: beq.s 9$
151: bra.s 7$
152:
153: 8$ ; --- CLASS == MOUSEBUTTONS
154: cmpi.w #SELECTUP, icode(a5)
155: bne.s 7$
156:
157: lea.l JumpTable(pc), a2
158: lea.l w(a4), a0
159: moveq #MAXWINDOWS-1, d1
160: 10$ move.l d1, d0
161: lsl.l #2, d0
162: move.l 0(a0, d0.w), d2
163: cmp.l msgwindow(a5), d2
164: beq.s 11$
165: dbra d1, 10$
166: bra.s 7$
167: 11$ move.l 0(a2, d0.w), a0
168: jsr (a0)
169: bra.s 7$
170:
171: 9$ ; --- CLASS == CLOSEWINDOW
172: movem.l d2-d3/a2-a3, -(sp)
173: moveq #2, d0
174: moveq #1, d1
175: moveq #0, d2
176: moveq #3, d3
177: lea.l taPlain(pc), a0
178: lea.l w(a4), a1
179: move.l W_BOOLREQ*4(a1), a1
180: lea.l EndText(pc), a2
181: lea.l EndGadgetText(pc), a3
182: LINKLIB LVOBuildBoolRequester, RequesterBase(a4)
183:
184: 4) movem.l (sp)+, d2-d3/a2-a3
185: tst.l d0
186: bne 7$
187:
188: 12$ CALLLIB LVOForbid
189: move.l IDCMPPort(a4), a0
190: CALLLIB LVOGetMsg
191: tst.l d0
192: beq.s 13$
193: move.l d0, a1
194: CALLLIB LVOReplyMsg
195: 13$ bra.s 12$
196: moveq #-1, d2
197: jsr CloseAll(pc)
198: CALLLIB LVOPermit
199:
200: 5$ move.l IDCMPPort(a4), -(sp)
201: jsr DeletePort(pc)
202: addq.l #4, sp
203:
204: 4$ move.l SysBase(a4), a6
205: 3$ CLOSELIB Requester
206: 2$ CLOSELIB Intuition
207: 1$ CLOSELIB Gfx
208: unlk a5
209: movem.l (sp)+, d2-d7/a2-a6
210: rts
211:
212: ; -----
213: ; --- END of main ---
214: ; -----
215: Handle Palette
216: movem.l d2-d5/a2-a3/a6, -(sp)
217: move.l GfxBase(a4), a6
218: move.l msgwindow(a5), a0
219: move.l wd_RPort(a0), a1
220: moveq #1, d0
221: CALLLIB LVOSetApen
222: lea.l colors(a4), a0
223: move.l a0, d0
224: move.l s(a4), a0
225: lea.l fmtstr2(pc), a1

```

Listing: RequesterDEMO.ASM


```

226:      move.l      RequesterBase(a4),a6
227:      CALLLIB      _LVObuildColorPalette
228:      tst.l        d0
229:      beq          2$
230:      move.l      msgwindow(a5),a0
231:      move.l      wd RPort(a0),a2
232:      lea.l        fmTstr6,a3
233:      jsr          PrintLine(pc)
234:
235:      moveq        #1,d2
236:      lsl.w        #DEPTH,d2
237:      moveq        #0,d3
238:
239: 3$      move.w      d3,d0
240:      lsl.w        #1,d0
241:      lea.l        colors(a4),a2
242:      move.w        0(a2,d0.w),d4
243:      move.w        d4,d0
244:      and.w        #$f,d0
245:      move.w        d0,-(sp)
246:      mulu.w        #100,d0
247:      divu.w        #15,d0
248:      move.w        d0,blue(a5)
249:
250:      move.w        d4,d0
251:      lsr.w        #4,d0
252:      and.w        #$f,d0
253:      move.w        d0,-(sp)
254:      mulu.w        #100,d0
255:      divu.w        #15,d0
256:      move.w        d0,green(a5)
257:
258:      move.w        d4,d0
259:      lsr.w        #8,d0
260:      and.w        #$f,d0
261:      move.w        d0,-(sp)
262:      mulu.w        #100,d0
263:      divu.w        #15,d0
264:      move.w        d0,red(a5)
265:
266:      move.w        blue(a5),-(sp)
267:      move.w        green(a5),-(sp)
268:      move.w        red(a5),-(sp)
269:      move.w        d3,-(sp)
270:      pea          fmTstr7(pc)
271:      pea          txtbuf(a4)
272:      jsr          sprintf(pc)
273:      lea.l        Z2(sp),sp
274:
275:      move.l      msgwindow(a5),a0
276:      move.l      wd RPort(a0),a2
277:      lea.l        txtbuf(a4),a3
278:      jsr          PrintLine(pc)
279:
280:      addq.w        #1,d3
281:      cmp.w        d2,d3
282:      blt          3$
283:      bra.s        1$
284:
285: 2$      move.l      msgwindow(a5),a0
286:      move.l      wd RPort(a0),a2
287:      lea.l        fmTstr5,a3
288:      jsr          PrintLine(pc)
289:
290: 1$      movem.l      (sp)+,d2-d5/a2-a3/a6
291:      rts
292:
293: Handle FileReq
294:      movem.l      d2/a2/a6,-(sp)
295:      move.l      GfxBase(a4),a6
296:      move.l      msgwindow(a5),a2
297:      move.l      wd RPort(a2),a1
298:      moveq        #1,d0
299:      CALLLIB      _LVOSetAPen
300:      move.l      wd RPort(a2),a1
301:      moveq        #0,d0
302:      CALLLIB      _LVOSetRast
303:      lea.l        dir(a4),a0
304:      move.l      a0,d0 ; dirname
305:      lea.l        node(a4),a0
306:      move.l      a0,d1 ; nodename
307:      lea.l        file(a4),a0
308:      move.l      a0,d2 ; filename
309:      move.l      a2,a0 ; window
310:      lea.l        fmTstr3(pc),a1 ; title
311:      LINKLIB      _LVObuildFileRequester,_RequesterBase(a
4)
312:      tst.l        d0
313:      beq          1$
314:      lea.l        file(a4),a0
315:      move.l      a0,d1
316:      moveq        #ACCESS_READ,d2
317:      LINKLIB      _LVOLock,_DOSBase(a4)
318:      move.l      d0,fl(a5)
319:      beq          1$
320:      move.l      #fib SIZEOF,d0
321:      MEMF PUBLIC MEMF CLEAR,d1
322:      LINKLIB      _LVOLAllocMem,_SysBase(a4)
323:      move.l      d0,fib(a5)
324:      beq          2$
325:      move.l      fl(a5),d1
326:      move.l      d0,d2
327:      LINKLIB      _LVOLExamine,_DOSBase(a4)
328:      tst.l        d0
329:      beq          3$
330:      move.l      msgwindow(a5),a1
331:      move.l      wd RPort(a1),a1
332:      moveq        #0,d0
333:      moveq        #FONT_HEIGHT,d1
334:      CALLLIB      _LVOMove
335:      move.l      msgwindow(a5),a1
336:      move.l      wd RPort(a1),a1
337:      lea.l        file(a4),a0
338:      jsr          strlen(pc)
339:      CALLLIB      _LVOText

```

Listing: RequesterDEMO.ASM



HERMES LoGG

NEU

Version 1.0 / 115 K © 1990 A. G. Kartsatos HERMES
Mit 10seitiger deutscher Anleitung

LoGG das neue, universelle Tool als Ersatz für das CLI. Als übersichtliches Programm, vollkommen mit der Maus gesteuert, bringt LoGG Komfort bei der Arbeit auf DOS-Ebene.

LoGG hat **48 Funktionen**, die gebräuchlichsten **CLI-Befehle**, im Programm eingebaut. Zudem können vom Benutzer bis zu **48 Macros** eingegeben werden, um Programme zu starten oder Befehle auszuführen.

LoGG kann **packen** und **entpacken**, mit den Befehlen ARC, LHARC, PAK, ZOO und WARP.

LoGG können Sie nach Ihren Wünschen die eigenen **Menü-Farben** geben und Ihre vorhandenen **Laufwerke** selber im Programm einbinden.

LoGG **spielt** IFF-Musik und **zeigt** IFF-Bilder.

LoGG hat einen eigenen **Format-** und **Diskcopy-Befehl**.

LoGG **erstellt** einen Katalog der Harddisk, speichert ihn oder schickt die Daten zum Drucker.

LoGG hat einen sehr schnellen **File Reader** mit einigen zusätzlichen **Menü-Funktionen** und gutem Scrolling.

LoGG **säubert** Ihre ASCII Files von Binary-Zeichen.

LoGG **sucht** Files nach Namen oder nur nach einem String ab und zeigt, wo sie zu finden sind.

LoGG **zeigt** Icons in jeder Grösse und in 2 bis zu 16 Farben.

LoGG eignet sich besonders für den **Einsteiger**, weil es logisch und leicht verständlich aufgebaut ist.

LoGG für den **Profi**, weil es die Arbeit schneller und leichter macht als im CLI!

HERMES IconLab

Version 1.3 / 164 K © 1988-1990 A. G. Kartsatos HERMES
Mit deutscher Anleitung und Übungsbeispiel

Räumen Sie auf mit den langweiligen Schubladen der Workbench, geben Sie Ihren Programmen eigene, selbst kreierte Icons! Mit IconLab ist dies möglich. Das Tool, das alles beherrscht, wenn es um Icons geht, ganz einfach mit der Maus zu bedienen.

IconLab macht Disk-, Drawer-, Tools- und Project-Icons. Entwerfen Sie mit dem **IconLab** Malprogramm neue Icons oder bearbeiten Sie bereits bestehende Icons in jeder Grösse und in 2 bis zu 16 Farben. **IconLab** wandelt Brushes in Icons oder Icons in Brushes und kann Brushes verkleinern, damit diese als Icons verwendet werden können. **IconLab** vereinigt zwei Einzel-Icons zu einem Doppel-Icon oder trennt Doppel-Icons. **IconLab** hat eine Text-Option, mit der jeder Font verwendet und beliebig plziert werden kann. Einfacher und sicherer als mit **IconLab** gehts nicht !!

Auf der Programm-Disk finden Sie noch zwei weitere HERMES-Programme: HERMIT, einen Screen Saver und COLORLAB, um Farben nach HEX-Zahlen zu bestimmen.

Ich bestelle:

☐ **LoGG** zum Preis von 35.- DM
☐ **IconLab** zum Preis von 25.- DM

In bar oder mit Check, plus 5.- DM Versandkosten.

Bei Nachnahme 10.- DM Versandkosten. Bitte Ihre Adresse nicht vergessen!

HERMES Software
Postfach 5
CH-6345 Neuheim/ZUG



Die rote Disk


```

340:      move.l    msgwindow(a5),a1
341:      move.l    wd_RPort(a1),a1
342:      moveq     #0,d0
343:      moveq     #FONT HEIGHT*2,d1
344:      CALLLIB   LVOMove
345:      move.l    Fib(a5),a0
346:      move.l    fib Size(a0),-(sp)
347:      pea       fmtStr4(pc)
348:      pea       txtbuf(a4)
349:      jsr       sprintf(pc)
350:      lea.l     I2(sp),sp
351:      move.l    msgwindow(a5),a1
352:      move.l    wd_RPort(a1),a1
353:      lea.l     txtbuf(a4),a0
354:      jsr       strlen(pc)
355:      CALLLIB   LVOText
356:
357: 3$      move.l    fib(a5),a1
358:      move.l    #fib SIZEOF,d0
359:      LINKLIB   _LVOfreeMem,_SysBase(a4)
360:
361: 2$      move.l    fl(a5),d1
362:      LINKLIB   _LVOUNlock,_DOSBase(a4)
363:
364: 1$      movem.l  (sp)+,d2/a2/a6
365:      rts
366:
367: Handle BoolReq
368:      movem.l    d2-d3/a2-a3/a5-a6,-(sp)
369:      move.l     GfxBase(a4),a6
370:      move.l     msgwindow(a5),a5
371:      move.l     wd_RPort(a5),a1
372:      moveq     #1,d0
373:      CALLLIB   LVOSetAPen
374:      move.l     wd_RPort(a5),a1
375:      moveq     #0,d0
376:      CALLLIB   LVOSetRast
377:      moveq     #2,d0
378:      moveq     #1,d1
379:      moveq     #0,d2
380:      moveq     #3,d3
381:      lea.l     taPlain(pc),a0
382:      move.l     a5,a1
383:      lea.l     RequesterText(pc),a2
384:      lea.l     GadgetText(pc),a3
385:      move.l     RequesterBase(a4),a6
386:      CALLLIB   _LVObuildBoolRequester
387:      move.w     d0,d2
388:      move.l     wd_RPort(a5),a1
389:      moveq     #125,d0
390:      moveq     #58,d1
391:      move.l     _GfxBase(a4),a6
392:      CALLLIB   _LVOMove
393:      move.w     d2,-(sp)
394:      pea       fmtStr1(pc)
395:      pea       txtbuf(a4)
396:      jsr       sprintf(pc)
397:      lea.l     I0(sp),sp
398:      move.l     wd_RPort(a5),a1
399:      lea.l     txtbuf(a4),a0
400:      jsr       strlen(pc)
401:      CALLLIB   LVOText
402:      movem.l    (sp)+,d2-d3/a2-a3/a5-a6
403:      rts
404:
405: PrintLine      ; a2 = rp a3 = buf
406:      movem.l    d2-d5/a6,-(sp)
407:      move.l     a2,a1
408:      moveq     #0,d0
409:      move.w     TextLine(pc),d1
410:      ext.l     d1
411:      move.l     _GfxBase(a4),a6
412:      CALLLIB   _LVOMove
413:      move.l     a2,a1
414:      move.l     a3,a0
415:      jsr       strlen(pc)
416:      CALLLIB   LVOText
417:      move.w     TextLine(pc),d0
418:      cmpi.w     #MAXLINE,d0
419:      blt.s     1$
420:      move.l     a2,a1
421:      moveq     #0,d0
422:      moveq     #FONT HEIGHT,d1
423:      move.l     d0,d2
424:      move.l     d0,d3
425:      move.l     #346,d4
426:      move.l     #118,d5
427:      CALLLIB   LVOScrollRaster
428:      bra.s     2$
429: 1$      lea.l     TextLine(pc),a0
430:      addq.w     #FONT HEIGHT,(a0)
431: 2$      movem.l  (sp)+,d2-d5/a6
432:      rts
433:
434: OpenAll
435:      movem.l    a2-a3/a6/d2,-(sp)
436:      lea.l     ns(pc),a0
437:      move.l     IntuitionBase(a4),a6
438:      CALLLIB   _LVOPenScreen
439:      move.l     d0,s(a4)
440:      beq.s     1$
441:      move.l     d0,a3
442:      moveq     #MAXWINDOWS-1,d2
443:      lea.l     nw(pc),a2
444: 2$      move.w     #nw SIZE,d0
445:      mulu.w     d2,d0
446:      lea.l     0(a2,d0.w),a0
447:      move.l     a3,nw Screen(a0)
448:      CALLLIB   _LVOPenWindow
449:      move.l     d2,d1
450:      lsl.w     #2,d1
451:      lea.l     w(a4),a0
452:      move.l     d0,0(a0,d1.w)
453:      beq.s     3$

```

Listing: RequesterDEMO.ASM

```

454:      move.l     d0,a0
455:      move.l     IDCMPPort(a4),wd_UserPort(a0)
456:      move.l     #IDCMPFLAGS,d0
457:      CALLLIB   LVOModifyIDCMP
458:      dbra      d2,2$
459:      moveq     #-1,d0
460: 1$      movem.l  (sp)+,a2-a3/a6/d2
461:      rts
462:
463: 3$      jsr       CloseAll(pc)
464:      moveq     #0,d0
465:      bra.s     1$
466:
467:
468: CloseAll      ; d2 = arg
469:      movem.l    d2/a2-a3/a6,-(sp)
470:      move.l     IntuitionBase(a4),a6
471:      lea.l     w(a4),a2
472: 1$      addq.w     #1,d2
473:      move.l     d2,d0
474:      cmpi.w     #MAXWINDOWS,d0
475:      bge.s     2$
476:      lsl.l     #2,d0
477:      move.l     0(a2,d0.w),a3
478:      move.l     a3,a0
479:      moveq     #0,d0
480:      move.l     d0,wd_UserPort(a0)
481:      CALLLIB   LVOModifyIDCMP
482:      move.l     a3,a0
483:      CALLLIB   _LVOCloseWindow
484:      bra.s     1$
485: 2$      move.l     s(a4),a0
486:      CALLLIB   _LVOCloseScreen
487:      movem.l    (sp)+,d2/a2-a3/a6
488:      rts
489:
490:
491: strlen
492:      move.l     a0,-(sp)
493:      moveq     #0,d0
494: 1$      tst.b     (a0)+
495:      beq.s     2$
496:      addq.l     #1,d0
497:      bra.s     1$
498: 2$      move.l     (sp)+,a0
499:      rts
500:
501: ; -----
502: ; --- DATA SECTION ---
503: ; -----
504:
505:      CNOP      0,4
506:
507: taPlain      dc.l 1$
508:              dc.w FONT HEIGHT
509:              dc.b 0,0
510: 1$           dc.b "topaz.font",0
511:
512: ns           dc.w 0,0,640,STDScreenHeight,DEPTH
513:              dc.b 0,1
514:              dc.w V HIRES,CUSTOMSCREEN
515:              dc.l taPlain,1$,0,0
516: 1$           dc.b "Requester-Library Demo - "
517:              dc.b "Bitte Fenster anklicken",0
518:
519: nw           dc.w 0,11,346,118
520:              dc.b 0,1
521:              dc.l 0,WINDOWFLAGS!WINDOWCLOSE,0,0,1$,0,0
522:              dc.w 0,0,0,0,CUSTOMSCREEN
523:
524:              dc.w 350,11,290,244
525:              dc.b 0,1
526:              dc.l 0,WINDOWFLAGS,0,0,2$,0,0
527:              dc.w 0,0,0,0,CUSTOMSCREEN
528:
529:              dc.w 0,131,346,124
530:              dc.b 0,1
531:              dc.l 0,WINDOWFLAGS,0,0,3$,0,0
532:              dc.w 0,0,0,0,CUSTOMSCREEN
533:
534: 1$           dc.b "ColorPalette Demo",0
535: 2$           dc.b "FileRequester Demo",0
536: 3$           dc.b "BoolRequester Demo",0
537:
538: RequesterText dc.l 1$,2$,3$
539: 1$           dc.b "Requester Library Version 33.12",0
540: 2$           dc.b "von F.J. Reichert, 1988,89,90 fuer"
541:              ,0
542: 3$           dc.b "AMIGA-DOS, DMV-Verlag",0
543:
544: EndText      dc.l 0,1$,0
545: 1$           dc.b "Wollen Sie wirklich aufhoeren?",0
546:
547: EndGadgetText dc.l 1$,0,2$
548: 1$           dc.b "= JA =",0
549: 2$           dc.b "= NEIN =",0
550:
551: GadgetText   dc.l 1$,2$,3$
552: 1$           dc.b "GADGET 0",0
553: 2$           dc.b "GADGET 1",0
554: 3$           dc.b "GADGET 2",0
555:
556: TextLine     dc.w TOPAZ EIGHTY
557: JumpTable    dc.l Handle_Palette,Handle_FileReq,Handle
558: BoolReq
559: RequesterName REQUESTERNAME
560: DOSName      DOSNAME
561: IntuitionName dc.b "intuition.library",0
562: GfxName      dc.b "graphics.library",0
563: fmtStr1      dc.b "Ergebnis: %d",0
564: fmtStr2      dc.b "Palettenaufruf aus Assembler",0
565: fmtStr3      dc.b "FJR's FileRequester Deluxe",0
566: fmtStr4      dc.b "%ld Bytes",0
567: fmtStr5      dc.b "Farbwerte bleiben unveraendert",0

```

Listing: RequesterDEMO.ASM


```

566: fntstr6      dc.b "Neue Farbwerte:",0
567: fntstr7      dc.b "%02d: R = %3d%% G = %3d%% B = %3d%%
568:              dc.b " | %x%x%x",0
569:
570: ; -----
571: ; --- BSS SECTION ---
572: ; -----
573:
574: SECTION _MERGED,BSS
575:
576: s              ds.l 1
577: w              ds.l 1 MAXWINDOWS
578: IDCMPort       ds.l 1
579: SysBase        ds.l 1
580: DOSBase        ds.l 1
581: RequesterBase  ds.l 1
582: IntuitionBase  ds.l 1
583: GfxBase        ds.l 1
584: Colors         ds.w MAXCOLORS
585: file           ds.b 256
586: dir            ds.b 128
587: node           ds.b 32
588: txtbuf         ds.b 256
589:
590: END
591:
592:

```

Listing: RequesterDEMO.ASM

```

1: /*=====
2: /*==
3: /*== RequesterDemo.BAS (AmigaBASIC)
4: /*==
5: /*== by Franz-Josef Reichert
6: /*==
7: /*== Folgende BitMaps muessen vorhanden sein:
8: /*== exec.bmap, dos.bmap, requester.bmap
9: /*==
10: /*=====
11:
12: CLEAR ,30000,10000
13:
14: DECLARE FUNCTION BuildFileRequester() LIBRARY
15: DECLARE FUNCTION BuildColorPalette() LIBRARY
16: DECLARE FUNCTION BuildBoolRequester() LIBRARY
17: DECLARE FUNCTION AllocMem() LIBRARY
18: DECLARE FUNCTION Lock() LIBRARY
19: DECLARE FUNCTION Examine() LIBRARY
20:
21: LIBRARY "requester.library"
22: LIBRARY "exec.library"
23: LIBRARY "dos.library"
24:
25: depth = 2:file$ = CHR$(0) + SPACE$(255)
26: dir$ = CHR$(0) + SPACE$(127)
27: node$ = CHR$(0) + SPACE$(31)
28:
29: CALL OpenAll
30: WINDOW OUTPUT 1
31: ts(0) = "REQUESTER.LIBRARY geladen! ="
32: ts(1) = "Zur Demonstration der entsprechenden"
33: ts(2) = "Routinen bitte Fenster anklicken!"
34: gs(0) = CHR$(0):gs(1) = CHR$(0):gs(2) = "OK"
35: CALL DisplayBoolRequester(ts(),gs(),rv&)
36: restart:
37: ON BREAK GOSUB ende
38: ON MOUSE GOSUB MakeDemo
39: MOUSE ON:BREAK ON
40: loop:
41: SLEEP:GOTO loop
42: MakeDemo:
43: MOUSE OFF
44: IF WINDOW(0) = 1 THEN CALL ColorPaletteDemo
45: IF WINDOW(0) = 2 THEN CALL FileRequesterDemo
46: IF WINDOW(0) = 3 THEN CALL BoolRequesterDemo
47: MOUSE ON
48: RETURN
49: ende:
50: MOUSE OFF:WINDOW OUTPUT 3
51: ts(0) = CHR$(0)
52: ts(1) = "Wollen Sie wirklich aufhoeren ?"
53: ts(2) = CHR$(0)
54: gs(0) = "JA":gs(1) = CHR$(0):gs(2) = "NEIN ="
55: CALL DisplayBoolRequester(ts(),gs(),rv&)
56: IF rv& = 2 THEN GOTO restart
57: CALL CloseAll:LIBRARY CLOSE:END
58:
59: SUB OpenAll STATIC
60: SHARED depth
61: SCREEN 2,640,256,depth,2
62: WINDOW 1,"ColorPalette Demo",(0,11)-(337,115),22,2
63: WINDOW 2,"FileRequester Demo",(350,11)-(631,241),22,2
64: WINDOW 3,"BoolRequester Demo",(0,131)-(337,241),22,2
65: END SUB
66:
67: SUB CloseAll STATIC
68: WINDOW CLOSE 1:WINDOW CLOSE 2
69: WINDOW CLOSE 3:SCREEN CLOSE 2
70: END SUB
71:
72: SUB ColorPaletteDemo STATIC
73: SHARED depth
74: WINDOW OUTPUT 1
75: title$ = "Palettenaufruf aus AmigaBasic" + CHR$(0)
76: w& = WINDOW(7):s& = PEEKL(w& + 46):t& = SADD(title$)
77: c& = AllocMem(32*2,65536+1)

```

Listing: RequesterDEMO.BAS

```

78: IF c& > 0 THEN
79:   w& = BuildColorPalette(c&,s&,t&)
80:   IF r& > 0 THEN
81:     PRINT "Neue Farbwerte:"
82:     FOR i = 0 TO (2*depth) - 1
83:       col% = PEEKW(r& + (i * 2))
84:       red% = (col% AND 3840) / 256
85:       green% = (col% AND 240) / 16
86:       blue% = (col% AND 15) / 1
87:       PRINT USING "###: R = ###% G = ###% B = ###% | $
";i,red% * 100 / 15,green% * 100 / 15,blue% * 100 / 15;
88:       PRINT HEX$(red%);HEX$(green%);HEX$(blue%)
89:     NEXT i
90:   ELSE
91:     PRINT "Farbwerte bleiben unveraendert"
92:   END IF
93:   CALL FreeMem(c&,32*2)
94: END IF
95: END SUB
96:
97: SUB BoolRequesterDemo STATIC
98: WINDOW OUTPUT 3
99: ts(0) = "Requester-Library Version 33.12"
100: ts(1) = "von F.J. Reichert, 1988,89,90 f" + CHR$(252)
101: + "r"
102: ts(2) = "AMIGA-DOS, DMV-Verlag"
103: gs(0) = "GADGET 0":gs(1) = "GADGET 1"
104: gs(2) = "GADGET 2"
105: CALL DisplayBoolRequester(ts(),gs(),erg&)
106: PRINT "Ergebnis: ";erg&
107: END SUB
108:
109: SUB DisplayBoolRequester (ts(),gs(),ret&) STATIC
110: ta$ = "topaz.font" + CHR$(0)
111: TextAttr(0) = SADD(ta$):TextAttr(1) = 8 * 65536&
112: FOR i = 0 TO 3
113:   IF LEFT$(ts(i),1) <> CHR$(0) THEN
114:     ts(i) = ts(i) + CHR$(0):t&(i) = SADD(ts(i))
115:   ELSE
116:     t&(i) = 0
117:   END IF
118:   IF LEFT$(gs(i),1) <> CHR$(0) THEN
119:     gs(i) = gs(i) + CHR$(0):g&(i) = SADD(gs(i))
120:   ELSE
121:     g&(i) = 0
122:   END IF
123: NEXT i
124: w& = WINDOW(7)
125: ret& = BuildBoolRequester(2,1,0,3,VARPTR(TextAttr(0)),w&,VARPTR(t&(0)),VARPTR(g&(0)))
126: END SUB
127:
128: SUB FileRequesterDemo STATIC
129: SHARED file$,dir$,node$
130: WINDOW OUTPUT 2
131: fib& = AllocMem(260,65536+1)
132: IF fib& <> 0 THEN
133:   title$ = "FileRequester in AmigaBasic" + CHR$(0)
134:   t& = SADD(title$):w& = WINDOW(7)
135:   rs& = BuildFileRequester(SADD(dir$),SADD(node$),SA
DD(file$),w&,t&)
136: IF rs& > 0 THEN
137:   fl& = Lock(SADD(file$),-2)
138:   IF fl& <> 0 THEN
139:     succ& = Examine(fl&,fib&)
140:     IF succ& <> 0 THEN
141:       length& = PEEKL(fib& + 124)
142:       bas$ = LEFT$(file$,INSTR(file$,CHR$(0)) - 1)
143:       PRINT bas$
144:       PRINT length&," Bytes"
145:     END IF
146:     CALL UnLock(fl&)
147:   END IF
148: END IF
149: CALL FreeMem(fib&,260)
150: END SUB

```

Listing: RequesterDEMO.BAS

```

1: /*=====
2: /*==
3: /*== RequesterDemo.C
4: /*==
5: /*== by Franz-Josef Reichert
6: /*==
7: /*=====
8:
9: /* lc -v -w -csf -Lcnd+requester_stubs.o */
10: /* requesterdemo
11:
12: #include <exec/types.h>
13: #include <exec/memory.h>
14: #include <intuition/intuition.h>
15: #include <intuition/intuitionbase.h>
16: #include <libraries/dosextens.h>
17: #include <proto/exec.h>
18: #include <proto/graphics.h>
19: #include <proto/intuition.h>
20: #include <proto/dos.h>
21: #include <stdio.h>
22: #include <string.h>
23: #include "requester.h"
24:
25: #define W_PALETTE 0
26: #define W_FILEREQ 1
27: #define W_BOOLREQ 2
28: #define DEPTH 2

```

Listing: RequesterDEMO.C


```

29: #define MAXWINDOWS 3
30: #define IDCMPFLAGS MOUSEBUTTONS|CLOSEWINDOW
31: #define MAXLINE 104
32: #define MAXCOLORS 32
33: #define FONT_HEIGHT TOPAZ_EIGHTY
34:
35: static VOID CloseAll(int);
36: static int OpenAll(VOID);
37: static VOID BoolRequesterDemo(VOID);
38: static VOID PrintLine(struct RastPort*,UBYTE*);
39:
40: static char file[256],dir[128],node[32];
41: static struct TextAttr taPlain = {
42:     "topaz.font",FONT_HEIGHT
43: };
44:
45: static struct NewScreen ns = {
46:     0,0,640,STDScreenHEIGHT,DEPTH,
47:     0,1,Hires,CUSTOMSCREEN,&taPlain,
48:     "Requester-Library Demo - Bitte Fenster anklicken!",
49: };
50:
51: static struct NewWindow nw[] = {
52:     {
53:         0,11,346,118,0,1,NULL,WINDOWCLOSE|WINDOWDRAG|
54:         WINDOWDEPTH|SMART_REFRESH|NOCAREREFRESH|
55:         GIMMEZEROZERO,NULL,NULL,"ColorPalette Demo",
56:         NULL,NULL,0,0,0,0,CUSTOMSCREEN
57:     },
58:     {
59:         350,11,290,244,0,1,NULL,WINDOWDRAG|WINDOWDEPTH|
60:         SMART_REFRESH|NOCAREREFRESH|GIMMEZEROZERO,NULL,NULL,
61:         "FileRequester Demo",NULL,NULL,0,0,0,0,CUSTOMSCREEN
62:     },
63:     {
64:         0,131,346,124,0,1,NULL,WINDOWDRAG|WINDOWDEPTH|
65:         SMART_REFRESH|NOCAREREFRESH|GIMMEZEROZERO,NULL,NULL,
66:         "BoolRequester Demo",NULL,NULL,0,0,0,0,CUSTOMSCREEN
67:     },
68: };
69:
70: static UBYTE *RequesterText[] = {
71:     "Requester-Library Version 33.12",
72:     "von F.J. Reichert, 1988,89,90 fuer",
73:     "AMIGA-DOS, DMV-Verlag",
74: };
75:
76: static UBYTE *EndText[] = {
77:     NULL,"Wollen Sie wirklich aufhoeren?",NULL
78: };
79:
80: static UBYTE *EndGadgetText[] = {
81:     "JA","NEIN"
82: };
83:
84: static UBYTE *GadgetText[] = {
85:     "GADGET 0","GADGET 1","GADGET 2"
86: };
87:
88: int TextLine = TOPAZ_EIGHTY;
89: static char txtbuf[256];
90: static UWORD colors[MAXCOLORS];
91: static struct Screen *s;
92: static struct Window *w[MAXWINDOWS];
93: static struct MessagePort *IDCMPPort;
94: static struct RequesterBase *RequesterBase;
95: extern struct IntuitionBase *IntuitionBase;
96: extern struct GfxBase *GfxBase;
97:
98: static VOID CloseAll(int arg) {
99:     while(--arg >= 0) {
100:         w[arg]->UserPort = NULL;
101:         ModifyIDCMP(w[arg],NULL);
102:         CloseWindow(w[arg]);
103:     }
104:     CloseScreen(s);
105:     return;
106: }
107:
108: static int OpenAll() {
109:     int i;
110:     if(s = OpenScreen(&ns)) {
111:         for(i = 0; i < MAXWINDOWS; i++) {
112:             nw[i].Screen = s;
113:             if((w[i] = OpenWindow(&nw[i])) == NULL) {
114:                 CloseAll(i);
115:                 return(0);
116:             }
117:             w[i]->UserPort = IDCMPPort;
118:             ModifyIDCMP(w[i],IDCMPFLAGS);
119:         }
120:         return(i);
121:     }
122:     return(0);
123: }
124:
125: static VOID BoolRequesterDemo() {
126:     int ret;
127:     SetAPen(w[W_BOOLREQ]->RPort,1L);
128:     SetRast(w[W_BOOLREQ]->RPort,0L);
129:     ret = BuildBoolRequester(2L,1L,0L,3L,&taPlain,
130:         w[W_BOOLREQ],RequesterText,GadgetText);
131:     Move(w[W_BOOLREQ]->RPort,125L,58L);
132:     sprintf(txtbuf,"Ergebnis: %d",ret);
133:     Text(w[W_BOOLREQ]->RPort,txtbuf,(LONG)strlen(txtbuf));
134: }
135:
136: static VOID PrintLine(struct RastPort *rp,UBYTE *buf) {
137:     Move(rp,0L,(LONG)TextLine);
138:     Text(rp,buf,(LONG)strlen(buf));
139:     if(TextLine >= MAXLINE) ScrollRaster
140:         (rp,0L,FontHeight,0L,0L,346L,118L);
141:     else TextLine += FontHeight;
142:     return;

```

Listing: RequesterDEMOC

```

143: }
144:
145: void stdargs main
146: (unsigned int argc,unsigned char *argv[]) {
147:     struct IntuiMessage *img;
148:     struct Window *msgwindow;
149:     struct FileInfoBlock *fib;
150:     BPTR fl; int i,red,green,blue;
151:     UWORD code; ULONG class;
152:
153:     if(GfxBase = (struct GfxBase*)
154:         OpenLibrary("graphics.library",LIBRARY_VERSION)) {
155:         if(IntuitionBase = (struct IntuitionBase*)
156:             OpenLibrary("intuition.library",LIBRARY_VERSION)) {
157:             if(RequesterBase = (struct RequesterBase*)
158:                 OpenLibrary(REQUESTERNAME,LIBRARY_VERSION)) {
159:                 if(IDCMPPort = CreatePort(NULL,0L)) {
160:                     if(OpenAll()) {
161:                         BoolRequesterDemo();
162:                         FOREVER {
163:                             WaitPort(IDCMPPort);
164:                             while(img = (struct IntuiMessage*)
165:                                 GetMsg(IDCMPPort)) {
166:                                 msgwindow = img->IDCMPWindow;
167:                                 code = img->Code;
168:                                 class = img->Class;
169:                                 ReplyMsg(&img->ExecMessage);
170:                                 switch(class) {
171:                                     case MOUSEBUTTONS:
172:                                         if(code == SELECTUP) {
173:                                             if(msgwindow == w[W_PALETTE]) {
174:                                                 SetAPen(msgwindow->RPort,1L);
175:                                                 if(BuildColorPalette(colors,s,
176:                                                     "Palettenaufruf aus 'Ch'",
177:                                                     sprintf(txtbuf,"%s","Neue Farbwerte:");
178:                                                     PrintLine(msgwindow->RPort,txtbuf);
179:                                                     for(i = 0; i < 1 << DEPTH; i++) {
180:                                                         red = ((colors[i] >> 8)
181:                                                             & 0x00ff) * 100 / 15;
182:                                                         green = ((colors[i] >> 4)
183:                                                             & 0x00ff) * 100 / 15;
184:                                                         blue = ((colors[i]
185:                                                             & 0x00ff) * 100 / 15;
186:                                                         sprintf(txtbuf,"%02d: R = %3d% G = %3d%
187:                                                             B = %3d% | %$x%$x",
188:                                                             i,red,green,blue,(colors[i] >> 8)
189:                                                             & 0x00ff,(colors[i] >> 4) & 0x00ff,
190:                                                             colors[i] & 0x00ff);
191:                                                         PrintLine(msgwindow->RPort,txtbuf);
192:                                                     }
193:                                                     else PrintLine(msgwindow->RPort,
194:                                                         "Farbwerte bleiben unveraendert");
195:                                                 }
196:                                             else if(msgwindow == w[W_FILEREQ]) {
197:                                                 SetAPen(msgwindow->RPort,1L);
198:                                                 SetRast(msgwindow->RPort,0L);
199:                                                 if(BuildFileRequester(dir,node,file,
200:                                                     msgwindow,"FJR's FileRequester Deluxe")) {
201:                                                     if(fl = Lock(file,ACCESS_READ)) {
202:                                                         if(fib = (struct FileInfoBlock*)
203:                                                             AllocMem(sizeof(struct FileInfoBlock),
204:                                                             MEMF_PUBLIC|MEMF_CLEAR)) {
205:                                                             if(Examine(fl,fib)) {
206:                                                                 Move(msgwindow->RPort,
207:                                                                     0L,FontHeight);
208:                                                                 Text(msgwindow->RPort,
209:                                                                     file,(LONG)strlen(file));
210:                                                                 Move(msgwindow->RPort,
211:                                                                     0L,FontHeight * 2);
212:                                                                 sprintf(txtbuf,"%ld Bytes",
213:                                                                     fib->fib_size);
214:                                                                 Text(msgwindow->RPort,
215:                                                                     txtbuf,(LONG)strlen(txtbuf));
216:                                                                 FreeMem((UBYTE*)fib,
217:                                                                     sizeof(struct FileInfoBlock));
218:                                                                 Unlock(fl);
219:                                                             }
220:                                                         }
221:                                                     }
222:                                                     else if(msgwindow == w[W_BOOLREQ]) {
223:                                                         BoolRequesterDemo();
224:                                                     }
225:                                                     break;
226:                                                 case CLOSEWINDOW:
227:                                                     if(BuildBoolRequester(2L,1L,0L,3L,&taPlain,
228:                                                         w[W_BOOLREQ],EndText,EndGadgetText) == 0)
229:                                                         goto ENDE;
230:                                                     break;
231:                                                     default:
232:                                                         break;
233:                                                 }
234:                                             }
235:                                         }
236:                                     }
237:                                 }
238:                             }
239:                         ENDE: Forbid();
240:                         while(img = (struct IntuiMessage*)
241:                             GetMsg(IDCMPPort)) ReplyMsg(&img->ExecMessage);
242:                         CloseAll(MAXWINDOWS);
243:                         Permit();
244:                         DeletePort(IDCMPPort);
245:                         CloseLibrary(&RequesterBase->rb_LibNode);
246:                         CloseLibrary(&IntuitionBase->LibNode);
247:                         CloseLibrary(&GfxBase->LibNode);
248:                     }
249:                 }
250:             }
251:         }
252:     }
253:     return;
254: }
255:

```

Listing: RequesterDEMOC

WINNER II – die neue A2000 SCSI-Filecard

AutoBoot bereits unter Kickstart 1.2 direkt von FastFile

Durch SCSI-II 16-bit-Technik sind Datenübertragungen über 1 MB/Sekunde möglich. Abschaltbar mit durchgeführtem Bus.

Einführungspreise

32 MB Seagate ST 138-N/28 mS. Filecard	1065,-
48 MB Seagate ST 177-N/28 mS. Filecard	1198,-
40 MB Quantum 40S/19 mS. Filecard	1338,-
80 MB Quantum 80S/19 mS. Filecard	1838,-
105 MB Quantum 105S/19 mS. Filecard	2098,-
120 MB Quantum 120S/15 mS. Filecard	2298,-
170 MB Quantum 170S/15 mS. Filecard	2768,-
210 MB Quantum 210S/15 mS. Filecard	2998,-

Winner-Autoboot-Filecard

31 MB Filecard	898,-
47 MB Filecard	1098,-
66 MB Filecard	1298,-

Autoboot-Harddisk

31 MB A500/1000	1098,-
47 MB A 500/1000	1298,-
63 MB A 500/1000	1498,-

Filecard und Harddisk bereits formatiert und mit WB 1.3 installiert. Harddisk-Gehäuse für Amiga 500/1000 als Monitoruntersatz.

512 KB-Ramkarte

für A 500	129,-
-----------	-------

abschaltbar, mit Uhr/Akku, Megabittechnik

512 KB-Ramkarte

für Amiga 500	99,-
---------------	------

abschaltbar, Megabittechnik, ohne Uhr.

2 MB-Ramkarte für

A 500	498,-
-------	-------

Erweiterung auf 2,3 MB für Amiga 500. Abschaltbar, mit Uhr, alles gesockelt.

8 MB-Ramkarte für A2000

512 KB bestückt	425,-
1,0 MB bestückt	475,-
2,0 MB bestückt	595,-
4,0 MB bestückt	998,-
2 MB-Box für A500/A1000	668,-
4 MB-Box für A500/A1000	1098,-

3.5"-Winner-Drive

extern mit durchgeführtem Bus und abschaltbar, amigafarbenes Metallgehäuse	169,-
--	-------

3.5" intern für

Amiga 500	165,-
-----------	-------

komplett mit original Auswurfaste

3.5" intern für

Amiga 2000	145,-
------------	-------

Mit Zubehör und Einbauanleitung.

3.5" intern für

Amiga 2000	219,-
------------	-------

Als drittes 3,5" Laufwerk, komplett mit Interface und 5,25" Einbaurahmen.

5.25" Laufwerk für alle Amiga

Mit Chinon FZ 506 oder FR 506	199,-
Mit Teak FD 55 GFR	229,-

extern mit durchgeführtem Bus, 40/80 Track schaltbar und abschaltbar.

5.25" intern für

Amiga 2000	269,-
------------	-------

Als DF2-Laufwerk mit Interface und Bootselector DF0/DF2. 40/80 Track schaltbar.

Elektr. Bootselector

DF0-DF3	48,-
---------	------

Es kann von allen Laufwerken gebootet werden, das interne LW ist abschaltbar. Kein Löten erforderlich.

DF2-Interface für

Amiga 2000	89,-
------------	------

Mit Bootselector DF0-DF2, für ein drittes Laufwerk im Amiga 2000 intern.

Winner-Midi A500/2000	89,-
-----------------------	------

Winner-Midi A1000	89,-
-------------------	------

Sounddigitizer

A 500/2000	89,-
------------	------

Mit Software

ROM-ROM-Umschaltplatine

komplett mit ROM 1.2	99,-
komplett mit ROM 1.3	99,-

Amiga 2000 C	1998,-
--------------	--------

mit zwei 3.5" Laufwerken

Citizen Swift 24	998,-
------------------	-------

mit Colour Kit

NEC Pinwriter P60	1890,-
-------------------	--------

NEC Pinwriter P70	2390,-
-------------------	--------

Colour Kit P60 / P70	198,-
----------------------	-------

Digi-Splitt Junior	428,-
--------------------	-------

Der Testsieger. Vollautomatischer RGB-Splitter, jetzt mit S-VHS

Pal-Genlock V 1.3	548,-
-------------------	-------

Überblenden von einer Bildquelle zur anderen.

Y-C Genlock	1120,-
-------------	--------

Baugleich Hama. RGB-Bandbreite 10 MHz. S-VHS Anschluß

OMTI-Adapter f. A2000	65,-
-----------------------	------

Filecardblech für 3.5" HD	15,-
---------------------------	------

OMTI 5528	189,-
-----------	-------

AutoBoot-Platine	125,-
------------------	-------

für Amiga 2000 mit Software und Anleitung

Autoboot-Set f. A2000	359,-
-----------------------	-------

OMTI 5528, Adapter, Kabelsatz, Autoboot-modul und Software

Autoboot-Set f. A500	359,-
----------------------	-------

OMTI 5528, Hostadapter mit Bus, Kabelsatz und Software

Amiga 500/1000

Autobootmodul	159,-
---------------	-------

zur Nachrüstung von Harddisk

Festplattengehäuse

A 500	68,-
-------	------

Schaltnetzteil	109,-
----------------	-------

A2090A Turbo-Chip-Satz	149,-
------------------------	-------

doppelte Geschwindigkeit

A2090 alt TurboAutobootkarte	159,-
------------------------------	-------

zusätzlich unter Kick. 1.2

Fast Lightning	29,-
----------------	------

Kopierprogramm für 2-4 Laufwerke

TurboCopy V 2.0	19,-
-----------------	------

White Lightning	19,-
-----------------	------

Kopierprogramm für 1 Laufwerk

BootBlockGenerator	19,-
--------------------	------

Zum Erstellen eines eigenen Vorspanns in den Bootblock, auch mit Sound

De Luxe View 4.1	378,-
------------------	-------

Digi View Gold	378,-
----------------	-------

Quarterback 4.0 neu	125,-
---------------------	-------

Festplatten-Backup

3.5" No Name 2DD 10 Stück	11,-
---------------------------	------

3.5" No Name 2DD 100 Stück	99,-
----------------------------	------

Halle 10 Stand 310

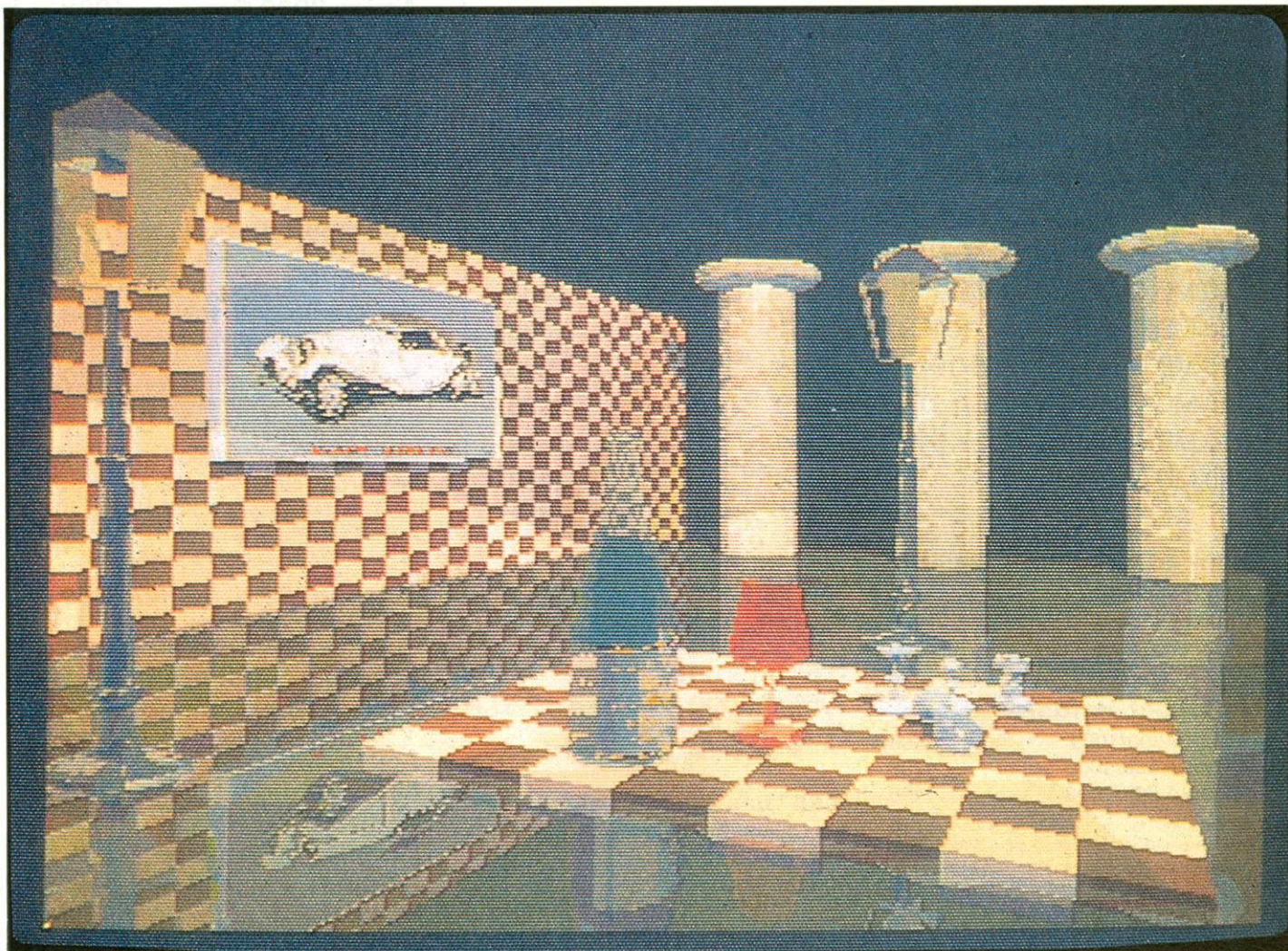
Auch wir sind auf der

AMIGA
Köln 90

vom 8.-11. November

Vesalia Computer

Industriestraße 25 · 4236 Hamminkeln · Tel. 02852/1068 · Fax 02852/1802 · Mo. – Fr. 9–17 Uhr



Bernd Rudolf

Reflections-Werkstatt

In diesem Teil unseres Reflections-Workshops wollen wir Ihnen die letzten Funktionen erklären und hilfreiche Arbeitstips geben.

Beginnen wir diesmal mit einem Tip, der vor allem für Amiga-Besitzer mit nur 512 kByte von großer Bedeutung ist. Wie Sie eventuell bemerkt haben werden, schluckt unsere Aufgabe vom letzten Mal aufgrund der zwei Rotationskörper einiges an Speicher.

Raytracing mit dem A 500

Doch auch mit 512 kByte Arbeitsspeicher ist das kein Problem: Formatieren Sie eine neue Diskette in DF0: mit dem

Kursfahrplan		
Teil 1: Grundsätzliche Bedienung		
Teil 2: Material und Licht		
Teil 3: Werkstoffe und Oberfläche		
Teil 4: Reflections in der Praxis		

Namen »REF«, und führen Sie den Befehl »install df0:« aus. Richten Sie ein Directory mit dem Namen »reflections«, eines mit dem Namen »s« und ei-

nes mit »devs« ein. In »devs« eröffnen Sie ein Unter-Directory namens »keymaps«, in das Sie aus dem entsprechenden Directory Ihrer Workbench-Di-

kette die Datei »d« kopieren. Mit einem Texteditor erzeugen Sie ein Textfile, das folgende Zeile enthält:

Run Manager.

Speichern Sie diesen Text als »Startup-Sequence« im S-Directory Ihrer Diskette ab. Jetzt müssen Sie nur noch alle Programme und Verzeichnisse aus dem Verzeichnis »reflections« der Reflections-Diskette in das gleichnamige Verzeichnis Ihrer Diskette kopieren.

Vergessen Sie nicht, daß auch das Programm »Manager« aus dem Haupt-Directory dazugehört. Am besten ist es, wenn Sie zum Kopieren ein Programm wie »CLI-Mate« nehmen, das gleich ganze Directories samt Unter-Directories kopiert. Wenn Sie nun diese Diskette booten, erscheint auf dem ersten CLI gleich der Manager. Ziehen Sie das Fenster einmal nach rechts. Wie Sie sehen, hat sich das CLI nach dem Aufruf mit dem Prompt »>« zurückgemeldet. Geben

Sie jetzt »EndCLI« ein, und Sie sparen weitere 50 kByte, auch wenn das CLI-Fenster sich nicht schließt. Da die deutsche Tastatur kaum Speicher belegt, können Sie sie einbauen, indem Sie vor die bestehende Zeile der Startup-Sequence die Zeile »SetMap d« einfügen. Vergessen Sie aber nicht, den Befehl aus dem c-Directory Ihrer Workbench-Diskette auf die Arbeitsdiskette zu kopieren. Er muß entweder im Haupt-Directory oder im c-Directory stehen. Nun haben Sie eine Arbeitsdiskette, die es Ihnen erlaubt, bei »Construct« mit über 2000 Punkten und Objekten zu arbeiten.

»Geomet«

Wenn Sie sich inzwischen ein wenig mit »Construct« vertraut gemacht haben, wird Ihnen aufgefallen sein, daß wir einige Funktionen noch gar nicht erwähnt haben. Die wichtigsten sind im Menü »Geomet« zu finden: Mit »Verändern« kann man Körper in ihrer Größe verändern oder verzerren. Wenn Sie sich für »Vergrößern« entscheiden, können Sie den Körper mit dem Mauszeiger an seiner Position »abholen« und durch Festhalten der linken Maustaste insgesamt vergrößern. Ach..., werden jetzt einige sagen, doch es gibt einen Unterschied zwischen »Vergrößern« und »Verzerren«!

Mit »Vergrößern« ist in diesem Fall das verhältnismäßige Vergrößern aller Seiten des Körpers gemeint. Beträgt zum Beispiel das Verhältnis der Seitenlängen eines Rechtecks 2:3, so bleibt dieses Ver-

hältnis beim Vergrößern gleich. Mit »Verzerren« können Sie hingegen einen Körper in jede Richtung beliebig strecken. Wenn Sie die linke Maustaste wieder loslassen, können Sie den Körper neu platzieren oder ihn nochmals vergrößern beziehungsweise verzerren. Der nächste Menüpunkt ist »Drehen«. Ich glaube, dazu muß man nichts weiter sagen, außer, daß es ebenso wie bei den anderen beiden Menüpunkten des Menüs »Geomet« darauf ankommt, von welcher Seite man den Körper gerade betrachtet. Kommen wir zum Thema »Knittern«. Mit dieser Funktion können Sie erreichen, daß Ihr symmetrisch kreierter Körper seine Symmetrie verliert, falls sie nicht erwünscht ist. Sie müssen sich dabei zwischen »Modus0« und »Modus1« entscheiden. Welcher Modus der geeignetere ist, entscheidet allein Ihr Geschmack. Probieren Sie beide einfach einmal aus. Seien Sie aber vorsichtig, denn einmal zerknitterte Körper kann man nicht mehr bügeln! Bei der Eingabe der Knitterstärke können Sie zwischen positiven und negativen Werten per Mausbewegung entscheiden. Dadurch beeinflussen Sie die Richtung der Knitterung.

Mens sana in corpore sano

Jetzt bleiben nur noch die Funktionen des Menüs »Körper« und vier Funktionen des Licht-Menüs zu besprechen. Wie der Name schon sagt, ent-

hält das Körper-Menü Tools, die der Verwaltung und Definition von Körpern dienen.

Als erstes finden wir dort »Zeigen«. Mit »Zeigen« können Sie sich Informationen über einzelne Körper oder auch alle Körper verschaffen, die dann im Informationsfenster unterhalb der Menüleiste angezeigt werden. Wenn Sie sich im Auswahlfenster für einen bestimmten Körper entscheiden, wird angezeigt, aus welchem Material er besteht. Handelt es sich dabei um ein Material, das eine Textur beinhaltet, so wird auch noch dargestellt, wie die Textur im Verhältnis zum Körper proportioniert ist. Die Funktion »AddNeu« ist sehr praktisch: Stellen Sie sich einmal vor, Sie haben einen Tisch »gebastelt«, der aus vier Beinen und einer Tischplatte (wie ein Tisch halt so aussieht) besteht und auf dem eine Vase steht. Jetzt möchten Sie den ganzen Tisch etwas verschieben, vergrößern oder sogar drehen. Das wäre eine Sisyphusarbeit, die man umgehen kann, indem man »AddNeu« anwählt. Bevor man nach einem neuen Namen gefragt wird, der beispielsweise »ganzer_Tisch« sein könnte, erscheint das Auswahlfenster mit allen vorhandenen Körpern, aus denen man durch Anklicken die gewünschten aussuchen kann. In diesem Fall wären es die Körper »Tischbein1« bis »Tischbein4«, »Tischplatte« und »Vase«. Wenn man abschließend die rechte Maustaste drückt, so erhält man einen neuen Körper namens »ganzer_Tisch«, den man genauso ansprechen

kann wie jeden anderen Körper auch. Allerdings beziehen sich alle Funktionen gleich auf die ausgewählten sechs Körper. Das ändert aber nichts an der Tatsache, daß man auch noch jeden der fünf einzeln ansprechen kann, wenn man feststellt, daß vielleicht ein Tischbein nicht an der richtigen Stelle sitzt. Anders könnte man sagen, die einzelnen Bestandteile des neu definierten Körpers gehen nicht verloren. Dabei ist jedoch unbedingt hinzuzufügen, daß unter Umständen Speicherprobleme auftreten können, wenn der Umfang der auf solche Weise zusammengestellten Körper zu groß wird. Ich habe es zwar nicht nachgemessen, aber ich hatte den Eindruck, daß sich die Rechenzeit erheblich erhöhte, nachdem ich die drei Säulen aus Bild 1 zusammengefaßt hatte, um sie verschieben zu können. Wenn Sie also Ihre Szenen fertig haben, empfiehlt es sich, mit der Funktion »Sub« die vier Tischbeine, die Tischplatte und die Vase wieder vom Körper »ganzer_Tisch« abzuziehen. Klicken Sie dazu zuerst »ganzer_Tisch« an und dann nacheinander die abzuziehenden Körper »Tischbein1«...

Sie beenden die Funktion wie immer mit der rechten Maustaste. Der Körper »ganzer_Tisch« ist hinterher verschwunden. Eine andere Möglichkeit ist »K_löschen« (siehe unten). Kommen Sie um Himmels Willen nicht auf die Idee, die Objekte zu löschen, denn wie gesagt greifen alle Funktionen auch auf alle dazugehörigen Körper zu. Ihre Szene würde



Bild 1. Diese Szene wurde mit den neuen Materialien erstellt

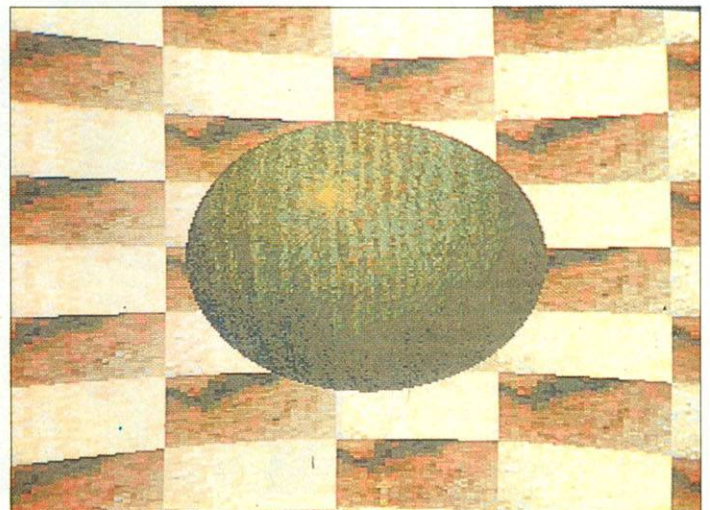


Bild 2. Hintergrund mit einer Kugeltextur belegt



Bild 3 bis Bild 6.
Beispiele für
eine Etikett-,
Genlock-,
Material- und
Punkttextur

also in diesem Fall um einen Tisch ärmer. Um beim Addieren und Subtrahieren zu bleiben, wollen wir noch »Add« erwähnen. »Add« verhält sich ebenso wie »AddNeu«, mit dem Unterschied, daß kein neuer Körper erzeugt wird, sondern einer zu einem bestehenden hinzugefügt wird. Wenn Sie etwa nachträglich noch eine Flasche auf den Tisch stellen, können Sie sie mit »Add« in »ganzer_Tisch« einbinden. Die Bedienung der Funktion ist mit der von »Sub« identisch.

»Define« ist ein besonderer Leckerbissen, denn mit »Define« lassen sich Objekte aus Körpern ausschneiden und zu Körpern machen. Somit ist es möglich, eine Lampe als Rotationskörper zu konzipieren und sie hinterher mittels »Define« in ihre Bestandteile zu zerlegen, denen man verschiedene Materialien zuordnen kann. Die Auswahl der Objekte erfolgt auf grafische Weise. Dabei stehen Ihnen zwei Modi zur Verfügung: Bei »Modus_0« werden all jene Dreiecke angesprochen, die vollständig in dem von Ihnen aufgezogenen Rahmen liegen. Bei »Modus_1« werden alle Dreiecke selektiert, die mit mindestens einem Punkt im Rahmen liegen.

Eine praktische Hilfe ist auch »Doppeln«. Bleiben wir einmal bei unserem Tisch. Angenommen Sie wollen gerade die vier Beine mit »Rot.Körp« konstruieren. Spätestens beim zweiten Bein werden Sie feststellen, daß Sie vergessen haben, wie viele Kästchen das erste lang und breit war. Da Sie aber nicht mehr nachsehen können,

wie das Verhältnis von Breite und Höhe war, bleibt keine andere Möglichkeit, als das erste Bein wieder zu löschen und nochmal von vorne anzufangen. Doch halt!

Wir haben ja »Doppeln«! Statt jedes Bein neu zu erstellen, doppeln wir das erste dreimal. Wundern Sie sich nicht, wenn Sie auch weiterhin nur ein Bein sehen, denn alle vier Beine befinden sich an derselben Stelle im Raum. Erst wenn Sie sie verschieben, sind sie zu erkennen. Den Menüpunkt »Material« haben wir bereits an früherer Stelle besprochen. Nachdem wir die Funktionen hinter uns gelassen haben, werden wir uns noch einmal eingehend mit Texturen beschäftigen. Zuvor müssen wir aber noch Löschen lernen. Keine Angst, wir werden Grisu, dem Feuerwehrdrachen, keine Konkurrenz machen. Es geht vielmehr darum, mit »K_löschen« Körper und mit

»O_löschen« Objekte zu löschen. Wo ist da der Unterschied? Ganz einfach, mit »K_löschen« löscht man eigentlich nur die Beschreibung eines Körpers. Er selber bleibt aber bestehen. So wird das ganze Gewusel von Körpern in einer umfangreichen Szene übersichtlicher. Der Casus Knaxus ist jedoch der, daß Sie sich die weitere Arbeit mit den Objekten des Körpers abschminken können.

Was nun?

Da blickt doch keiner mehr durch! Gelöscht und doch nicht gelöscht? Wie soll man das denn verstehen? Ganz einfach. Wie wir im ersten Teil des Kurses gelernt haben, arbeiten wir mit Körpern, die »Construct« intern in Objekte, sprich Dreiecke und Kugeln, zerlegt. Die Brücke zwischen den Grundobjekten und dem Benutzer sind die Körper. Ähnlich wie wir mit »AddNeu« aus mehreren einzelnen Körpern einen umfassenden neuen ge-

schaffen haben, verhält sich auch der Zusammenhang zwischen Objekten und Körpern. Wir haben uns in der Hierarchie nur eine Etage tiefer gegeben. Mit »K_löschen« sprengen wir also die Brücke, die Körperbeschreibung heißt. Dabei bleiben die Objekte, die »Construct« erzeugt hat und verwaltet, erhalten. Wir können sie nur nicht mehr modifizieren, da alle Seile gekappt sind. »O_löschen« geht hingegen gleich den Objekten an den Kragen. Es ist einzusehen, daß für die gelöschten Objekte eines Körpers keine Körperbeschreibung mehr erforderlich ist. Diese Funktion hat zur Folge, daß alles, was mit der ausgewählten Körperbeschreibung in Verbindung steht, flöten geht. Exitus. Was im einzelnen die Funktion »Schneiden« bewirkt, war leider nicht genau festzustellen. Die Theorie im Handbuch ist zwar recht aufschlußreich, der Effekt für die Praxis aber leider noch nicht ganz klar.

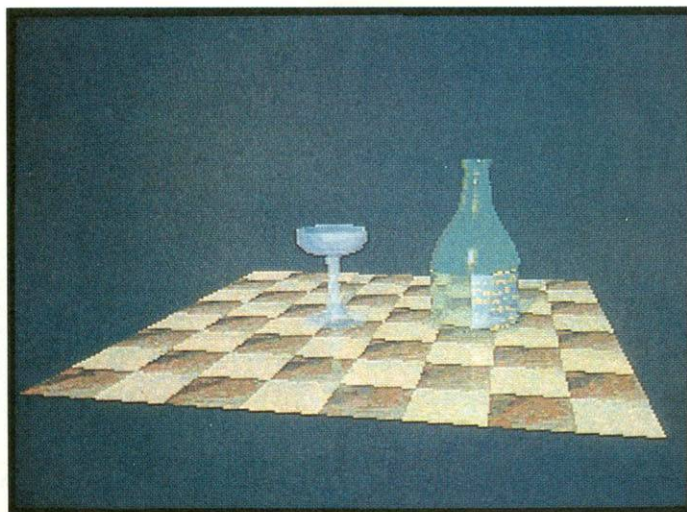


Bild 7. So könnte eine Lösung der Aufgabe des letzten Teils aussehen

Die beiden Funktionen »Rund« und »Eckig« sind uns schon einmal über den Weg gelaufen und benötigen keiner großartigen Erklärung, da sie zum Abrunden von Körpern da sind und sich gegenseitig rückgängig machen. Wenn Sie eine Flasche kreieren, sollten Sie sie abrunden, da sie so echter wirkt. »DefMat« verhält sich ähnlich wie »AddNeu«. Angenommen, unser Beispieltisch hat Beine aus Holz und eine Platte aus Marmor. Die einzelnen Körperbeschreibungen sind bereits mit »K_löschen« gelöscht worden, und der Tisch läßt sich also nur noch über die Bezeichnung »ganzer_Tisch« ansprechen. Plötzlich kommen Sie auf den

Old Grandma's
Edelfusel
gefuzelt, gekellert und
gepanzert 1988
im Weinkeller Sautkopf

dummen Gedanken, daß die Tischbeine doch etwas zu kurz geraten sind, und wollen sie verlängern. Hierbei hilft Ihnen »DefMat«. Bei der Auswahl im Körper-Fenster wählen Sie »ganzer_Tisch« und als Material »Holz«. Wenn »Construct« im angegebenen Körper Objekte aus Holz findet, faßt es diese zusammen und fragt Sie nach einem neuen Namen dafür. Auf die Routine »Umdreh« wollen wir hier nicht näher eingehen, da sie für uns nicht weiter von Bedeutung ist.

Und es ward Licht

Auf die beiden Funktionen »NeuLicht« und »LichtPos« sind wir in der letzten Folge bereits eingegangen. Es bleiben daher nur noch »Lichtfarbe« und »Licht_weg« zu erklären. Bevor Sie eine der beiden Funktionen benutzen, sollten Sie mit den Funktionen des Menüs »Plotten« dafür sorgen, daß Sie die gewünschte Lampe auf dem Bildschirm sehen, da Sie sie durch Anklicken identifizieren müssen. Damit ist »Licht_weg« auch schon fertig, denn die Lichtquelle verschwindet. Bei »Lichtfarbe« verändern Sie die Lichtfarbe genauso, wie Sie auch bei »Neu_Licht« (siehe Teil 2) vorgegangen sind. Diesmal müssen Sie nur eine Lichtquelle vorher anklicken. Mit »Hintergrund« können Sie dem Hintergrund, der ja normalerweise schwarz ist, ein Material verpassen. Wenn es eine Textur enthält, dürfen Sie zwischen einer Kugel- und einer Flächen-Textur wählen. Wie eine Kugel-Textur als Hintergrund aussieht, zeigt Ihnen Bild 3.

Leider war es mit der uns vorliegenden Version von Reflections nicht möglich, mit Hilfe

der Funktion »Umgeb.Hell« ein Umgebungslicht zu simulieren. Bei dem schlichten Versuch begab sich entweder »Beams« oder »Show« sofort in den ewigen Jagdgründen auf Guru-Suche.

Die Maschine mit dem BING

Für all diejenigen unter Ihnen, die im Umgang mit Malprogrammen nicht sehr geschickt sind, haben wir ein paar Tips auf Lager, die es auch ohne künstlerische Ader erlauben, realistische Raytracing-Bilder zu erstellen.

Die meisten Materialien lassen sich am besten über Texturen erstellen. Man kommt dabei nicht umhin, möglichst echte Textur-Muster zu kreieren. Was liegt da näher, als einfach mal mit einer Videokamera und Digitizer auf Materialjagd zu gehen. Dies könnte zwar zur Folge haben, daß man von etwaigen Raytracing-Puristen als »Banause« oder »Ignorant« bezeichnet wird, was aber nichts an der Tatsache ändert,

daß sich mit dieser Methode hervorragende Effekte erzielen lassen, besonders wenn es um Realitätstreue geht. Da vermutlich nicht jeder eine Videokamera und einen entsprechenden Digitizer zur Hand hat, haben wir für Sie einen Streifzug durch Haus und Garten unternommen. Das Ergebnis sehen Sie in den Bildern 1 und 2. Insgesamt haben wir 50 Materialien mit und ohne Textur für Sie zusammengestellt und auf die Databox kopiert, für die Sie im Heft entsprechende Bestellkarten finden. Zum Digitalisieren noch ein paar Tips: Wählen Sie ein Format von 160×128 Punkten bei maximal 32 Farben, denn »Reflections« kann keine Extra-Halfbright- und HAM-Bilder verarbeiten. Die Größe ist sehr gut geeignet, da noch Details zu erkennen sind, sie aber nicht allzuviel Speicher verbrauchen. Oft ist es empfehlenswert, nach dem Digitalisieren die Bilder mit einem Malprogramm nachzubearbeiten. Das heißt, daß man sich aus dem digitalisierten Bild per Brush noch einmal einen Aus-

schnitt herausschneidet. Bei der Glaswand in Bild 2 haben wir das so gemacht, da es nicht möglich war, mit der Kamera genau einen Glasbaustein auf das Bild zu bekommen.

Im letzten Teil haben wir die einzelnen Texturarten etwas vernachlässigt. Vielleicht haben Sie schon einmal ein bißchen herumexperimentiert. Falls nicht, so sehen Sie sich einfach einmal die Bilder 3 bis 6 an. Man hat manchmal das Gefühl, daß »Beams« ab und zu keine Lust auf Texturen hat. Es ist nicht nur einmal passiert, daß ein mit einer Textur belegter Körper behandelt wurde, als wäre er aus »Nichts«. So war hinter den Säulen aus Bild 1 ursprünglich eine Glaswand wie in Bild 2. Ob »Beams« der Meinung war, sie würde stilistisch nicht passen? Warum sie zwar in der Szene korrekt vorhanden ist, aber im Bild nicht, weiß wohl nur der Guru.

Noch größere Probleme können mit Etikett-Texturen auftreten. Da kam es vor, daß die Flasche aus Bild 4 nur aus grünem Glas ohne Etikett bestand oder einmal auch aus nichts. Es ist daher immer sinnvoll, vor der Berechnung eines Bildes in Low-Res erst ein Testbild zu berechnen. Selbst wenn dies eine Stunde dauern sollte, lohnt es sich. Da es nicht möglich war, herauszufinden, wo die Schnecke im Hafer liegt, können wir Ihnen nur den Tip geben, daß Sie bei einer speziellen Textur einfach eine Flächentextur wählen, falls Sie bei Rotationskörper- oder Kugeltextrur nicht den gewünschten Erfolg haben. Auf diese Weise konnte man bisher (fast) immer eine Textur vergeben.

Bevor wir in aller Ausführlichkeit besprechen, wie man die Aufgabe aus dem letzten Teil lösen kann, möchte ich noch

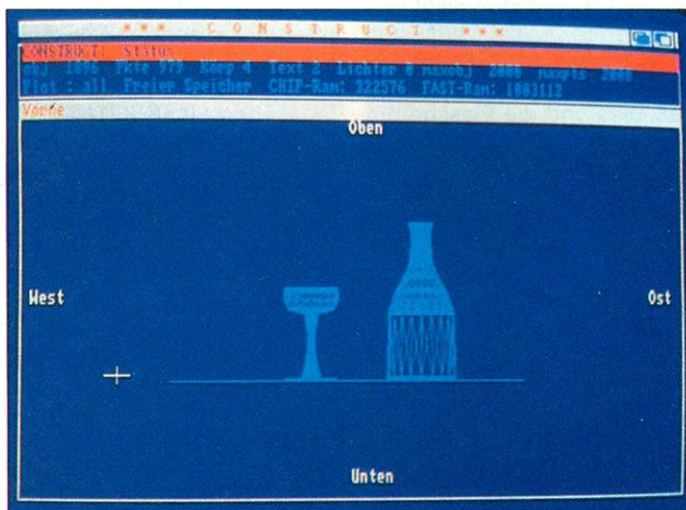


Bild 8. In »Construct« sieht die Szene aus Bild 7 von vorne gesehen so aus

ein paar Worte über die Vergabe von Texturen verlieren. Sie sollten nämlich wissen, was Sie eigentlich bei der Aufforderung »Verschieben Sie das Rechteck« machen.

Auflösung der Aufgabe

Haben Sie eine Flächentextur gewählt, so können Sie sich vorstellen, die Textur würde als Dia auf den Körper projiziert. Das Rechteck legt die Größe des Bildes fest. Ist es kleiner als der Körper, wird er in alle Richtungen nahtlos wiederholt, bis der Körper vollständig ausgefüllt ist. Ganz ähnlich ist es bei Rotationskörper- und Kugelt Texturen. Allerdings werden die Körper hier in die Textur »eingewickelt«. Sie müssen nun einstellen, wieviel Grad des Körpers die Textur bedecken soll. Falls sie weniger bedeckt als 360°, müssen Sie unterhalb der Grad-Skala angeben, von wo bis wo die Textur den Körper bedecken soll. Das geschieht einfach durch entsprechendes Platzieren des Rechtecks. »Hoch« und »runter« legt die Position vertikal fest, rechts und links verschieben die Start- und Endposition entsprechend der Skala am oberen Rand. 0 beziehungsweise 360° ist, von vorne betrachtet, mitten auf dem Körper. Jetzt dürften Ihnen Texturen eigentlich keine Probleme mehr bereiten.

Die Schachbrettszene ist eigentlich nicht schwer zu kreieren, wenn man sich mit Texturen auskennt. Als erstes müssen Sie mit einem Malprogramm (DPaint oder ähnliches) ein Etikett zeichnen, das Sie dann als Brush im Ordner »reflections/text« unter dem Namen »Etikett« abspeichern. Es könnte ungefähr so aussehen wie Bild 8. Rufen Sie »Construct« auf, und erstellen Sie einen Quader. Mit »links« stellen Sie die Blickrichtung ein. Jetzt müssen Sie den Körper verzerren, da der Quader viel zu dick ist. Das macht die Funktion »Verändern« für uns. Die Höhe bleibt, die Breite wird soweit verändert, daß aus dem Quader ein Brett wird, das wir dann mit »Drehen« um 90° drehen. Das ist das Schachbrett. Das Glas und die Flasche sind Rotationskörper mit je 16 Winkelschritten. Die Gestaltung

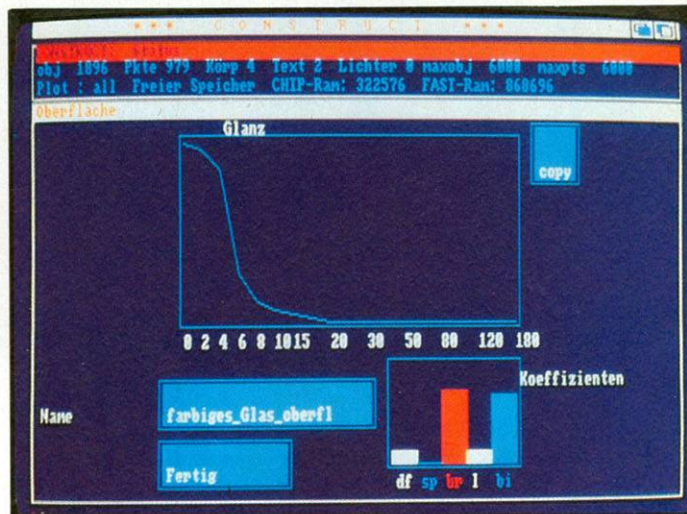


Bild 9. Diese Einstellung müssen Sie vornehmen, um die Oberfläche »farbiges_Glas_oberfl« zu erhalten

bleibt Ihrem Geschmack oder auch Ihrer Vorstellung von diesen Körpern überlassen. Vergessen Sie nur den Boden nicht, denn wenn Sie »Rotkörper« aufrufen, müssen Sie den ersten Punkt auf die Rotationsachse legen. Wiederholen Sie den Vorgang so lange, bis das Ergebnis Ihren Vorstellungen entspricht. Danach runden Sie die beiden Körper mit »Rund«. Lassen Sie sich nun die Szene von vorne plotten. Wie Sie sehen, liegen die beiden Rotationskörper noch immer. Also müssen wir sie um den Mittelpunkt drehen. Dabei müssen Sie diesmal um 270° drehen, also muß der Mauszeiger unterhalb der 0°-Linie auf -90 positioniert werden. Danach bringen wir beide Körper auf die richtige Größe. Das geschieht am besten mit »Verändern« alias »Vergrößern«, damit die Proportionen nicht verlorengehen. Als nächstes schauen Sie sich die Szene von oben an und verschieben das Glas und die Flasche mit »Verändern« und »Vergrößern«, ohne allerdings dabei die linke Maustaste zu drücken. Prinzipiell können Sie auch »Verzerren« nehmen, sollten Sie jedoch aus Versehen auf die Maustaste drücken, können Sie eine Vergrößerung einfach rückgängig machen als eine Verzerrung. Wenn Sie hier angelangt sind, müßte Ihre Szene von vorne so aussehen wie Bild 9. Laden Sie mit »LoadMat« die Materialdatei. Vergeben Sie mit »Material« das Material »Schach« an das Schachbrett. Dabei ist es wichtig,

daß Sie immer noch von oben auf die Szene sehen, denn Sie wollen ja das Schachbrett nicht auf den Rand des Brettes projizieren. Die gewünschte Textur ist eine Flächentextur. Deshalb brauchen Sie die Größe nicht mehr zu verändern, sondern nur noch die rechte Maustaste zu drücken. Da wir keinen Hintergrund haben, ist es vorteilhaft, für das Glas »mattes_Glas« zu nehmen. Sonst könnte es passieren, daß Sie je nach Kameraeinstellung das Glas gar nicht erkennen können. Wie Sie »mattes_Glas« erstellen, finden Sie im letzten Kurs. So, es geht ans Eingemachte: Wir erstellen das Material »grünes_Glas+Etikett«. Wenn Sie anfangs das Etikett bereits geschaffen haben und es sich im bereits Verzeichnis »reflections/text« befindet, können wir gleich mit »NeuTex« loslegen, ansonsten müssen Sie die oben aufgeführten Schritte nachholen. In »NeuTex« geben wir als IFF-Namen »Etikett« an. Genauso nennen wir auch die Textur. Klicken Sie »Mat-Tab« an, und wählen Sie als erstes Material irgendein undurchsichtiges Material. Es spielt keine Rolle, welches Sie nehmen, da die Farbe sowieso von der Textur überschrieben wird. Vergessen Sie nicht, den Schalter »Etik« einzuschalten, bevor Sie »Fertig« anklicken und die Textur sichern. Für das farbige Glas brauchen wir auch noch eine neue Oberfläche. Wählen Sie »NeuOb«, und stellen Sie die Werte folgendermaßen ein: Die Kurve des

Glanzlichtes besorgen Sie sich per »copy« von »Glas_oberfl«. »df« sollte ungefähr 7 Punkte hoch sein, »sp« 0, »br« 34, »l« 7 und »bi« 32 (Bild 10). Nennen Sie die neue Oberfläche einfach »farbiges_Glas_oberfl«. Mit »NeuMat« müssen Sie jetzt nur noch die Textur »Etikett« und die Oberfläche »farbiges_Glas_oberfl« zum Material »grünes_Glas+Etikett« vereinigen. Als Farbe stellen Sie am besten ein zartes Dunkelgrün ein. Legen Sie nun die neue Textur von vorne auf die Flasche. Dazu müssen Sie »Material« aufrufen und erst »Flasche« und danach »grünes_Glas+Etikett« anklicken. Es handelt sich hier nicht um eine Flächentextur, sondern um eine Rotationskörper texture. Verändern Sie das Rechteck so, daß es 180 »breit« ist und sich von 0° bis 360° erstreckt. Die Höhe sollten Sie so einstellen, daß das Etikett nicht bis in den Flaschenhals hineinragt. Bleiben Sie mit Ihrer Textur auf jeden Fall in dem durch Körperhöhe und Skala festgelegten Bereich, sonst kann es zu Komplikationen kommen.

Bevor wir Kamerakind spielen, sollten wir das Licht einschalten. Das passiert mit »NeuLicht«. Die Farbe stellen wir auf »Schach« ein und die Lichtstärke auf etwa 1200. Bei der Kameraposition haben Sie freie Hand. Das einzige, worauf Sie achten sollten ist, daß die Anwahl von »Kamera« jede Einstellung aus »Kam.beweg« rückgängig macht. Wenn man das Licht neben der Kamera positionieren will, was sich meistens empfiehlt, muß man eventuell mit der Funktion »Bild*2« dem Bildausschnitt solange zu Leibe rücken, bis man sowohl das Kreuz, das die Kamera repräsentiert, als auch die Kreise der Lampen sieht. Dann verschiebt man die Lampen einmal aus der Sicht von oben und dann aus der Sicht von vorne mit »Lichtpos«. Nun können Sie die Szene abspeichern. Vergessen Sie nicht, mit »Save_Kam/Li« auch die Kameraposition und die Lichtverhältnisse abzuspeichern und mit »SaveMat« die zwei neuen Materialien zu retten. Als nächstes starten Sie »Beams«... Das Ergebnis spricht für sich.

(mm)

Wenden wir uns als erstes der normalen Bildschirmaufteilung zu, wie wir sie ja beim Amiga gewohnt sind (wir setzen hierfür einfach mal eine Low-Res-Auflösung mit 256 Zeilen voraus): Da haben wir zunächst links oben in der Ecke den Punkt mit den Koordinaten 0/0. In der rechten oberen Ecke finden wir den Punkt 319/0, in der linken unteren Ecke 0/255 und rechts unten letztendlich den Punkt mit den Koordinaten 319/255 (Bild 1).

Wenn wir also einen Punkt in der Mitte des Bildschirms positionieren wollen, so geben wir die Koordinaten 160/128 an (Bild 1). Sie sehen, daß dieser Punkt einen ganz bestimmten Abstand von der linken oberen Ecke hat. Und nicht nur das, er liegt sogar noch in einem ganz bestimmten Winkel zur X- beziehungsweise Y-Achse. Man kann diesen Punkt also nicht nur dadurch genau bestimmen, daß man seine Koordinaten angibt, sondern auch indem man die Entfernung von der linken oberen Ecke und den Winkel zur X-Achse angibt. Dies ist nämlich die sogenannte Vektorschreibweise. Um nun unsere 3D-Stars zu programmieren, ist es aber nötig, das Koordinatenkreuz ein wenig zurechtzurücken, und zwar so, daß unsere Bildschirmmitte der Punkt ist, an dem sich X- und Y-Koordinate schneiden (Bild 2). Wenn wir jetzt also einen Punkt in unserer Bildschirmmitte zeichnen wollen, so müßten wir als Koordinaten 0/0 angeben. Um diesen Punkt jetzt zum Beispiel mit der Routine »WritePixel«

Torsten Budesheim

Stars and Bytes

Manche Demos begeistern durch vorbeifliegende, manchmal sogar rotierende Sterne. Doch wie programmiert man sie?

auf den Bildschirm zu bringen, müssen wir die Koordinaten aber erst wieder umrechnen. Dies ist allerdings recht einfach; indem wir nämlich zur X-Koordinate die Breite des Bildschirms, geteilt durch zwei, und zur Y-Koordinate die Höhe des Bildschirms, geteilt durch zwei, hinzuaddieren. In unserem Fall wären dies

$$X = 0 + 320 / 2 = 160$$

$$Y = 0 + 256 / 2 = 128$$

Nehmen wir nun weiter an, wir hätten einen Punkt, dessen Vektor 20 und dessen Winkel 90 Grad sei. Aber wie rechnen wir nun diese Werte in Koordinaten für unser Koordinatensystem um?

Nun, dafür gibt es in der Mathematik zwei kleine Formeln, die ich hier nicht weiter ableiten möchte, aber die wir eben für unser Programm brauchen, und zwar errechnet sich dabei der X-Wert mit der Formel:

$$X = \cos(\text{Winkel}) * \text{Vektor}$$

Und der Y-Wert durch:

$$Y = \sin(\text{Winkel}) * \text{Vektor}$$

Für unser Beispiel erhalten wir hierbei die Werte $X = \cos(90) * 20 = 0$ und $Y = \sin(90) * 20 = 20$. Dies sind

wohlgerneht die Koordinaten für unser zurechtgerücktes Koordinatensystem! Um die Punkte also auf den Bildschirm zu bringen, müssen wir diese erst, wie weiter oben beschrieben, umrechnen. So, dies wären schon mal die grundlegenden Kenntnisse aus der Mathematik. Ich möchte nun noch ein wenig auf die programmtechnischen Probleme eingehen, die hierbei auf uns zukommen.

Da wäre zunächst einmal die Frage danach, wie wir die Sterne nun erstellen und bewegen, und vor allem, wie wir dieses nette Drehen der Sterne hinbekommen. Diese Fragen sind so schnell erklärt, wie sie gestellt sind:

1. Um einen Stern zu erhalten, rufen wir zweimal eine Zufallsroutine auf, einmal für den Vektor und zum anderen für den Winkel. Wir müssen hierbei nur darauf achten, daß der Winkel dann nicht größer als 359 Grad ist und zum anderen der Vektor möglichst klein ist (sonst bekommen wir in der Mitte ein "schwarzes Loch").
2. Um einen Stern zu bewegen, müssen wir nur den Vek-

tor mit einem bestimmtem Wert multiplizieren.

3. Um einen Stern zu drehen, erhöhen oder verringern wir einfach seinen Winkel, und schon erhalten wir eine Drehung. Auch hier kommt wieder ein einfacher, aber wirkungsvoller Trick zur Geltung: Wir berechnen die Werte nicht jedesmal neu, sondern holen uns diese Werte aus einer Tabelle. Diese Tabelle wird gleich am Anfang des Programms generiert. Und damit wir nicht mit langsamen Kommazahlen rechnen müssen, haben wir die Werte, bevor wir sie in die Tabelle eintragen, noch um 15 Bits nach links "geschiftet" (mit 215 multipliziert). Sind wir dann mit einer Berechnung fertig, so müssen wir das Ergebnis wieder nur noch durch 215 teilen, und schon stimmt alles wieder. Da dies auch erledigt wäre, wenden wir uns dem Programm zu.

Das Programm besteht im Prinzip aus zwei Teilen: Zum einen die eigentlichen, zum Erzeugen der Sterne nötigen Daten und zum anderen der Programmteil, der den Screen öffnet und die entsprechenden Daten an die Megastars-Routine übergibt. Am Anfang des Sources stehen die nötigen Label-Definitionen. Danach beginnt das Programm. Zuerst werden die Initialisierungen vorgenommen: Initialisieren der Sinus-Tabelle, Öffnen der »intuition.library«, Öffnen des Screens und so weiter und Initialisierung der »MegaStars«. Danach wird getestet, ob eventuell beide Maustasten gedrückt worden sind, und wenn dies der Fall ist, alle Initialisierungen rückgängig ge-

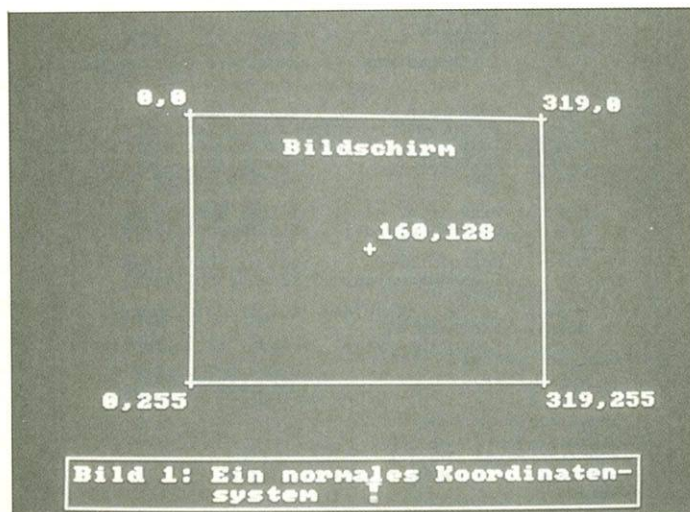


Bild 1. Das normale Koordinatensystem

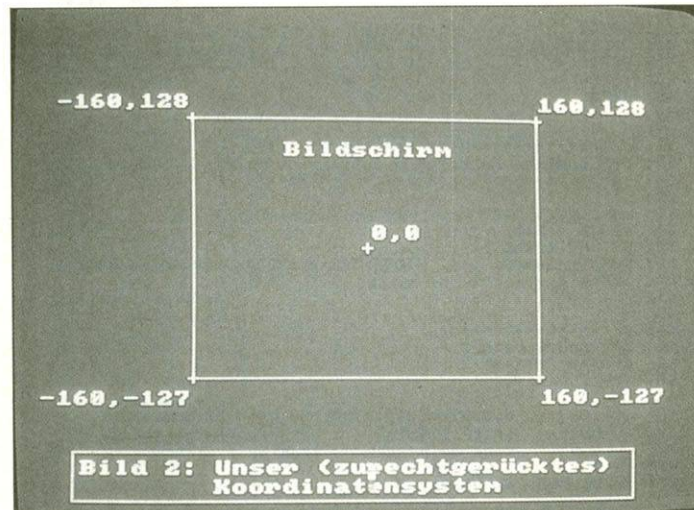


Bild 2. Das "zurechtgerückte" Koordinatensystem

macht und das Programm verlassen. Ansonsten werden die Sterne immer weiter bewegt, bis eben beide Maustasten gedrückt wurden. Es folgen einige Speicherreservierungen und nötige Strukturen, bevor die eigentlichen Routinen beginnen. Als erstes haben wir da zwei Routinen zum Setzen beziehungsweise Löschen von Punkten. Ich habe hier nicht die Routine »WritePixel« verwendet, da diese viel zu langsam ist. Die Routinen »preset« und »pset« setzen einen Punkt jeweils nur in einer »Plane«. Sollten Sie also einen Punkt in mehreren Ebenen setzen wol-

len, so müssen Sie die Routine mehrmals aufrufen. Ich möchte noch auf etwas hinweisen, was zugleich auch eine kleine Programmieraufgabe für Sie darstellt: Die Routine »Randomize« stammt aus einem Public-Domain-Programm und ist nicht besonders gut. Versuchen Sie, diese durch eine schnellere und effektivere zu ersetzen. Außerdem sind einige Passagen des Programms (noch) nicht besonders schnell. Ich habe diese bewußt nicht optimiert, damit Sie diese durch schnellere ersetzen können. Noch einen kleinen Tip: Die Routine »new_vec« er-

stellt einen neuen Vektor beziehungsweise einen neuen Winkel. Dazu wird die Zufallsroutine angesprochen, und überflüssige Bits werden entsprechend ausmaskiert. Probieren Sie hier doch mal zum Beispiel bei den Shift- beziehungsweise And-Befehlen (aber nur für den Vektor) andere Werte aus!

Die »MegaStars« sind so programmiert, daß sie unter beliebigen Auflösungen laufen. Sie können dies auch ausprobieren, indem Sie die entsprechenden Werte (»WIDTH« = Breite, »HEIGHT« = Höhe, »MODE« = 8000 bei High-Res

und »MODE« = 2 bei Low-Res) am Anfang des Listings ändern. Als weitere veränderbare Konstanten stehen hier noch »maxStars«, welche die maximale Zahl der vom Programm zu verwendenden Sterne darstellt, und »INC«, die den Wert enthält, der bei eventuell gedrückter rechter beziehungsweise linker Maustaste zum Winkel addiert oder subtrahiert wird. Noch eines sollte ich vielleicht erwähnen: Der Source läuft unter dem Seka-Assembler. Eventuelle Anpassungen an andere Assembler müssen Sie noch vornehmen.

(jb)

Listings



```

1: ;*****
2: ;* MegaStars V1.0 - geschrieben 1989 von *
3: ;* Torsten Budesheim *
4: ;*****
5:
6:
7:
8: ;*****
9: ;* Exec-Offsets definieren *
10: ;*****
11:
12: ExecBase      = 4
13: OpenLibrary    = -30 -522
14: CloseLibrary   = -30 -384
15:
16:
17: ;*****
18: ;* Intuition-Offsets definieren *
19: ;*****
20:
21: OpenScreen     = -30 -168
22: CloseScreen    = -30 -36
23: OpenWindow     = -30 -174
24: CloseWindow    = -30 -42
25:
26:
27: ;*****
28: ;* Graphics-Offsets definieren *
29: ;*****
30:
31: LoadRGB4       = -30 -162
32:
33:
34: ;*****
35: ;* MathTrans-Offsets definieren *
36: ;*****
37:
38: SPSin           = -30 -6
39:
40:
41: ;*****
42: ;* MathFFP-Offsets definieren *
43: ;*****
44:
45: SPMul           = -30 -48
46: SPDIV           = -30 -54
47: SPFix           = -30 -0
48: SPFlt           = -30 -6
49: SPTst           = -30 -18
50:
51:
52: ;*****
53: ;* Sonstige Konstanten definieren *
54: ;*****
55:
56: maxStars = 100
57: WIDTH = 640
58: HEIGHT = 256
59: MODE = $8000 ;if WIDTH>320 => MODE=$8000 else MODE=2
60: INC = 2
61:
62:
63: ;*****
64: ;* Programmstart *
65: ;*****
66:
67: run:
68:   lea    SinusTable(pc),a4 ;Zeiger auf SinTable
69:   bsr    initSinTable      ;Sinuswerte eintragen
70:   beq    back_to_system    ;Fehler ? ja: ->
71:
72:   move.l ExecBase,a6

```

Listing: Megastars



```

73:   lea    IntuitionName,a1 ;Zeiger auf Library-Name
74:   move.l #31,d0            ;Version
75:   jsr    OpenLibrary(a6)   ;Library oeffnen
76:   move.l d0,IntuitionBase ;Zeiger retten
77:   beq    back_to_system    ;bei Fehler beenden
78:
79:   lea    Screen,a0         ;Zeiger auf Screen-Defini
80:   tionen
81:   move.l d0,a6             ;Zeiger auf Intuition-Base
82:   jsr    OpenScreen(a6)    ;Screen oeffnen
83:   move.l d0,ScrPointer     ;Zeiger auf Screen retten
84:   beq    close_IntuiBase   ;bei Fehler abschliessen
85:
86:   move.l d0,-(sp)          ;Zeiger auf Screen merken
87:   move.l ExecBase,a6
88:   lea    GfxName,a1        ;Zeiger auf Library-Name
89:   move.l #31,d0            ;Version
90:   jsr    OpenLibrary(a6)   ;Library oeffnen
91:   move.l d0,a6
92:   beq.s  open_win          ;bei Fehler abschliessen
93:   move.l (sp)+,d0          ;Zeiger auf Screen holen
94:   move.l d0,a0
95:   lea    44(a0),a0         ;Zeiger auf ViewPort holen
96:   lea    myColors,a1       ;Zeiger auf eigene Farbpal
97:   ette
98:   moveq  #4,d0             ;Anzahl der Farben: 4
99:   jsr    LoadRGB4(a6)     ;Farben initialisieren
100:  move.l a6,a1             ;Zeiger auf Library
101:  jsr    CloseLibrary(a6)   ;Gfx-Library wieder schlie
102:  ssen
103:  move.l IntuitionBase,a6
104:  open win:
105:  lea    Window,a0         ;Zeiger auf Window-Definit
106:  ionen
107:  jsr    OpenWindow(a6)     ;Window oeffnen
108:  move.l d0,WinPointer     ;Zeiger retten
109:  beq.s  close_scr         ;bei Fehler Programm beenden
110:
111:  move.l d0,a0
112:  move.l 50(a0),a0         ;Zeiger auf RastPort
113:  move.l 4(a0),a0          ;Zeiger auf BitMap
114:  move.l 12(a0),a1         ;Zeiger auf Plane2
115:  moveq  #0,d0
116:  moveq  #0,d1
117:  move.w 0(a0),d0          ;BytesPerRow
118:  move.w 2(a0),d1          ;Height
119:  move.l 8(a0),a0          ;Zeiger auf Plane1
120:  mulu   #8,d0             ;Width = BytesPerRow*8
121:  bsr    InitMegaStars      ;MegaStars-Routine initial
122:  isieren
123:  loop:
124:  bsr    MegaStars          ;Sterne bewegen
125:  btst   #6,$bfe001         ;Hier wird getestet, ob beide
126:  bne.s  loop               ;Maustasten gedrueckt sind und
127:  btst   #2,$dff016         ;wenn nicht, die Sterne
128:  bne.s  loop               ;einfach weiter bewegt
129:
130:  move.l WinPointer,a0     ;Zeiger auf Window
131:  jsr    CloseWindow(a6)   ;Window schliessen
132:  close_scr:
133:  move.l ScrPointer,a0     ;Zeiger auf Screen
134:  jsr    CloseScreen(a6)   ;Screen schliessen
135:  close_IntuiBase:
136:  move.l IntuitionBase,a1 ;Zeiger auf Library
137:  move.l ExecBase,a6
138:  jsr    CloseLibrary(a6) ;close 'intuition.library'
139:  back_to_system:
140:  moveq  #0,d0             ;kein Fehlercode
141:  rts                     ;Programm verlassen
142:
143:  even
144:  WinPointer: dc.l 0
145:  IntuitionBase: dc.l 0
146:  myColors: dc.w 0,$333,$999,$666
147:  Screen:

```

Listing: Megastars


```

144: dc.w 0,0,WIDTH,HEIGHT,2
145: dc.b 0,1
146: dc.w MODE,15
147: dc.l 0,Titel,0,0
148: Window:
149: dc.w 0,0,WIDTH,HEIGHT
150: dc.b 1,3
151: dc.l $200,$11800,0,0,0
152: ScrPointer:
153: dc.l 0,0
154: dc.w 0,0,0,0,15
155:
156: Titel: dc.b 'MegaStars 1989 by MovingSigns',0
157: IntuitionName:
158: dc.b 'intuition.library',0
159: GfxName:
160: dc.b 'graphics.library',0
161: even
162:
163:
164:
165: ;*****
166: ;-----
167: ;*****
168:
169: ;*****
170: ;*** Loesche Pixel ***
171: ;***
172: ;*** A3: Zeiger auf BitPlane ***
173: ;*** D0: X-Koordinate ***
174: ;*** D1: Y-Koordinate ***
175: ;*** D2: Groesse der ***
176: ;*** BitPlane in Bytes ***
177: ;*****
178: preset:
179: movem.l d0-d3,-(sp)
180: muls d2,d1
181: move.w d0,d3
182: ext.l d0
183: asr.l #3,d0
184: add.l d0,d1
185: moveq #5f,d0
186: and.w d0,d3
187: sub.w d3,d0
188: bclr d0,0(a3,d1.l)
189: movem.l (sp)+,d0-d3
190: rts
191:
192: ;*****
193: ;*** Setze Pixel ***
194: ;***
195: ;*** A3: Zeiger auf BitPlane ***
196: ;*** D0: X-Koordinate ***
197: ;*** D1: Y-Koordinate ***
198: ;*** D2: Groesse der ***
199: ;*** BitPlane in Bytes ***
200: ;*****
201: pset:
202: movem.l d0-d3,-(sp)
203: muls d2,d1
204: move.w d0,d3
205: ext.l d0
206: asr.l #3,d0
207: add.l d0,d1
208: moveq #5f,d0
209: and.w d0,d3
210: sub.w d3,d0
211: bset d0,0(a3,d1.l)
212: movem.l (sp)+,d0-d3
213: rts
214:
215: ;*****
216: ;*** Mega-Stars ***
217: ;***
218: ;*** Written 03/25/89 by Torsten Budesheim ***
219: ;*** Copyright 1989 by MovingSigns ***
220: ;*****
221: ;Achtung!:
222: ;- Mega-Stars benutzt die Routinen 'pset' und 'preset'
223: ;- Bevor man 'Mega-Stars' benutzen kann, muss die
224: ; Routine 'InitMegaStars' aufgerufen werden
225:
226: MegaStars:
227: movem.l d4-d6/a0/a1,-(sp)
228:
229: lea Sinustable,a0 ;Zeiger auf Sin-/Cos-Tabelle
230: lea StarTable,a1 ;hier werden Vektor und
231: ;Winkel gespeichert
232: moveq #0,d6 ;Zaehler fuer die Sterne
233: stars_moving:
234: moveq #0,d4
235: moveq #0,d5
236: move.w (a1)+,d4 ;Vektor holen
237: move.w (a1)+,d5 ;Winkel holen
238: lea preset,a2
239: bsr doStar ;Stern loeschen
240:
241: mulu #11,d4 ;Vektor = Vektor * 1.1
242: divu #10,d4 ;
243: bsr.s TestLMB ;Sterne drehen ?
244: bsr.s TestRMB ;
245: bsr.s TestStar ;Stern ausser Reichweite ?
246:
247: move.w d4,-4(a1) ;neuen Vektor speichern
248: move.w d5,-2(a1) ;(neuen) Winkel speichern
249: lea pset,a2
250: bsr doStar ;Stern zeichnen
251:

```

Listing: Megastars

```

252: addq.l #4,d6
253: cmp.l #maxStars*4,d6 ;schon alle Sterne bewegt ?
254: bne.s stars_moving ;nein ? dann von vorne
255:
256: movem.l (sp)+,d4-d6/a0/a1
257: rts ;zurueck
258:
259: ;--- Testet, ob X/Y-Koordinate ---
260: ;--- des Sterns schon zu gross ---
261: ;d4 - Vektor
262: ;d5 - Winkel
263: ;a0 - Zeiger auf Sinustabelle
264: ;-----
265: TestStar:
266: movem.l d0-d2,-(sp)
267:
268: bsr GetXY ;Stern umrechnen in Koordinaten
269: move.w x_max,d2 ;maximale X-Koordinate
270: cmp.w d0,d2 ;X-Koordinate zu gross ?
271: bhl.s x_ok ;nein ? dann weiter
272: bsr.s new_vec ;sonst neuen Vektor und Winkel
273: holen
274: bra.s y_ok
275: x_ok:
276: move.w y_max,d2 ;maximale Y-Koordinate
277: cmp.w d1,d2 ;Y-Koordinate zu gross ?
278: bhl.s y2_ok ;nein ? dann weiter
279: bsr.s new_vec ;sonst neuen Vektor und Winkel
280: holen
281: bra.s y_ok
282: y2_ok:
283: move.w x_min,d2 ;minimale X-Koordinate
284: cmp.w d0,d2 ;X-Koordinate zu klein ?
285: bhl.s x2_ok ;nein ? dann weiter
286: bsr.s new_vec ;sonst neuen Vektor und Winkel
287: holen
288: bra.s y_ok
289: x2_ok:
290: move.w y_min,d2 ;minimale Y-Koordinate
291: cmp.w d1,d2 ;Y-Koordinate zu klein ?
292: bhl.s y_ok ;nein ? dann weiter
293: bsr.s new_vec ;sonst neuen Vektor und Winkel
294: holen
295: y_ok:
296: movem.l (sp)+,d0-d2
297: rts
298:
299: ;- Testet, ob rechte
300: ;- Maustaste gedrueckt ist
301: ;- (wenn ja: dec Winkel)
302: ;-
303: ;- d5 - Winkel
304: ;-----
305: TestRMB:
306: btst #2,$dff016 ;rechte Maustaste gedrueckt ?
307: bne.s end_RMB ;nein ? dann zurueck
308: subq.w #1,INC,d5 ;Winkel verringern
309: bpl.s end_RMB ;Winkel schon zu klein ?
310: add.w #360,d5 ;Winkel korrigieren
311: end_RMB:
312: rts
313:
314: ;- Testet, ob linke
315: ;- Maustaste gedrueckt ist
316: ;- (wenn ja: inc Winkel)
317: ;-
318: ;- d5 - Winkel
319: ;-----
320: TestLMB:
321: btst #6,$bfe001 ;linke Maustaste gedrueckt ?
322: bne.s end_LMB ;nein ? dann zurueck
323: addq.w #1,INC,d5 ;Winkel erhoehen
324: cmp.w #359,d5 ;Winkel schon zu gross ?
325: bhl.s end_LMB ;wenn nein: zurueck
326: sub.w #360,d5 ;sonst Winkel korrigieren
327: end_LMB:
328: rts
329:
330: ;- neuen Winkel und Vektor erstellen
331: ;- Rueckgabe:
332: ;- d4 - neuer Vektor
333: ;- d5 - neuer Winkel
334: ;-----
335: new_vec:
336: move.l d0,-(sp)
337: bsr.s Randomize ;Zufallszahl holen
338: and.l #5f,d0 ;maximaler Vektor = 16
339: bset #1,d0
340: lsl.l #2,d0
341: bset #1,d0
342: move.l d0,d4
343: bsr.s Randomize ;Zufallszahl holen
344: and.l #1fff,d0 ;max Winkel = 359
345: cmp.w #359,d0 ;Winkel zu gross ?
346: bhl.s new_angle_ok ;nein : dann ueberspringen
347: sub.w #360,d0 ;sonst Winkel korrigieren
348: new_angle_ok:
349: move.l d0,d5
350: move.l (sp)+,d0
351: rts
352:
353: ;--- Randomize ---
354: ;Rueckgabe
355: ;d0 - Zufallszahl
356: ;-----
357: Randomize:
358: movem.l d1-d3/a1,-(sp) ;diese Zufallsroutine stammt
359: lea rndsav,a1 ;aus dem Public-Domain-Programm
360: move.w #54321,d0 ;'Stars.asm'

```

Listing: Megastars

TIPS & TRICKS

```

357:      mulu      (a1),d0      ;Sie ist nicht besonders schnell
358:      addq.l    #1,d0        ;und auch nicht besonders gut und
359:      move.w    d0,(a1)      ;dient hier nur als Demonstration
360:      move.b     (a1),d1      ;bzw. anregung.
361:      eor.b     d0,d1
362:      move.b     d1,rndlow
363:      move.w     $dff006,d3
364:      eor.w     d3,d0
365:      movem.l    (sp)+,d1-d3/a1
366:      rts
367:
368: ;--- Stern zeichnen/loeschen ---
369: ;d4 - Vektor
370: ;d5 - Winkel
371: ;a0 - Zeiger auf Sinustabelle
372: ;a2 - Zeiger auf preset/pset
373: ;-----
374: doStar:
375:      movem.l    d0-d3/a3,-(sp)
376:
377:      bsr.s      GetXY        ;Vektor/Winkel => X/Y
378:      move.l     BytesPerRow,d2 ;Bildschirmbreite holen
379:
380:      cmp.w      #30,d4        ;Hier wird unterschieden, wie
381:      bhl.s      vec_cmp_2     ;gross der Vektor ist und
382:      move.l     Starplane1,a3 ;dementsprechend seine Farbe
383:      jsr        (a2)          ;gesetzt
384:      bra.s      vec_cmp_end    ;
385: vec_cmp_2:
386:      cmp.w      #90,d4        ;
387:      bhl.s      vec_cmp_3     ;
388:      move.l     Starplane1,a3 ;
389:      jsr        (a2)          ;
390:      move.l     Starplane2,a3 ;
391:      jsr        (a2)          ;
392:      bra.s      vec_cmp_end    ;
393: vec_cmp_3:
394:      move.l     Starplane2,a3 ;
395:      jsr        (a2)          ;
396: vec_cmp_end:
397:
398:      movem.l    (sp)+,d0-d3/a3
399:      rts
400:
401: ;-- Wandle Vektor in X/Y-Koordinate um --
402: ;d4 - Vektor
403: ;d5 - Winkel
404: ;a0 - Zeiger auf Sinustabelle
405: ;Rueckgabe:
406: ;d0 - X-Koordinate
407: ;d1 - Y-Koordinate
408: ;-----
409: GetXY:
410:      moveq      #0,d0
411:      moveq      #0,d1
412:      lsr.l      #3,d1
413:      move.l     d5,-(sp)      ;Winkel sichern
414:      asl.w      #1,d5         ;Winkel-Offset als Word
415:      move.w     0(a0,d5.w),d1 ;y=sin(Winkel)
416:      muls      d4,d1          ;
417:      lsr.l      #8,d1         ; *Vektor
418:      lsr.l      #7,d1
419:      move.l     (sp),d5       ;Winkel zurueckholen
420:      addi.w     #90,d5        ;Cos-Offset
421:      cmp.w      #359,d5       ;Winkel groesser als 359 ?
422:      bls.s      angle_ok      ;nein: dann ueberspringen
423:      sub.w      #360,d5       ;Winkel korrigieren
424: angle_ok:
425:      asl.w      #1,d5         ;Winkel-Offset als Word
426:      move.w     0(a0,d5.w),d0 ;X=Cos(Winkel)
427:      muls      d4,d0          ;
428:      lsr.l      #8,d0         ; *Vektor
429:      lsr.l      #7,d0
430:      move.l     (sp)+,d5
431:      and.l      $ffff,d0
432:      and.l      $ffff,d1
433:      add.w      x_middle,d0    ;Koordinaten in in Amiga-
434:      add.w      y_middle,d1    ;Koordinaten umrechnen
435:      rts
436:
437: ;-- 'MegaStars' initialisieren --
438: ;a0 - Zeiger auf BitPlane 1
439: ;a1 - Zeiger auf BitPlane 2
440: ;d0 - BitPlane-Breite
441: ;d1 - BitPlane-Hoehe
442: ;-----
443: InitMegaStars:
444:      movem.l    d2-d5/a0/a1/a4,-(sp)
445:      move.l     a0,Starplane1 ;Zeiger auf BitPlane 1 sichern
446:      move.l     a1,Starplane2 ;Zeiger auf BitPlane 2 sichern
447:      movem.l    d0/d1,-(sp)
448:      subq.l     #3,d0          ;Rand lassen
449:      subq.l     #3,d1
450:      move.w     d0,x_max       ;maximale X-Koordinate speichern
451:      move.w     d1,y_max       ;maximale Y-Koordinate speichern
452:      movem.l    (sp)+,d0/d1
453:      move.l     d0,d2          ;Breite umrechnen in
454:      divu      #8,d0           ;Zeichen pro Zeile
455:      move.l     d0,BytesPerRow ;und merken
456:      divu      #2,d2           ;Mittelpunkt X berechnen
457:      divu      #2,d1           ;Mittelpunkt Y berechnen
458:      move.w     d2,x_middle    ;Mittelpunkte speichern
459:      move.w     d1,y_middle
460:
461:      move.l     #maxStars-1,d3 ;maximale Anzahl der Sterne
462:      lea        Startable,a4   ;Zeiger auf Vektor/Winkel-Spe
463:      lea        Sinustable,a0  ;Zeiger auf Sinustabelle

```

Listing: Megastars

```

464: install_stars:
465:      bsr        new_vec       ;Vektor und Winkel erstellen
466:      move.w     d4,(a4)+      ;und sichern
467:      move.w     d5,(a4)+
468:      lea        pset,a2       ;und entsprechenden Punkt
469:      bsr        doStar        ;setzen
470:      dbf        d3,install_stars ;fuer alle Sterne wiederhol
471: en
472:      movem.l    (sp)+,d2-d5/a0/a1/a4
473:      rts
474:
475: ;-- erstellt eine Sinustabelle --
476: ;-- a4: Zeiger auf Puffer --
477: ;-- (Puffer = 720 Byte) --
478: ;-----
479: initSinTable:
480:      moveq      #0,d5         ;Fehler-Code vorerst = 0
481:      lea        mathffp_name(pc),a1 ;Zeiger auf Library-Name
482:      moveq      #0,d0         ;Version ist egal
483:      move.l     ExecBase,a6   ;A6:= ExecBase
484:      jsr        OpenLibrary(a6) ;Library oeffnen
485:      move.l     d0,MathBase    ;Base-Pointer sichern
486:      beq        endsin        ;Fehler ? ja: -->
487:
488:      lea        mathtrans_name(pc),a1 ;Zeiger auf Library-Na
489:      moveq      #0,d0         ;Version ist egal
490:      jsr        OpenLibrary(a6) ;Library oeffnen
491:      move.l     d0,MathTransBase ;BasePointer sichern
492:      beq        close_mathbase ;Fehler ? ja: -->
493:
494:      move.l     d5,d4         ;Variable fuer Winkel
495:      moveq      #1,d5         ;Fehler-Code jetzt = 1
496: calc:      move.l     d4,d0     ;D0:= Winkel
497:      move.l     MathBase,a6   ;A6:= MathBase
498:      jsr        SPflt(a6)     ;Winkel in float umwandeln
499:      move.l     #$c90fdb42,d1 ;D1:= PI
500:      jsr        SPMul(a6)     ;Winkel * PI
501:      move.l     #$b4000048,d1 ;D1:= (float)180
502:      jsr        SPDiv(a6)     ;D0 / 180
503:
504:      move.l     MathTransBase,a6 ;A6:= MathTransBase
505:      jsr        SPSin(a6)     ;Sinus berechnen
506:      move.l     d0,FloatSin    ;und sichern
507:
508:      move.l     MathBase,a6   ;A6:= MathBase
509:      move.l     #$9ffec053,d1 ;D1:= (float)32767*10
510:      jsr        SPMul(a6)     ;Sin * D1
511:      jsr        SPFix(a6)     ;Sinuswert in long wandeln
512:      move.l     d0,LongSin    ;und sichern
513:      move.l     FloatSin,d1   ;D1:= (float) Sinuswert
514:      jsr        SPTst(a6)     ;Sinuswert > 0 ?
515:      ble.s      sin_pos      ;ja: -->
516:
517:      move.l     LongSin,d0     ;D0:= (long) Sinuswert
518:      addq.l     #5,d0         ;Sinuswert runden
519:      bra.s      store_sin     ;-->
520:
521: sin_pos:
522:      move.l     LongSin,d0     ;D0:= (long) Sinuswert
523:      subq.l     #5,d0         ;Sinuswert runden
524:
525: store_sin:
526:      divs      #10,d0         ;Sinuswert / 10
527:      move.l     d4,d3         ;D3:= Winkel
528:      asl.l     #1,d3         ;Winkel als Offset
529:      move.w     d0,(a4,d3.w)  ;Sinuswert aspeichern
530:      addq.w     #1,d4         ;Winkel + 1
531:      cmp.w     #360,d4        ;schon alle Sinuswerte ?
532:      blt.s     calc          ;nein: -->
533:
534:      move.l     MathTransBase,a1 ;A1:= MathTransBase
535:      move.l     ExecBase.w,a6 ;A6:= Library-Base
536:      jsr        CloseLibrary(a6) ;MathTrans-Library schliess
537: en
538: close_mathbase:
539:      move.l     MathBase,a1   ;A1:= MathBase
540:      jsr        CloseLibrary(a6) ;MathFFP-Library schliessen
541:      move.l     d5,d0         ;Fehler-Code
542:      rts
543: LongSin:      dc.l 0
544: FloatSin:     dc.l 0
545: MathBase:     dc.l 0
546: MathTransBase: dc.l 0
547: mathtrans_name: dc.b 'mathtrans.library',0
548: mathffp_name:  dc.b 'mathffp.library',0
549:
550: ;--- MegaStars - DATA ---
551: even
552: Starplane1:   dc.l 0
553: Starplane2:   dc.l 0
554: x_max:        dc.w 0
555: y_max:        dc.w 0
556: x_min:        dc.w 3
557: y_min:        dc.w 11
558: x_middle:     dc.w 0
559: y_middle:     dc.w 0
560: BytesPerRow:  dc.l 0
561: rndsav:       dc.w $5279
562: rndlow:       dc.w $79
563: Startable:    blk.w maxStars*2
564:
565: even
566: ;--- Speicher fuer Sinustabelle ---
567: SinusTable:
568:      blk.w 360,0

```

Listing: Megastars



Grafikkarte Highgraph V

mehr Auflösung zu einem guten Preis !!!

technische Daten zur Grafikkarte:

- * max. Auflösung von 832 x 620 Punkten
- * keine schwarzen Zeilen im Non-Interlaced Modus
- * volle 4096 Farben darstellbar - läuft auch im HAM-Modus
- * 50 Hz Ausgabefrequenz (Vollbild)
- * 31.25 kHz Horizontal-Ablenkfrequenz
- * 768 kByte dynamischer Ram
- * 9 pol. Sub-D Ausgang für RGB Analogsignal
- * RGB Digitalausgang auf der Platine über Steckerleiste

AMIGA
Köln 90

*Wir stellen aus
09.-11. November 1990
Halle 10, Stand 605/609*

Preis: 898,- DM

Speichererweiterung CA2000.01 für Amiga 2000

- * 2 Megabyte: 698,- DM
- * 4 Megabyte: 1048,- DM
- * 6 Megabyte: 1398,- DM
- * 8 Megabyte: 1748,- DM

- abschaltbar autokonfigurierend
- 0 Wait-State
- industriell gefertigt
- hochwertige Präzisionssockel
- Steckerkontakte vergoldet

Die Karte wird grundsätzlich mit den PAL-Sätzen für 2/4 und 6/8 Megabyte ausgeliefert!

Quantum 3,5 Zoll Festplatten:

Model	Kapazität	Preis
ProDrive 40S	42 MB	998,- DM
ProDrive 80S	84 MB	1548,- DM

Preise für größere Kapazitäten auf Anfrage.

Die Festplatten sind mit einem **SCSI-Interface** ausgerüstet.

Quantum Festplatte 42 MB 19 msec mit SCSI-Controller: 1698,- DM

Quantum Festplatte 84 MB 19 msec mit SCSI-Controller: 2248,- DM

Einbaurahmen für 3 1/2 Zoll Festplatten im 5 1/4 Zoll Schacht des Amiga 2000: 20,- DM

Disketten:

ECC 3.5 Zoll 1.0 MB 2DD Double Sided:
10er Pack 19,90 DM

Händleranfragen erwünscht!

Höchste Herstellungsqualität sichert Ihnen eine lange Lebensdauer!

RALF JOCHHEIM COMPUTER TUNING

Osnabrücker Straße 96 - 4802 Halle - Tel.: 02823/1275 - Fax: 02823/1350

Gewußt wie!

Hier sind sie wieder, unsere kurzen, knackigen Tips, diesmal aus Platzgründen jedoch nur auf zwei Seiten. Im nächsten Heft gibt es dafür wieder mehr.

Wenig Programm, viel Farbe

Das hier abgedruckte Assemblerprogramm schreibt so lange die aktuelle vertikale Rasterstrahlposition in das Hintergrundfarb-Register, bis die linke Maustaste gedrückt wurde. Dadurch entsteht ein farbiger Effekt auf dem Bildschirm. Man kann die kleine Routine sehr gut in eigene Programme einbauen, da sie ja so gut wie keinen Speicher benötigt. Die Routine ist zum Beispiel sehr praktisch zum Untermalen einer Warnmeldung oder ähnlichem. Am Anfang des Programms werden die Interrupts abgeschaltet, um Störungen zu verhindern, am Ende werden sie dann wieder eingeschaltet.

Hier nun das Listing, das zu allen Assemblern kompatibel ist:

```
lea    $dff000,a5
move.w #$0000,$9a(a5) ;Ausschalten aller IRQ
Loop:
move.b $06(a5),$181(a5)
btst   #6,$bfe001 ; Test left mousebutton
bne.s  Loop
move.w #$c000,$9a(a5) ;Einschalten aller IRQ
rts
```

(Oliver Schmelzle/jb)

Drucker angeschlossen?

Das nachfolgende kleine Modula2-Programm soll zeigen, wie man in »M2Amiga« den Drucker abfragen kann. Dazu wird eine Variable an einer bestimmten Adresse (0BFD0C8H) als »UByte« (Unsigned Byte) definiert. Dann wird diese Variable abgefragt. Wenn diese Variable den Wert 252 (oder HEX 0FCH) bekommt, dann ist der Drucker »online«. Der Wert 253 (oder HEX 0FDH) repräsentiert im Gegensatz den Status »offline«. Enthält die Variable den Wert 254 (oder HEX 0FEH), dann ist das Papier alle, aber das merkt man ja sowieso. Der Wert 255 (oder HEX 0FFH) steht für einen ausgeschalteten Drucker.

Diese kleine Abfrage kann man in jedem M2-Programm einbauen, um den Drucker abzufragen.

```
1 MODULE TestDrucker;
2 FROM Terminal IMPORT WriteString, WriteLn;
3 FROM Exec IMPORT UByte;
4
5 VAR DruckerStatus [0BFD0C8H] : UByte;
6 BEGIN
7 CASE DruckerStatus OF
8 | 0FCH : WriteString ("Drucker ist ONLINE.");
9 | 0FDH : WriteString ("Drucker ist OFFLINE.");
10 | 0FEH : WriteString ("Papier alle.");
11 | 0FFH : WriteString ("Drucker ist AUS.");
12 END;
13 WriteLn
14 END TestDrucker.
```

(Thomas Igracki/jb)

Die Supertips

Ärger mit der Echtzeithuhr?

Manchmal, wenn die Uhr auf der Speichererweiterung A501 versagt, kann ein einzelner BASIC-POKE-Befehl weiterhelfen, nämlich POKEW 14417980&,0 (\$DC003C). Mit ihm wird die Uhr wieder in Gang gesetzt, sofern die Batterie (Akku) noch in Ordnung ist.

Einige Bemerkungen zu der Uhr: Beim Netzeinschalten befindet sich der MC68000 in den ersten Mikrosekunden in einem undefinierten Zustand. Somit können auf dem Bus Signalfeldzustände auftreten, die der Uhr einen realen Befehl vortäuschen. Da vom Entwickler kein Hardwareschutz (ein TTL-Baustein SN 7400 hätte genügt) vorgesehen wurde, der undefinierte Signale verhindert, kann die Uhr gelegentlich verstellt oder ganz gestoppt werden. Da mir das häufiger (ca. 3-5 mal im Jahr) passiert und ich nicht in die Hardware eingreifen will, habe ich das folgende Programm erstellt. Das Programm kann zwar den genannten Mangel nicht verhindern, aber es kann eine stillgelegte Uhr wieder aktivieren.

Die Uhrenregister sind ab Adresse \$DC0000 zu erreichen. Um die Hardware einfach zu halten, ist jedes Register über ein Langwort zu erreichen! Von dem jeweiligen Langwort sind nur die Bits »b0« bis »b3« von Interesse, die restlichen Bits sind ohne Bedeutung. Die Codierung der Zeitregister erfolgt im BCD-Code (binär). Einzelheiten entnehmen Sie bitte der Tabelle.

Adresse	Register	D0	D1	D2	D3	Kommentar
DC0000	r0	s1	s2	s4	s8	0.9 Sekunden-Einer
DC0004	r1	s10	s20	s40	x	0.5 Sekunden-Zehner
DC0008	r2	m1	m2	m4	m8	0.9 Minuten-Einer
DC000C	r3	m10	m20	m40	x	0.5 Minuten-Zehner
DC0010	r4	h1	h2	h4	h8	0.9 Stunden-Einer
DC0014	r5	h10	h20	x	x	0.2 Stunden-Zehner
DC0018	r6	t1	t2	t4	t8	0.9 Tage-Einer
DC001C	r7	t10	t20	x	x	0.3 Tage-Zehner
DC0020	r8	M1	M2	M4	M8	0.9 Monate-Einer
DC0024	r9	M10	x	x	x	0.1 Monate-Zehner
DC0028	rA	j1	j2	j4	j8	0.9 Jahre-Einer
DC002C	rB	j10	j20	j40	j80	0.9 Jahre-Zehner
DC0030	rC	w1	w2	w4	x	0.6 Wochentag
DC0034	rD	HOLD	BUSY	IRQ	ADJ	Steuerregister D
DC0038	rE	MASK	INTR	T0	T1	Steuerregister E
DC003C	rF	RSET	STOP	24/12	TST	Steuerregister F

TST -> 0, 24/12 -> 0 (24h Mode), STOP -> 0 (Uhr läuft) und RSET -> 0.

Achtung: Die angegebenen Adressen und Inhalte gelten für den AMIGA 500, ob es so beim A2000 auch funktioniert, muß probiert werden.

Beispiel: Uhr lesen

```
1 FOR i&=14417920& TO 14417920&+&H3C STEP 4
2 PRINT HEX$(i&),HEX$(PEEKL(i&) AND &HF)
3 NEXT i&
```

Eine letzte Anmerkung: Die Angaben auf Seite 55 im Buch »Amiga Intern« von Data Becker in der dritten Auflage 1988 stimmen zumindest für den A500 nicht, da dort von »Words« (16 Bit Breite) ausgegangen wird.

(Klaus Küchler / jb)

Was sagt das Laufwerk?

Viele Benutzer erkundigen sich immer wieder nach einer Abfragemöglichkeit über die angeschlossenen Laufwerke, ohne daß einer der manchmal recht lästigen Systemrequester erscheint. Ich habe hierfür eine Routine in GFA-BASIC entwickelt. Die Routine arbeitet hundertprozentig mit dem Sy-

stem zusammen, ohne irgendwelche illegalen Systemzugriffe oder -veränderungen. Man übergibt der Funktion die Nummer des zu prüfenden Laufwerks (0-3), sowie drei Variablen des Typs »Boolean«. In diese Variablen werden dann die ermittelten Laufwerks-Zustände eingetragen. Dabei werden die folgenden Zustände abgefragt:

- Ist das Laufwerk physikalisch vorhanden?
- Wenn ja, ist eine Diskette im Laufwerk eingelegt?
- Wenn ja, ist diese Diskette beschreibbar?

Für jede positive Antwort wird die entsprechende Variable auf "TRUE" gesetzt, ansonsten bleibt sie auf "FALSE". Bei fehlerfreiem Durchlauf setzt die Funktion den Rückgabewert "TRUE". Hier nun das Programm

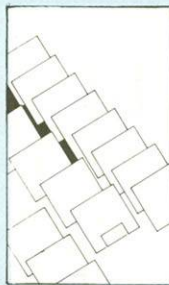
```
1 FUNCTION testdrive(drive!,VAR exist!,disk!,write!)
2   LOCAL iostdreq%,device$,ok%,offset!,io_device%,msgport%,
   flagbits!,back!
3 ' 4   'exist: TRUE = Laufwerk ist vorhanden
5   'disk : TRUE = Diskette im Laufwerk vorhanden
6   'write: TRUE = Diskette nicht schreibgeschützt
7 '
8   drive!:=MIN(drive!,3)    !Laufwerk Nr.0-3
9   CLR exist!,disk!,write!  !Ausgangsstatus setzen
10  iostdreq%=MALLOC(48,&H1 OR &H10000)    !PUBLIC!CLEAR
11  IF iostdreq%=0
12     RETURN FALSE
13  ENDIF
14  device$="trackdisk.device"+CHR$(0)
15  ~FRE(0)
16  ok%=OpenDevice(V:device$,0,iostdreq%,0)
17 '
18  IF ok%=0
19     io_device%={iostdreq%+20}    !Trackdisk-Device-
   Struktur
20     offset!:=ADD(36,SHL(drive!,2))    !Offset für MsgPort
   von Laufwerk x
21     msgport%={io_device%+offset!}    !MsgPort für Laufwerk
   x ermitteln
22 '
23     IF msgport%    !Existiert ein MsgPort?
24        exist!=TRUE
25        flagbits!:=BYTE[msgport%+64]    !Status-Bits ermitt.
26        IF NOT BTST(flagbits!,1)    !Laufwerk nicht leer?
27           disk!=TRUE
28           IF NOT BTST(flagbits!,4)    !Diskette nicht
   schreibgeschützt?
29              LET write!=TRUE
30          ENDIF
31      ENDIF
32  ENDIF
33 '
34  ~CloseDevice(iostdreq%)
35  back!=TRUE    !Kein Fehler aufgetreten!
36  ENDIF
37 '
38  ~MFREE(iostdreq%,48)
39 '
40  RETURN back!
41 ENDFUNC
```

(Torsten Holland/jb)

Modula2 und die FileInfos

Recht seltsame Dinge geschehen, wenn Sie in Modula2 RECORDS in lokalen Prozeduren verwenden und diese nicht in der richtigen Reihenfolge angeben.

```
FROM Dos IMPORT ..., FileLock, FileLockPtr,
FileInfoBlock, FileInfoBlockPtr;
...
PROCEDURE ReadDir;
VAR DirLock : FileLock;
InfoBlock : FileInfoBlock;
DirLockPtr : FileLockPtr;
InfoBlockPtr : FileInfoBlockPtr; BEGIN (* ReadDir *)
DirLockPtr := Lock("DF0:", accessWrite);
```



Tausche Tip gegen DATABOX

Wir machen Ihnen wieder unser Angebot: Sie schicken einen oder mehrere kurze Tips an uns; bei Gefallen bekommen Sie von uns im Austausch die DATABOX des Heftes zugeschickt, in dem Ihr Tip erscheint.

Dazu gibt es aber jetzt noch etwas mehr:

Die drei besten Tips werden mit je 100,- DM belohnt.

Die Redaktion der AMIGA DOS sucht ab sofort unter allen Einsendern bis zu drei **Supertips pro Monat** aus, die außer der DATABOX noch je einhundert deutsche Marker (100,- DM) bekommen.

Dabei sollten Sie noch folgendes beachten:

– Schreiben Sie Ihren Tip als ASCII-Datei auf Ihre Diskette, möglichst ohne eigensinnige Verschnörkelungen, da alle Texte zur weiteren Verarbeitung auf Personal-Computer transferiert werden müssen. Sonder- und Steuerzeichen haben dabei meist seltsame Auswirkungen.

– Schreiben Sie auf den Brief sowie auf einen kleinen Aufkleber auf der Diskette Ihre lückenlose Anschrift und einen kleinen Hinweis, was auf der Diskette zu finden ist.

Dann können Sie sicher sein, daß die DATABOX Sie erreicht. Schicken Sie das Ganze an den

DMV-Verlag
Redaktion AMIGA DOS
Kennwort "Tausche Tip gegen Databox"
Postfach 250
D-3440 Eschwege

```
...
InfoBlockPtr := ADR(InfoBlock);
DosError := Examine(DirLockPtr, InfoBlockPtr);
WHILE DosError # FALSE DO
ExNext(DirLockPtr, InfoBlockPtr);
WriteString(InfoBlockPtr↑.fileName);
WriteLn;
END (* WHILE *);
END (* IF *);
END ReadDir;
```

Das oben abgedruckte (unvollständige) Programm sollte eigentlich den Inhalt der Diskette in »DF0:« in das CLI-Fenster ausgeben. Im günstigsten Fall werden die zwei ersten Zeichen des Filenamens abgeschnitten, im ungünstigsten Fall meldet sich der Herr der Programmierer (»GURU«) wieder mal. Warum?

Aus irgendeinem Grund befindet sich der Anfang der »InfoBlock«-Struktur an einer ungeraden Adresse. Versucht das Programm nun, dem »InfoBlockPtr« die Adresse des InfoBlocks zuzuweisen, gibt es Ärger.

Beseitigen läßt sich dieser Fehler, wenn Sie die Variablen-Deklaration folgendermaßen ausführen:

```
...
PROCEDURE ReadDir;
VAR DirLock : FileLock;
DirLockPtr : FileLockPtr;
InfoBlock : FileInfoBlock;
InfoBlockPtr : FileInfoBlockPtr; BEGIN (* ReadDir *)
...
(tb)
```


Willi Bäcker

Wie geht das in Assembler?

So gut Assemblerprogramme im Endeffekt aussehen, so schwer ist es manchmal, alles zu verstehen.

Dieses kleine Demo-Programm dient zur einfachen Ein- und Ausgabe von Texten im aktuellen CLI-Window. Bei der Texteingabe sind 80 Zeichen möglich. Bei der Textausgabe ist allerdings zu beachten, daß der String mit einem Nullbyte beendet wird.

»DosPrint« und »DosInput«

Der Unteroutine »DosPrint« wird ein Zeiger im Register »A0« übergeben, der auf die Adresse des Textes hinweist:

```
lea    Text0(pc),a0
bsr    DosPrint
```

Bei »DosInput« werden die eingelesenen Zeichen bis [Return] in den Speicher ab »InBuff« gelesen (siehe Listing). Dort können Sie dann mit »DosPrint« ausgegeben werden.

Greetings from India

Das Programm »GuruTest« zeigt, wie man unter Zuhilfenahme der »Intuition-Library« die berühmte »Störstelle« für sich selbst nutzt. Als Parameter müssen nur die Boxgröße, die Position des Textes und der Text selbst an die Funktion »DisplayAlert« übergeben werden, nachdem Sie die »Intuition-Library« geöffnet haben. Nun muß die Funktion nur noch aufgerufen werden. Den freien Speicher anzeigen zu lassen ist oft wünschenswert, vor allem, wenn man nicht genug davon hat. »ShowMemory« ist ein kleines Programm, das Informationen über Chip-Mem und Fast-Mem von sich gibt. Benutzt werden die »Exec-Library« mit den Funktionen »AvailMem«,

»GetMsg« und »ReplayMsg«, um die augenblickliche Speicherbelegung auszugeben, die »Intuition-Library« für Windows und die »Graphics-Library«, um dem Text ein besseres Aussehen zu verpassen.

Zeigt her euren Speicher...

Die Anzeige im Window wird alle 0,5 Sekunden aktualisiert, ein »Close-Gadget« beendet das Programm. Ohne Joystick oder Maus geht beim Amiga (fast) nichts. Doch wie werden Sie abgefragt? Ein Register des CIA-Chips A und ein Register des Custom-Chips »Paula« sind hierbei

wichtig: Das Register »CIAA-PRA« mit seinem Bit Nummer 6 in der CIA für die linke Maustaste, und das Register »POTGOR« des Paula-Chips, das ein Auslesen der rechten Maustaste in Bit 10 ermöglicht.

Test the Byte

Im Bit 7 des CIA-Registers steht außerdem noch die Information, ob der Joystick-Knopf gedrückt wurde. Durch überprüfen des jeweiligen Bits durch den Befehl »BTST« (Bit-Test) wird entweder zum Label zurückgesprungen (bei Bit=0) oder zum nächsten Label weitergefahren (Bit=1).

Halt! Stop! Warte!

»Level_3_Int_Demo«, hinter diesem Programmnamen verbirgt sich eine Routine, die den Level-3-Interrupt (Vertical-Blank-Interrupt) zum Thema hat. Dieser Interrupt wird jede 1/50 Sekunde ausgelöst, und ist daher hervorragend für das Abarbeiten von Programnteilen geeignet, die das Hauptprogramm sonst unnötig unterbrechen würden. Als Beispiel sei hier das Verändern der Copper-Listen genannt. Experimentieren Sie ruhig einmal. (jb/vb)

Listings

```
1: ;*****
2: ;*      DOSPRINT + DOSINIT      *
3: ;*  kleine Demo-Routine für Text Ein- und Ausgabe auf *
4: ;*  dem CLI Window!            *
5: ;*      by Willi Bäcker        *
6: ;*      (c) 1990 DMV          *
7: ;*      Sprache: Assembler     *
8: ;*****
9:
10: EXEC = 4
11:
12: OpenLibrary = -552
13: CloseLibrary = -414
14:
15: Read = -42
16: Write = -48
17: Input = -54
18: Output = -60
19:
20: Start:
21:   move.l    a7,InitialSP      ;Stackpointer retten
22:   move.l    EXEC,a6           ;only absolute!
23:   moveq     #0,d0              ;Version
24:   lea.l     DosName(pc),a1     ;Library Name
25:   jsr       OpenLibrary(a6)    ;(exec.library)
26:   beq.s     LibraryError       ;Fehler ?
27:   move.l    d0,DosBase        ;DosBasis retten
28: ;*****
29:   move.l    d0,a6              ;DosBasis holen
30:   jsr       Output(a6)         ;(dos.library)
31:   move.l    d0,ClIHandle       ;OutputHandle
32:   jsr       Input(a6)          ;(dos.library)
33:   move.l    d0,ClIHandle       ;InputHandle
34: ;*****
35:   lea       Text0(pc),a0       ;Aufrufen der
36:   bsr       DosPrint           ;Print und Input
37:   bsr       DosInput          ;Routine.
38:   lea       Text1(pc),a0
39:   bsr       DosPrint
40:   lea       InBuff(pc),a0
41:   bsr       DosPrint
42:   lea       Text2(pc),a0
43:   bsr       DosPrint
44: ;*****
45: CloseLib:
```

Listing 1. DOSPRINT + DOSINPUT

Software-Autoren für die Amiga-Computer gesucht

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. Senden Sie uns Ihr Programm mit

- (a) allen benötigten Files auf der mit dem Programmnamen bezeichneten Diskette,
- (b) den kompletten Ausdrucken/Listings aller Files der Diskette,
- (c) einer Beschreibung Ihres Programms und
- (d) einer genauen Bedienungsanleitung.

Die Bedienungsanleitung und die Beschreibung sollten als Textdatei mit auf der Programm-Diskette enthalten sein. Wichtig für uns zu wissen wäre noch, mit welcher Konfiguration Sie arbeiten, welchen Drucker Sie benutzen, ob Sie ein zweites Laufwerk angeschlossen haben usw.

Wenn Sie der Meinung sind, ein solches Programm geschrieben zu haben, dann nichts wie einschicken an den

**DMV-Verlag
Redaktion AMIGA DOS
Postfach 250
3440 Eschwege**


```

46:      move.l   EXEC,a6           ;ExecBase holen
47:      move.l   DosBase(pc),a1    ;DosBase holen
48:      jsr      CloseLibrary(a6)  ;Library schließen!
49:      moveq     #0,d0            ;CLI Return-Code auf
50:                                ;RETURN OK setzen!
51:      move.l   InitialSP,a7      ;Stackadresse zurück
52:      rts      ;Beam me up, Scotty!
53: ;*****
54: LibraryError:
55:      moveq     #20,d0
56: ; CLI-Return-Code auf RETURN_FAIL setzen! Erwirkt
57: ; Abbruch des Script Files!
58:      rts
59: ;*****
60: DosPrint:
61:      movem.l   d1-d3/a0,-(sp)    ;Register auf Stack
62:      moveq     #-1,d3            ;Länge löschen
63:      DosPrint0:
64:      addq.l    #1,d3            ;Stringlänge
65:      cmpi.b    #0,(a0,d3.w)      ;ermitteln
66:      bne.s     DosPrint0
67:      move.l    a0,d2            ;Pointer auf String
68:      move.l    ClioHandle(pc),d1 ;Windowhandle holen
69:      move.l    DosBase(pc),a6    ;Library Basis
70:      jsr      Write(a6)          ;(dos.library)
71:      movem.l   (sp)+,d1-d3/a0    ;Register zurück
72:      rts      ;Rücksprung
73: ;*****
74: DosInput:
75:      movem.l   d0-d3/a0,-(sp)    ;Register retten
76:      lea       InBuff(pc),a0     ;Buffer Adresse
77:      moveq     #79,d0            ;zuerst Buffer
78:      DosLoop:
79:      clr.b     (a0)+            ;löschen
80:      dbra     d0,DosLoop
81:      move.l    DosBase(pc),a6    ;LibraryBasis holen
82:      move.l    ClioHandle(pc),d1 ;Windowhandle holen
83:      move.l    #InBuff,d2       ;Bufferadresse holen
84:      move.l    #80,d3           ;max Länge
85:      jsr      Read(a6)          ;(dos.library)
86: ;In d0 steht jetzt die Anzahl der tatsächlich
87: ;gelesenen Zeichen
88:      lea.l     InBuff(pc),a0     ;Bufferadresse holen
89:      clr.b     -1(a0,d0.w)       ;RETURN Byte löschen
90:      movem.l   (sp)+,d0-d3/a0    ;Register zurück
91:      rts      ;Rücksprung
92: ;*****
93: DosBase:      dc.l 0             ;DosLibrary Basisadresse
94: CliOHandle:   dc.l 0             ;Outputhandle des CLI-Windows
95: CliIHandle:   dc.l 0             ;Inputhandle des CLI-Windows
96: InitialSP:    dc.l 0             ;Stackadresse
97: ;*****
98: InBuff:       blk.b 80,0         ;Buffer für Texteingaben
99: ;*****
100: DosName:      dc.b "dos.library",0 ;Library Name
101: ;*****
102: Text0:        dc.b 10,10
103:               dc.b 9,"Bitte Namen eingeben :",0
104: Text1:        dc.b 9,"Guten Tag ",0
105: Text2:        dc.b 10,0
106: ;*****

```

Listing 1. DOSPRINT + DOSINPUT

```

1: ;*****
2: *          Guru-Test
3: *          by Willi Bäcker
4: *          (c) 1990 DMV
5: *          Sprache: Assembler
6: ;*****
7:
8: ExecBase      =      4
9:
10: OldOpenLibrary = -408
11: CloseLibrary  = -414
12: DisplayAlert  = -90
13:
14: Guru_Box_Hoehe: equ 40
15:
16: start:
17:      move.l   ExecBase,a6       ;ExecBase holen
18:      lea      IntName(pc),a1    ;LibNameadresse holen
19:      jsr      OldOpenLibrary(a6);Library öffnen
20:      beq.s    LibError          ;Fehler!
21:      move.l   d0,a6            ;IntBase retten
22:      lea      Alerttext(pc),a0  ;Adresse der Textstruktur
23:      moveq     #0,d0            ;RECOVERY Alert
24:      move.l    #Guru_Box_Hoehe,d1;AlertBox Höhe
25:      jsr      DisplayAlert(a6)  ;Do a Guru
26:      move.l    a6,a1            ;ho!e IntBasisadresse
27:      move.l    ExecBase,a6      ;ExecBasisadresse holen
28:      jsr      CloseLibrary(a6) ;Library schließen
29: LibError:
30:      moveq     #0,d0            ;keine Parameter
31:      rts      ;zurück
32: ;-----
33: ;DemoTextstruktur für DisplayAlert (INTUITION.LIBRARY)
34: Alerttext:
35:      dc.b     0                ;xpos high
36:      dc.b     220              ;xpos low
37:      dc.b     15              ;ypos
38:      dc.b     'Amiga Guru Meditation V1.0' ;string
39:      dc.b     0                ;Text 1 Ende
40:      dc.b     1                ;noch ein Text
41:      dc.b     0                ;xpos high
42:      dc.b     220              ;xpos low
43:      dc.b     18              ;ypos
44:      dc.b     ' '                ;string
45:      dc.b     0                ;Text 2 Ende
46:      dc.b     2                ;noch ein Text
47:      dc.b     0                ;xpos high
48:      dc.b     220              ;xpos low
49:      dc.b     30              ;ypos
50:      dc.b     'Bitte Maustaste dr',252,'cken !' ;string

```

Listing 2. Guru Test

```

51:      dc.b     0                ;Text 3 Ende
52:      dc.b     0                ;kein Text mehr
53:      even
54:
55: IntName: dc.b 'intuition.library',0

```

Listing 2. Guru Test

```

1: ;*****
2: *          ShowMemory
3: *          by Willi Bcker
4: *          (c) 1990 DMV
5: *          Sprache: Assembler
6: ;*****
7:
8: Exec: equ 4
9: MsgPort: equ 86
10:
11:
12: *** EXEC LIB ***
13: OpenLibrary = -552
14: CloseLibrary = -414
15: AvailMem = -216
16: GetMsg = -372
17: ReplyMsg = -378
18:
19: *** INTUITION LIB ***
20: OpenWindow = -204
21: CloseWindow = -72
22: WBenchToFront = -342
23:
24: *** GRAPHICS LIB ***
25: SetAPen = -342
26: SetBPen = -348
27: Move = -240
28: Text = -60
29:
30: *** Dos ***
31: Delay = -198
32: ;*****
33: go:
34:      lea      IntName(pc),a1    ;Zeiger auf Libname
35:      moveq     #0,d0            ;Version
36:      move.l    EXEC,a6          ;ExecBasis holen
37:      jsr      OpenLibrary(a6)   ;Library ffnen
38:      beq      NoIntuition       ;Fehler
39:      move.l    d0,IntBase       ;IntBasis retten
40:      lea      GfxName(pc),a1    ;Zeiger auf Libname
41:      moveq     #0,d0            ;Version
42:      move.l    EXEC,a6          ;ExecBasis holen
43:      jsr      OpenLibrary(a6)   ;Library ffnen
44:      beq      NoGraphics        ;Fehler
45:      move.l    d0,GfxBase       ;GfxBasis retten
46:      lea      DosName(pc),a1    ;Zeiger auf Libname
47:      moveq     #0,d0            ;Version
48:      move.l    EXEC,a6          ;ExecBasis holen
49:      jsr      OpenLibrary(a6)   ;Library ffnen
50:      beq      NoDos             ;Fehler
51:      move.l    d0,DosBase       ;DosBasis retten
52:      lea      WindowStruct(pc),a0;Zeiger auf Struktur
53:      move.l    IntBase,a6       ;IntBasis holen
54:      jsr      OpenWindow(a6)    ;Window ffnen
55:      tst.l     d0               ;test auf Fehler
56:      beq      NoWindow          ;Fehler?
57:      move.l    d0,WindowPointer ;Pointer retten
58:
59:      move.l    WindowPointer(pc),a0 ;und holen
60:      move.l    50(a0),a1        ;RastPort nach a1
61:      moveq     #4,d0            ;Pen
62:      move.l    GfxBase,a6       ;GfxBasis holen
63:      jsr      SetAPen(a6)       ;Zeichenfarbe setzen
64:      move.l    WindowPointer(pc),a0 ;und holen
65:      move.l    50(a0),a1        ;RastPort nach a1
66:      moveq     #3,d0            ;Pen
67:      move.l    GfxBase,a6       ;GfxBasis holen
68:      jsr      SetBPen(a6)       ;Zeichenfarbe setzen
69:
70: ;*****
71: Wait Msg:
72:      bsr      Memory            ;Werte holen
73:      move.l    WindowPointer(pc),a0 ;Windowp. holen
74:      move.l    MsgPort(a0),a0   ;Port holen
75:      move.l    EXEC,a6          ;ExecBasis holen
76:      jsr      GetMsg(a6)        ;(exec.library)
77:      tst.l     d0               ;keine Msg da?
78:      beq.s     NoMsg            ;etwas warten
79:      move.l    d0,a1            ;Msg holen
80:      move.l    EXEC,a6          ;ExecBasis holen
81:      jsr      ReplyMsg(a6)      ;Msg quittieren
82:      bra.l     Close_Window    ;Window schließen
83: NoMsg:
84:      moveq     #25,d1           ;0.5 sec warten
85:      move.l    DosBase,a6       ;Dos Basis holen
86:      jsr      Delay(a6)         ;(dos.library)
87:      bra.s     Wait Msg        ;Nochmal!
88: ;*****
89: Memory:
90:      move.l    EXEC,a6          ;hier wird festgestellt
91:      moveq     #2,d1            ;wieviel
92:      jsr      AvailMem(a6)      ;Speicher noch frei
93:      move.l    d0,ChipMem       ;ist
94:      move.l    EXEC,a6

```

Listing 3. ShowMemory



TIPS & TRICKS

```

95: moveq #4,d1
96: jsr AvailMem(a6)
97: move.l d0,FastMem
98: move.l ChipMem(pc),d1
99: add.l d1,d0
100: move.l d0,Chip_Fast
101: move.l dummy(pc),d1 ;Vergleichswert holen
102: cmpi.l d0,d1 ;hat sich was gendert?
103: beq out ;nein!
104: move.l Chip_Fast(pc),dummy ;neuer Vergleichs Wert
105: lea text0(pc),a0 ;ab hier werden die
106: adda.l #07,a0 ;binr Zahlen in
107: move.l ChipMem(pc),d0 ;ASCII umgewandelt
108: bsr Con2ASCII ;und in die Textstruktur
109: lea text0(pc),a0 ;geschrieben
110: adda.l #23,a0
111: move.l FastMem(pc),d0
112: bsr Con2ASCII
113: lea text0(pc),a0
114: adda.l #39,a0
115: move.l Chip_Fast(pc),d0
116: bsr Con2ASCII
117:
118: move.w #30,d0 ;x pos
119: move.w #07,d1 ;y pos
120: move.l WindowPointer(pc),a1 ;WindowPointer
121: move.l 50(a1),a1 ;RastPort holen
122: move.l GfxBase(pc),a6 ;GfxBasis holen
123: jsr Move(a6) ;GfxCursor positioni.
124:
125: move.l WindowPointer(pc),a1 ;WindowPointer
126: move.l 50(a1),a1 ;RastPort holen
127: lea Text0(pc),a0 ;Pointer auf String
128: moveq #47,d0 ;Lnge
129: move.l GfxBase(pc),a6 ;GfxBase holen
130: jsr Text(a6) ;Text ausprinten
131: out: rts
132: *****
133: Close_Window:
134: move.l WindowPointer(pc),a0
135: move.l IntBase(pc),a6
136: jsr CloseWindow(a6) ;Window schließen
137: NoWindow:
138: move.l DosBase(pc),a1
139: move.l EXEC,a6
140: jsr CloseLibrary(a6) ;DosLib schließen
141: NoDos:
142: move.l GfxBase(pc),a1
143: move.l EXEC,a6
144: jsr CloseLibrary(a6) ;GfxLib schließen
145: NoGraphics:
146: move.l IntBase(pc),a1
147: move.l EXEC,a6
148: jsr CloseLibrary(a6) ;IntLib schließen
149: NoIntuition:
150: moveq #0,d0
151: rts ;beam me up,Scotty!!
152: *****
153: ;Zahl in d0 wird in Buffer ab a0 als ASCII-Zahl
154: ; geschrieben.
155:
156: Con2ASCII: moveq #7,d1 ;Zhler
157: ConLoop0:
158: rol.l #4,d0
159: move.l d0,d2
160: andi.b #f,d0
161: cmp.b #9,d0
162: ble.s ConLabel0
163: addq.b #7,d0
164: ConLabel0:
165: addi.b #30,d0
166: move.b d0,(a0)+
167: move.l d2,d0
168: dbra d1,ConLoop0
169: rts
170: *****
171: WindowStruct:
172: dc.w 150 ;links
173: dc.w 0 ;oben
174: dc.w 470 ;Breite
175: dc.w 10 ;Hhe
176: dc.b -1 ;Fein-Stift
177: dc.b -1 ;Grob-Stift
178: dc.l $200 ;IDCMP-Flag(REPORT-MOUSE)
179: dc.l $100e ;Flags
180: dc.l 0 ;First Gadget
181: dc.l 0 ;CheckMark
182: dc.l WindowName ;Pointer to WindowName
183: dc.l 0 ;Pointer to Screen
184: dc.l 0 ;User BitMap
185: dc.w 0 ;Min. Breite
186: dc.w 0 ;Hhe
187: dc.w 0 ;Max. Breite
188: dc.w 0 ;Hhe
189: dc.w 1 ;Screen-Typ
190: *****
191: WindowPointer:
192: dc.l 0 ;Pointer auf Window
193: WindowName:
194: dc.b "ShowMem written by Willi Bcker of FACT"
195: dc.b "OR 5 "
196: dc.b 0
197: even

```

Listing 3. ShowMemory

```

198: *****
199: IntName: dc.b 'intuition.library',0
200: GfxName: dc.b 'graphics.library',0,0
201: DosName: dc.b 'dos.library',0
202: IntBase: dc.l 0
203: GfxBase: dc.l 0
204: DosBase: dc.l 0
205: TextPos: dc.l 0
206: StringPointer: dc.l 0 ;Pointer auf akt. Text
207: Flag: dc.l 0
208: ChipMem: dc.l 0
209: FastMem: dc.l 0
210: Chip_Fast: dc.l 0
211: Dummy: dc.l 0
212: TextYPos: dc.w 18 ;First Y Text-Pos
213: *****
214: text0:
215: dc.b ' CHIP $00000000 FAST $00000000 BOTH $00
000000',0

```

Listing 3. ShowMemory

```

1: ;*****
2: ; Buttons
3: ; Einfaches Programm zur Abfrage der Buttons!
4: ; by Willi Bcker
5: ; (c) 1990 DMV
6: ; Sprache: Assembler
7: ;*****
8:
9:
10: CIAAPRA = $bfe001 ;CIA A Portregister
11: POTGOR = $dff016 ;PotI Fead
12:
13:
14: Mouseleft:
15: btst #6,CIAAPRA ;testet linke Maustaste
16: bne.s MouseLeft
17:
18: Mouseright:
19: btst #10,POTGOR ;testet rechte Maustaste
20: bne.s Mouseright
21:
22: Joystickbutton:
23: btst #7,CIAAPRA ;testet Joystickknopf
24: bne.s Joystickbutton
25: rts

```

Listing 4. Buttons

```

1: ;*****
2: ; Level 3 Int Demo
3: ; by Willi Bcker
4: ; (c) 1990 DMV
5: ; Sprache: Assembler
6: ;*****
7:
8: jmp START
9:
10: ;*****
11: ExecBase: equ 4
12: ;***** Exec-Library *****
13: OldOpenLibrary: equ -408
14: CloseLibrary: equ -414
15: ;*****
16: ;Konstanten,Buffer etc. *****
17: ;*****
18: GfxName: dc.b 'graphics.library',0,0
19: GfxBase: dc.l 0
20: OldCop: dc.l 0
21: ;*****
22: align 4
23: START:
24: move.l ExecBase,a6 ;Exec Basisadresse holen
25: lea.l GfxName(pc),a1 ;Adresse des Libnamens
26: jsr OldOpenLibrary(a6) ;Library öffnen
27: tst.l d0 ;Ergebnis testen
28: beq LibError ;zurück wenn Fehler
29: move.l d0,GfxBase ;Gfx_Lib basisadresse retten
30:
31: move.l $6c,oldvec ;Alten Level3 Vektor retten
32: move.l #Level3,$6c ;neuen installieren
33:
34: mouse:
35: btst #6,$bfe001 ;linke Maustaste
36: bne.s mouse ;abfragen.
37:
38: move.l oldvec,$6c ;Alten Interrupt Vektor
39: ;zurückschreiben.
40: move.l ExecBase,a6 ;Exec Basis-Adresse holen
41: move.l GfxBase,a1 ;Library Base holen
42: jsr CloseLibrary(a6) ;Library schließen
43: LibError:
44: moveq #0,d0 ;no CLI parameter
45: rts ;beam me up,Scotty!
46: ;
47: ; ** Level 3 Interrupt Handler
48: ;
49: Level3:
50: movem.l d0-d7/a0-a6,-(a7) ;Register retten
51: btst #5,$dff01f ;Vblank interrupt ?
52: beq.s exit ;ja
53: bsr.s intl ;Neue Interrupt Routine
54: exit:
55: movem.l (a7)+,d0-d7/a0-a6 ;register wiederholen
56: dc.w $4ef9 ;jmp xxxxxxxx.l
57: oldvec: dc.l $0
58:
59: intl:
60: move.w $dff006,$dff180
61: bchg #1,$bfe001 ;Power Led
62: rts ;zurück

```

Listing 5. LEVEL_3_INT_DEMO

SECOND HAND COMPUTER

Ankauf * Verkauf * Vermittlung * Inzahlung

Gebraucht-Computer:

- * Zubehör
- * Neugeräte
- * alle Marken
- * Konkursware-Ankauf
- * Ankauf defekter Geräte

Angebote solange Vorrat

WO SONST

Wir kaufen und verkaufen:

- * Homecomputer
- * XTs und ATs
- * Büroanlagen
- * Anrufbeantworter

NEU:
Jetzt auch
in Kassel!

386er z.B. DELL, 30 Mhz 8 MB RAM, 300 MB 16 Ms HD, VGA, VGA-Farb-Monitor, Coproz., ESDI-Controller, etc. 15690,-
IBM z.B. 386-SC, 2 MB RAM, 60 MB HD, VGA-Monitor 5995,-
MODEM 2400 Baud, Hayes-kompat., neu, (*) 399,-
ATs z.B. 12 Mhz, neu, 512 KB, 102Tast., 200 Watt Netz, 386-SX z.B. 16 Mhz, neu, 1 MB, 102Tast., 200 Watt Netz, Monochrom-Karte, Baby-Gehäuse, 1 Laufwerk 999,-
386-SX z.B. 16 Mhz, neu, 1 MB, 102Tast., 200 Watt Netz, Monochrom-Karte, Baby-Gehäuse, 1 Laufwerk 1499,-
DRUCKER z.B. Mannesmann MT81, 130 Z/sek., neu 389,-
Mannesmann MT230, Farba, Stapel-EBEZ (neu 5000,-) 2395,-
Bei mit (*) gekennzeichneten Geräten ist der Betrieb in der BRD lt. Par. 16 FAG bei Strafe verboten.
Weitere gebrauchte und neue Drucker, Monitore, XTs, ATs und 386er sowie Laptops, Bücher, Software, RAM-Erweiterungen und vieles mehr auf tel. Anfrage!!!

ALPHA 2000 GmbH, 24 Std. Info: 069-443000, 6 Frankfurt/M. 1, Ingolstädter Str. 27
ALPHA 2001 GmbH, 24 Std. Info: 05 61-52 50 66, 3501 Niestetal (bei Kassel), Witzenhäuser Str. 10

Public Domain Studio Nürnberg GmbH

Werderstr. 4, 8500 Nürnberg 20

Tel.: 09 11/53 63 36

Ca. 10000 Public Domain + Shareware Disketten
für Amiga, Atari ST und PC.
Gratis Katalog und Info für Ihren Computer anfordern !!!

DISKETTEN

3.5" 2DD			5.25" 2D		
	10 St.	50 St.		10 St.	100 St.
Blau	DM 10.90	50.00	Schwarz	DM 5.90	54.90
Grau	DM 10.90	50.00	Weiss	DM 9.90	94.90
Gelb	DM 17.90	85.90	Rot	DM 9.90	94.90
Rot	DM 17.90	85.00	Gelb	DM 9.90	94.90
Grün	DM 17.90	85.90	Grün	DM 9.90	94.90
Schwarz	DM 17.90	85.90	Blau	DM 9.90	94.90

3.5 HD		5.25" HD	
Schwarz	DM 29.90 140.00	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

Ist Ihr Programm der Hit?

Der DMV-Verlag sucht ständig nach neuer, interessanter Software zur Aufnahme in sein Softwaresortiment.

Dabei ist es einerlei, ob Sie nun ein Anwendungs- oder ein Spielprogramm geschrieben haben.

Der DMV-Verlag bietet Ihnen sein Software-Know-how an!

DMV-Verlag, Fuldaer Str. 6, 3440 Eschwege

PULSAR

THE RISING STAR IN SOFTWARE

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	
Pixel 3D	Einzigartiger Autotracer. Ein MUSS für jeden, der sich mit 3D - Grafik beschäftigt. D Gesammelte Bilder können in wenigen Sekunden zu 3D - Objekten konvertiert werden. D Alle Farben der Bitmap werden übernommen. D Über einen Glättungsfaktor kann der störende "Treppen - Effekt" eliminiert werden. D Unterstützt viele 3D - Formate wie: Turbo Silver - Sculpt Animate - Autocad DXF - Videscape.	169,- DM
Vo Rec One	Spracherkennung für den Amiga! Unterscheidet bis zu 70 verschiedene Wörter und bietet folgende Features: D Alle Amiga Grafikfunktionen können gesteuert werden. D Scroll Bars können gesteuert werden. D Digitalisierte Sounds können abgespielt werden. D Zeigt und "liest" Texte. D Wird mit deutscher oder englischer Anleitung ausgeliefert. D 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 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.	179,- 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)	79,- DM
Kabellose Maus	Jetzt können Sie Ihren Amiga über 3 m Entfernung mit der Infrarotmaus "fernbedienen" (Practical Solution) Der Betrieb dieser Modems am Bundesdeutschen Postnetz ohne Postzulassung ist unter Strafandrohung verboten.	228,- DM

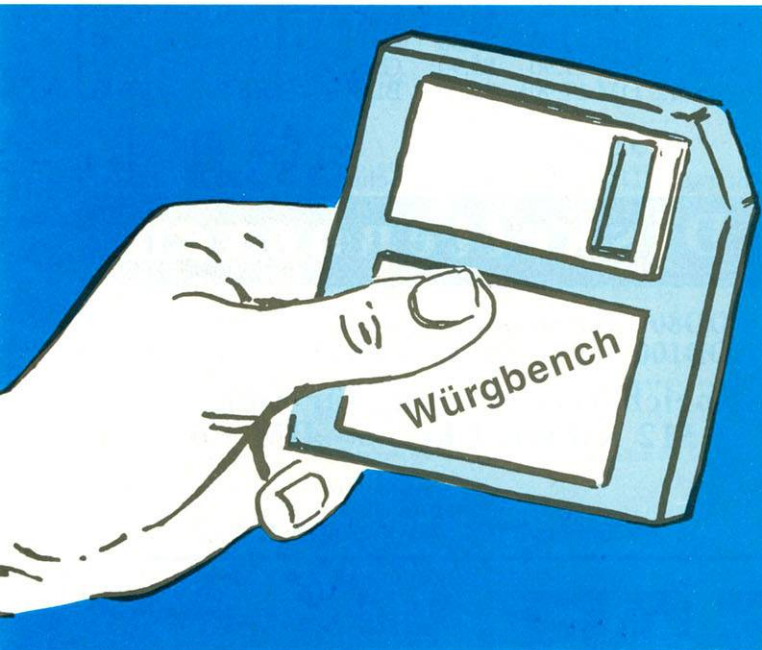
8 MB Speichererweiterung für A2000 0MB unbestückt *	389,- DM	A1000 30MB Hard Drive Quantum 11 ms. SCSI Contr. *	1579,- DM
8 MB Speichererweiterung für A2000 2MB bestückt *	589,- DM	A1000 40MB Hard Drive Quantum 11 ms. SCSI Contr. *	1789,- DM
8 MB Speichererweiterung für A2000 4MB bestückt *	989,- DM	A1000 80MB Hard Drive Quantum 11 ms. SCSI Contr. *	2389,- DM
8 MB Speichererweiterung für A2000 6MB bestückt *	1589,- DM	A2000 40MB Hard Card Quantum 11 ms. SCSI Contr. *	1279,- DM
8 MB Speichererweiterung für A2000 8MB bestückt *	1589,- DM	A2000 80MB Hard Card Quantum 11 ms. SCSI Contr. *	1899,- DM
Insider II 1.5MB Speichererweiterung für A1000 unbestückt	379,- DM	A2000 105MB Hard Card Quantum 11 ms. SCSI Contr. *	2130,- DM
Insider II 1.5MB Speichererweiterung für A1000 1.5MB	559,- DM	A500 40MB Hard Drive Quantum 11 ms. SCSI Contr. *	1479,- DM
Baseboard 4MB Ram für A500 intern unbestückt	329,- DM	A500 80MB Hard Drive Quantum 11 ms. SCSI Contr. *	2179,- DM
U.S. Robotics HST 14.400 Modem*	1489,- DM	44MB Wechselplatte Syquest mit Cartridge	1945,- DM
2400 Baud Modem extern **	295,- DM	44MB Syquest Cartridge	299,- DM
2400 Baud Modem intern für A2000 **	489,- DM	Mouse Stick (programmierbarer Joystick - Gravis)	179,- DM

Alle mit * gekennzeichneten Artikel sind von SUPRA.

Oliver Wagner

Joke-Handler

Die Interna des AmigaDOS werden allgemein als großes Geheimnis angesehen.



Dies ist kein Wunder, sind doch Begriffe wie »BCPL-Zeiger«, »Packets«, »synchrone und asynchrone IO« geradezu geschaffen dafür, weniger erfahrene Programmierer sich schauernd abwenden zu lassen. Die schon notorisch ungenügende Dokumentation trägt das ihrige zur Verwirrung bei.

In diesem Artikel sollen einmal exemplarisch die Innereien des DOS durchleuchtet werden, in dem ein DOS-Handler entworfen wird.

Das Dateisystem unter AmigaDOS ist sehr modular aufgebaut. An unterster Ebene stehen die Hardware-Treiber, auch Devices genannt. Diese bieten eine (fast) einheitliche Schnittstelle zwischen Betriebssystem und Hardware. Beispiele für Devices sind das »trackdisk.device« zur Ansteuerung von Diskettenlaufwerken oder das »console.device«, das die Textein- und -ausgabe in Fenster steuert. Wesentliches Merkmal eines Devices ist, daß es vollkommen trans-

parent ist, also keine Organisation der Daten vornimmt.

Da sich das AmigaDOS aber an Objekten, das heißt Dateien, orientiert, ist eine zusätzliche Stufe nötig, die eben diese Organisation vornimmt. Dazu dienen die Handler-Prozesse, die wiederum eine einheitliche Schnittstelle zwischen Anwendungsprogrammen und DOS bieten.

Diese Schnittstelle basiert auf der Kommunikation mittels er-

weiterten Exec-Nachrichten – den »Packets«. Ein »Packet« ist eine normale Exec-Message, deren Name-Feld auf eine Packet-Struktur zeigt, in der Befehle und Parameter abgelegt sind (siehe Abb. 1). Das Anwenderprogramm schickt diese Message nun an den Handler-Prozeß. Dieser führt den Befehl aus und schickt die Message wieder zurück. Diese Art der Kommunikation nennt man »asynchron«, da das Anwenderprogramm nicht gezwungen ist, auf die Beendigung der Arbeit zu warten. Denn diese wird vollkommen unabhängig vom Handler-Prozeß erledigt.

Reise in die Interna

Die in der Regel benutzten Funktionen der »dos.library« wie »Open()«, »Read()« und so weiter stellen nichts anderes als Hilfsfunktionen dar, die einen einfachen Umgang mit den »Packets« erlauben. Leider sind diese Funktionen recht unflexibel, da sie nur synchronen Zugriff erlauben. Das bedeutet, wenn man mit »Read()« einen 471 kByte langen Block von Diskette liest, muß das Hauptprogramm so lange warten.

Dies sollte nun kein Appell sein, in Zukunft die »böse« »dos.library« keines Blickes mehr zu würdigen und die gesamte Ein- und Ausgabe eines Programms nur noch durch direkte Kommunikation mit den Handlern durchzuführen. Denn der Aufwand dafür steht in keinem Verhältnis zum erreichbaren Vorteil. Wenn wir uns aber mit der »anderen« Seite des DOS beschäftigen wollen, führt an den »Packets«

kein Weg vorbei. Ein Handler tut nun nichts anderes, als auf ein »Packet« zu warten, das ihm von irgendwoher geschickt wird. Die darin enthaltenen Informationen werden ausgewertet, eventuell notwendige Aktionen getroffen und dann das Packet zurückgeschickt. Welche Funktionen man überhaupt implementiert, ist von der später beabsichtigten Funktion des Handlers abhängig. Absolut notwendig ist es aber, die Funktionen »Open (ACTION_FINDINPUT, ACTION_FINDOUTPUT, ACTION_FINDUPDATE)«, »Close (ACTION_END)«, »Read (ACTION_READ)« und »Write (ACTION_WRITE)« zu implementieren. Eine Aufstellung der wesentlichen Packet-Funktionen samt Parameter bietet Abb. 2.

Planspiel: Jokehandler

Kommen wir zu einem konkreten Projekt. Für unsere Zwecke soll ein Minimal-Handler mit den obengenannten Funktionen reichen (Handler, die auch Verzeichnisfunktionen und mehr bieten, werden allgemein als File-System bezeichnet). Damit es nicht so trocken ist, implementieren wir einen »JOKE-Handler«, der bei jedem Öffnen einen neuen Witz bereitstellt.

Das Listing »Misc« stellt die Grundfunktionen zum Umgang mit »Packets« zur Verfügung: »getpacket()« wartet auf und empfängt ein »Packet« vom angegebenen Message-Port, »returnpacket()« sendet das entsprechende »Packet« mit den angegebenen Parametern zurück.

Das Listing »Joke-Handler« ist der eigentliche Handler. Hier-

```
struct DosPacket {
    struct Message *dp_Link;      Zeiger zurueck auf die Exec-Message
    struct MsgPort *dp_Port;      ReplyPort fuer das Packet
    LONG dp_Type;                 Typ des Packets
    LONG dp_Res1;                 Rueckgabewerte
    LONG dp_Res2;                 ""
    LONG dp_Arg1;                 Parameter
    LONG dp_Arg2;                 ""
    LONG dp_Arg3;                 ""
    LONG dp_Arg4;                 ""
    LONG dp_Arg5;                 ""
    LONG dp_Arg6;                 ""
    LONG dp_Arg7;                 ""
};
```

Abb. 1. Die DOS-Packet-Struktur

zu einige Erläuterungen: Der Handler wird geladen, sobald er das erste Mal angesprochen wird. AmigaDOS schickt dann eine Startup-Message, die einige, für uns uninteressante Informationen enthält (zum Beispiel über den Namen des Handlers). Wichtig ist nur der Zeiger auf die »DevNode«, die AmigaDOS zur Identifikation jedes Handlers, Volumes (Diskette) oder logischen Geräts (Assigns) benutzt und die wir entsprechend initialisieren müssen.

Danach braucht der Handler nur noch auf ankommende Packets zu warten. Diese werden innerhalb der While-Schleife empfangen und ausgewertet, wobei zu den entsprechenden Funktionen verzweigt wird. Die erste Funktion ist »Open«, die gleich drei Einsprünge anbietet: »FINDUP-DATE«, »FINDINPUT«, »FINDOUTPUT«. Es handelt sich dabei um die verschiedenen Möglichkeiten zum Öffnen eines Files (entsprechen »MODE_NEWFILE«, »MODE_OLDFILE«, »MODE_READ-

»makefile«

```
OBJ = joke.o misc.o
l: joke-handler: $(OBJ)
blink $(OBJ) to l: joke-handler nd sc sd
misc.o: misc.c
lc -v -csmuq -b0 -rr -ms $(GO) misc
joke.o: joke.c
lc -v -csmuq -b0 -rr -ms $(GO) joke
```

WRITE«). Diese Unterscheidung ist eigentlich nur bei der Programmierung eines Filesystems von Belang und kann daher von unserem Handler ignoriert werden.

DOS setzt zur Verwaltung von Dateien die »FileHandles« ein (Abb. 3). Diese Struktur wird unserem Handler zwar fertig allokiert und belegt angeliefert, nichtsdestotrotz müssen wir einige Initialisierungen vornehmen. Das Feld »fh_Port« zeigt, so belegt, dem DOS an, daß es sich um eine interaktives File handelt. Das heißt, es ist möglich, mittels der Funktion »WaitForChar()« auf das Ankommen eines Zeichens zu warten. Da dies in unserem Fall reichlich überflüssig ist, wird das Feld gelöscht.

Die Einträge »fh_Arg1« und »fh_Arg2« stehen für den Handler zur freien Verfügung und dienen normalerweise zur internen Verkettung von geöffneten Dateien. Der Inhalt von »fh_Arg1« (und nicht der Zeiger auf den »FileHandle« selber) wird nämlich bei »Read« und »Write« an den Handler als Parameter übergeben. Um uns das Allokieren einer speziellen Struktur für jedes geöffnete File zu sparen, greifen wir ein wenig in die Trickkiste und lassen »fh_Arg1« auf den FileHandle selber und »fh_Arg2« auf den Text des gerade aktuellen Witzes zeigen.

Die nächste Funktion ist »Close (ACTION_END)«, die nichts weiter macht, als den Öffnungszähler um eins zu ver-

mindern und bei Null den Handler zu beenden.

Bei »Read« wird es dann interessant: Übergeben wird uns »fh_Arg1«, das (weil wir vorher so schön schlaue waren) auf den »FileHandle« zeigt und wir damit mittels »fh->fh_Arg2« den Zeiger auf den Witz bekommen. Mit »strlen()« läßt sich dann problemlos die Zahl der noch anstehenden Bytes feststellen. Verlangt das aufrufende Programm mehr als die zur Verfügung stehenden Bytes, wird einfach entsprechend gekürzt. Dann werden die Bytes in den übergebenen Puffer kopiert und das »Packet« mit der Anzahl der geschriebenen Bytes zurückgeschickt. Ist der Witz schon komplett gelesen, ist Länge gleich null, was einem »EOF« (End Of File) entspricht.

Witziges

Da die bisherige Version unseres Handlers das Schreiben eines Witzes noch nicht unterstützt (die Funktionen zum Prüfen auf Stubenreinheit und so weiter dürften den Rahmen dieses Artikels sprengen), wird ein entsprechendes Ersuchen mit einem Write-Protected-Fehler zurückgewiesen. Ist der Typ des angekommenen »Packets« nicht bekannt, schickt der Handler es unverrichteter Dinge per Action-not-known-Fehler zurück.

Damit ist die Funktionsbeschreibung des Handlers bereits beendet. Der Lattice-C-Source des Programms läßt sich dank des beigefügtem »makefile« (siehe Box »makefile«) problemlos mit LMK compilieren. Dann ist noch der entsprechende Eintrag mit:

```
JOKE:Handler = Ljoke-handler
Stacksize = 4096
Priority = 5
GlobVec = -1
```

#

in die Mountlist einzufügen, und schon kann der Handler mit

MOUNT JOKE:

angemeldet werden. Mit

COPY JOKE: RAM:witz

kann dann der Handler getestet werden - die Datei »witz« in der RAM-Disk sollte dann den ersten Witz der Liste enthalten. Theoretisch sollte es auch möglich sein, mit

TYPE JOKE:

Name	Wert	Funktion
ACTION_NIL	0	Keine
ACTION_SET_MAP	4	Keymap setzen
ACTION_DIE	5	Handler beenden
ACTION_LOCATE_OBJECT	8	Entspricht Lock()
ACTION_RENAME_DISK	9	Relabel, Disk umbenennen
ACTION_WRITE	'W'	Entspricht Write()
ACTION_READ	'R'	Entspricht Read()
ACTION_FREE_LOCK	15	Entspricht UnLock()
ACTION_DELETE_OBJECT	16	Entspricht Delete()
ACTION_RENAME_OBJECT	17	Entspricht Rename()
ACTION_MORE_CACHE	18	Addbuffers, Handler mehr Puffer zuweisen
ACTION_COPY_DIR	19	Entspricht DupLock()
ACTION_WAIT_CHAR	20	Entspricht WaitForChar()
ACTION_SET_PROTECT	21	Entspricht SetProtection()
ACTION_CREATE_DIR	22	Entspricht CreateDir()
ACTION_EXAMINE_OBJECT	23	Entspricht Examine()
ACTION_EXAMINE_NEXT	24	Entspricht ExNext()
ACTION_DISK_INFO	25	Entspricht Info()
ACTION_FLUSH	27	Puffer zurueckschreiben
ACTION_SET_COMMENT	28	Entspricht SetComment()
ACTION_PARENT	29	Entspricht ParrentDir()
ACTION_INHIBIT	31	Handler von Device aussperren
ACTION_DISK_CHANGE	33	Diskettenwechsel anzeigen
ACTION_SET_DATE	34	SetDate, Filedatum setzen
ACTION_SCREEN_MODE	994	Fuer CON:, CR/LF-Umschaltung
ACTION_SEEK	1008	Entspricht Seek()
ACTION_FINDUPDATE	1004	Entspricht Open("xxx",MODE_READWRITE)
ACTION_FINDINPUT	1005	Entspricht Open("xxx",MODE_OLDFILE)
ACTION_FINDOUTPUT	1006	Entspricht Open("xxx",MODE_NEWFILE)
ACTION_END	1007	Entspricht Close()

Die folgenden Packets sind nur fuer das FastFilesystem

ACTION_TRUNCATE	1022	File kuerzen
ACTION_WRITE_PROTECT	1023	Lock, Laufwerk schreibschuetzen

Abb. 2. Die gebräuchlichsten Packet-Typen

einen Witz direkt auszugeben. "Theoretisch" bedeutet, daß dies auch einwandfrei funktioniert, solange man den Original-Type-Befehl der Workbench benutzt. Das inzwischen sehr häufig verwendete ARP-Type hat nämlich einen hier sehr unangenehmen Bug und weist diesen Befehl mit dem Fehler »Can't open JOKE:« zurück.

Für Interessierte: Im Gegensatz zum Original-WB-Type ermöglicht das ARP-Type die Angabe eines Suchmusters wie beispielsweise

```
TYPE dh0:s/startup*
```

Die Suchmusterfunktionen von ARP benutzen allerdings die DOS-Funktionen »Examine()« und »ExNext()«, die wiederum auf die entsprechenden Packets zurückgreifen. Nun sind diese Packets in unserem Handler nicht implementiert, werden also mit »ERROR_ACTION_NOT_KNOWN« zurückgewiesen. »Type« prüft aber nur, ob die Suchmusterfunktion mit einem »ERROR_NO_MORE_ENTRIES« zu-

```
struct FileHandle {
    struct Message *fh_Link;      // Zeiger auf die dazugehörige Exec-Message
    struct MsgPort *fh_Port;      // Reply-Port fuer WaitForChar()
    struct MsgPort *fh_Type;      // Port des Handlers
    LONG fh_Buf;                  // Handler-intern, Puffer
    LONG fh_Pos;                  // Handler-intern, Position
    LONG fh_End;                  // Handler-intern, End-Position
    LONG fh_Funcs;                // Read-Funktion
#define fh_Func1 fh_Funcs
    LONG fh_Func2;                // Write-Funktion
    LONG fh_Func3;                // Close-Funktion
    LONG fh_Args;                 // Argument, wird an Read uebergeben
#define fh_Arg1 fh_Args
    LONG fh_Arg2;                 // ""
};
```

Abb. 3. Die Struktur »FileHandle«

rückkehrt, dem normalen Kennzeichen, daß das gesamte Verzeichnis durchsucht wurde. Jeder andere Fehler, so auch das von unserem Handler gelieferte »ACTION_NOT_KNOWN« wird als Fehler beim Öffnen der Datei interpretiert und entsprechend quittiert. Dieser Fehler tritt nicht nur beim »joke-handler« auf, auch das »AUX:« der Workbench 1.3 führt zu selbem Ergebnis. Nun ließe sich das Pro-

blem umgehen, wenn man die fehlenden Funktion »Examine()« und »ExNext()« einfach implementieren würde. Leider führt das dazu, daß man auch eine Lock-Verwaltung hinzuziehen müßte, was eindeutig in Richtung File-System geht und hier nicht weiter verfolgt werden soll.

Nun, wie bereits oben erwähnt, bietet das AmigaDOS durch seinen modularen Aufbau ungeahnte Erweiterungs-

möglichkeiten. So sind bereits Handler implementiert, die den Zugriff auf MS-DOS-Disketten erlauben. Für den Endanwender bietet eine solche Lösung den Vorteil, daß sie sich vollkommen in die gewohnte Arbeitsumgebung einbindet. Er kann so mit seinem Lieblings-Directory-Hilfsprogramm auch auf die MS-DOS-Disketten zugreifen, ohne vorher umständlich Dateien umkopieren zu müssen. (vb)

Biete Software

Verkaufe folgende Originale:
TV Sports Basketball für 55 DM,
Larry III für 60 DM, Turbo Print II für 50 DM, 3-D Pool Billard für 40 DM, Murder in Venice für 40 DM,
Emlyn Hughes International Soccer für 55 DM, Graffiti Man für 15 DM, Peter Beardsley International Soccer für 20 DM.
ZUSAMMEN FÜR NUR 250 DM!!!
Nur schriftlich an: Willy Bruckner
Oberer Gernauweg 7,
8200 Rosenheim

PD-Soft für AMIGA - PC - ATARI
Katalogdisk 1,50 DM in Briefm.
bei: Brunosoft/Schreiber
1000 Berlin 51, Sommerstr. 37 **G**

>>> **C-Komplettpaket** <<<
mit Compiler, Linker, EDITOR...
mit ANLEITUNG u. EINFÜHRUNG IN C.
IN DEUTSCH... = startbereit
zus. nur 20,- DM inkl. Porto bei
dPD, Nienort 46, 45 Osnabrück

Video-Manager, das Programm zur einfachen Verwaltung Ihrer Videocassetten - Optimale Ausnutzung der Cassetten durch Restlaufzeitkontrolle. Disk 15,- DM
TEL/Btx: 021 35-7 54 65,
ab 20 Uhr

BAVARIAN-Public-Domain, d.h. 180 Disketten fast nur deutscher PD. Gratisinfo bei Fr. Neuper, 8473 Pfreimd, Postfach 72 **G**

Erstellen Sie Ihre persönlichen PD-Disketten mit Programmen, die Sie auch wirklich brauchen
INFO bei: T. Eichhorn, Eisenkehlstr. 37, 6739 Weiden
G

Public Domain Disketten je 3,5" nur 1,50 DM inkl. Diskette.
Fordern Sie unsere Katalogdisketten gegen 5 DM an.
Doris Herberich, Hagerstr. 47
6960 Osterburken

VERKAUFE: MAXELL MF2-DD DISKS, 3 1/2" 20 STCK. FÜR 55 DM INKL. DISK-BOX!! UNBENUTZT! BEI: M. HOLM, 7564 FORBACH 4

Amiga-Originale, Super-Angebote, Sehr preisgünstig, 1A-Zustand, Viele mit PP-Prädikat - Wer hier nicht zugreift ist selber Schuld
Tel. 07 11/79 62 86

700R6. a50, z.B. Desert Flood; E.lut Soccer; Kick 0 + 2, Lead'B + Touru; 30 Tennis; Larry; a35; BeachVol; HighPatrol 2; Auarchg; HardDriv; a40; Chicago 90; Batman; Atomix; FullMetal Planet + Porto + neue usw. Tel.: 04 21/61 78 37

Verkaufe Sourcecodes (for Seka only!) Habe alles was interessant ist. (Demos + Co KG) 1 Disk 3.5" zu 4 DM West.Lieferie innerhalb 48 St. Keine Raubkopien!!! Liste gegen Rückporto bei: PLK 04 10 19D, D-7570 Baden-B.

Verkaufe PagesetterII - Original für DM 150,- zuzüglich Versandkosten. Jan Schmidt, Blumenweg 12, 7270 Nagold

Requester Constr. Tool R.C.T. 85 DM; Videoscape 3D Workshop (Buch + Disk) 35 DM; GFA-Basic auf dem Amiga (Buch) 30 DM; alles neu; jeweils 1 Original
Suche A4 Scanner 05 41-4 54 96

Biete Hardware

■■■ DAS DISK-CODER-MODUL. ES CODIERT JEDE DISK, SO DAS MAN DIE DISK OHNE MODUL NICHT MEHR LADEN/LESEN KANN. EXT. LAUFWERK NÖTIG, INFO GEGEN RÜCKPORTO! BARKI, BERGSTR. 218, 4370 MARL ■■■

Amiga 500 mit Monitor 1081, DISKETTEN UND HANDBÜCHER für 1000,- DM zu verkaufen
D. Grabowski, Gutenbergstr. 11, 4815 Schloß Holte

Verkaufe folgende Festplatten: von Quantum eine 40MB für 650,- DM und eine 80MB für 950,- DM. Verkaufe 48-1MB Rams das Stück für 14,50 DM
Tel. 030-6 18 44 57

Amiga 500 Kickstart 1.3 + 512 KB Speichererweiterung + Hardware-Uhr + Datamat + Textomat + Literatur. Alles Original verpackt!
DM 700,- Tel: 021 58/85 26

AMIGA 500 1MB + 2.Laufw. + Abdeckhaube + Monitor 1084S + PD-Software + viel Literatur wg. Zeitmangel zu verk. alles 1AZust. DM 1200.-
Tel. 083 02-7 21

A500 1MB GIGATRONKARTE MIT AKKUUEHR BIS 2,3MB AUFRÜSTBAR + SPIELE + Joystick wegen SYSTEMWECHSEL für 950,- DM zu Verk. NP. 1400,- DM, 3/4J. alt,
Tel.0 77 21/2 74 11 ab 18.00 UHR

Matrix-Drucker MELCHERS CP80x zu verkaufen. 250,- DM Anschlußfertig für Amiga. Bitte melden ab 18 Uhr unter 02 11/66 54 22, Oliver

XPRT-XT-Karte bis 12 MHz und Multi/J/O-Karte mit 640KB + 5 1/4Laufwerk DM 1000,- (zusammen), EGA-KARTE bis 800x600, VGA-AUFLÖSUNG DM 150,- TEL: 0 62 21-47 38 25

500KB-RAM-ERWEITERUNG INKL. UHR, ABSCHALTBAR, A-500 für DM 120. ZWEITLAUFWERK A-2000 INTERN DM 130. ZWEITLAUFWERK EXTERN DM 150. DATAPHON S21-23d, btxfähig DM 250. 0731/536 16 AB 17

A2000B im Tower, Big Agnus; 2/8MB, ALF2SCSI 80MB, Kick1.3 u 1.2, 2.LW, Ext.LW, 2000er Geh., 2.Tast, Div. orig. Soft, PC-XT-Karte 8MHz, Hercules-Karte + Mon. Multi-Sync, VB 5500 DM
Tel: ab 18h, 02 28/69 51 07

Suche Software

SUCHE GÜNSTIGE AMIGA-SOFTWARE. KEINE RAUBKOPIEN !!! ANGEBOTE AN RUDOLF GEIER, BEILNGRIESENSTR. 95, 8070 INGOLSTADT, 0841/59586 TEL/BTX

SUCHE GUTEN/BILLIGEN C-COMPILER, DEBUGGER + HANDBUCH ODER ANLEITUNG SASCHA KUCZIL FELDSTR. 8 5013 ELSENDORF 2

SUCHE AMATEURFUNK-SOFT + HARDWARE FÜR DEN AMIGA 500. H. C. MARKUS, POSTFACH 10 17 44, 4630 BOCHUM 1

Suche dt. Anleitg. zu Aztec-C3.6a, auch leihweise!
K. Sattler, Raimseser Str. 16, DDR, 4250 Eisleben

Suche PD zum Thema ASTROLOGIE. TIPS AN DAVID SCHWARZFISCHER, L.-THOMAS-STR. 36, 8066 UNTERBACHERN. TAUSCHE AUCH PD!

Suche Original LaserScript für PageSetter V1, Angebote an: Thomas Semmel, Consul-Vejento Str. 10, 6501 Klein-Winternheim

WER HILFT VIRUSARMEN CODER?? SUCHE ALLE(!) BOOT + LINKVIREN! DISK 100% ZURÜCK! (+ SPÄTER DEN VIRUSKILLER!) SEND'EM TO: S. WAGNER, DAHLIENWEG 15, 6382 FRIEDRICHSDORF

Suche Wargames für Amiga! nur Originale mit Anleitung! Angebote an Tel.06621-65175 Hans Georg Wegner Kurzer Weg 2 6430 Bad Hersfeld

CHARTTECH GESUCHT
TEL.0234-65817

Suche Hardware

MULTISYNC-MONITOR 14", FARBE, MÖGLICHST NEC ODER EIZO, ENTSPIEGELT FÜR AMIGA 2000 GESUCHT. W. BÜCHOLD, SAARLANDSTR.2, 8710 KITZINGEN, TEL. 09321/33856

AMIGA 2000B/C m. Mon. 1084 und HD, evtl. m. Zweitlaufwerk 3 1/2", weiteres Zubehör preisg. zu kaufen gesucht. Zuschr.an J. Hans, Ellerbreite 13, DDR, 4502 Dessau

Achtung!! Suche!! Achtung!! Festplatte für A500 m.40-60MB mit Geh./Anschl.-fertig!! Autoboot mögl. ab Kick1,2 Preisvorst. an: K. Stich Händelstr. 28, 3150 Peine/T12867

EPSON FX100 EINZELBLATTEINZUG GESUCHT, ZUSCHRIFTEN MIT PREIS AN KARL STOLLE SCHOLTEN STR. 36 4130 MOERS 1

SUCHE GEBRAUCHTE GENLOCKS ALLER ART. ANGEBOTE AN: THOMAS WINTER, HDH-STR.12, 7082 OBERKÖCHEN. BITTE BESCHREIBUNG BEILEGEN UND PREIS ANGEBEN

Verschiedenes

SUCHE ZWECKS BRIEFL. ERFAHRUNGSUSTAUSCHES KONTRAKT ZU HOBBYPROGRAMMIERERN (AMIGABASIC, HISOFTBASIC), ANDREAS BLUHM, DETMOLDERSTR. 439, 4800 BIELEFELD 1

Verk. AMIGA-MAGAZIN 4/88-akt. Ausg.: je 5 DM. AMIGA-BASIC für Profis: 69 DM. Programmierparxis AMIGA-BASIC: 49 DM. Programmieren mit A-BASIC: 49 DM. AMIGA-Sonderh. 3-10: je 10 DM. 02 03/85319 (Falk)

AMIGA-BILDERDIENST, Farbausdrucke in Fotoqualität auf Papier oder Folie m. Xeros-Tintenstrahl- o. Calcomp-Thermotransfer-Drucker DM 6,- (Papier DIN A4). Infos Tel. 0251/62214 (Druckmuster anfordern!) CGD Dr. Buddemeier, Schlesienstr. 40, 4400 Münster

HALLO AMIGANER! Suche SPIELPARTNER (f. FALCON, POPULUS, FIREPOWER ODER FLIGHT II) ÜBER MODEM (1200ER) RUFT AN 030/2153225 Mo-So 12-18 UHR

Biete Kostenlos an: DATAEX, AmigaSuice. Schicke einfach 1-2 leere Disketten + frankierten adressierten Rückumschlag an folgende Adresse: PLK 024895B, 4100 Duisburg 1 PD's dürfen geschickt werden

Wer verschenkt Lernprogramme für Amiga 500 z.B. Tippprogramm zum Lernen der Tastatur o. Einf. in CLI usw. Georg Geier, Obfling 9 1/2 8359 Schöllnach

PERRY RHODAN-FANS UND AMIGS-BESITZER: WER WILL BEI DER ENTWICKLUNG EINES DISK-MAGAZINS AUF PD-BASIS MITMACHEN? SCHREIBT AN: ERICH HERBST, SCHELLINGSTR. 101, 8000 MÜNCHEN 40.

KOMPLETTLÖSUNGEN A Lösungen aller Art. Für fast jedes Computerspiel. Liste anfordern - > 3,- DM. Für reg. Kunden Gratis. Jörg Kopmann Postfach 1616 - > 3110 Uelzen 1. Chiffre 90/00128 AMIGA

** Modula2-Fans aufgepaßt! ** Suche Kontakt zu Modula2-Usern, die wie ich nicht mehr alleine Modula2 lernen und programmieren wollen. Peter Weiland, Schubertstr. 60, 6300 Gießen

Wer hat Lust mit mir, Jürgen, 28 J., die Amigasprache C zu lernen? Aztec 5.0 und Literatur vorhanden! Du solltest aus dem Raum Witten (5810) sein! Tel. 02302/12521

Tausch

Tausche ASM-Ausgaben + Sonderhefte gegen Amiga Hard und Software sowie Literatur. Angebote nur schriftlich an Carsten Brendel, Brahmweg 9, 6500 Mainz 31

Suche PD-Tauschpartner. Habe ca. 500 Disks!! Auch Anfänger willkommen. Schickt Listen an: J. Erlewein, Kiem-Pauli-Weg 85, 8014 Neubiberg. Or call: 089/6014453. 1 DM Porto beilegen.

PUBLIC DOMAIN TAUSCHE + KAUF PD-DISKS SCHICKT MIR EURE KATALOG-DISKS, MEINE FOLGEN UMGEHEND, MARC PAHLENKÄMPER GLÜCKSBURGER STR.15, 4530 IBBENBÜREN

AMIGA-PD: WER TAUSCHT MIT MIR AMIGA-PD? BESITZE CA.200 DISKS UND BEKOMME BALD MEHR! SCHICKT EURE LISTE AN: POLDY, AM BARGFELD 22, 2072 BARGTEHEIDE. MEINE SERIEN: FF, RPD, EROTIK, BAVAR. USW.

SUCHE TAUSCHPARTNER FÜR AMIGA PD-SOFT UND SPIELE M. SCHARFENBERG AM HAMMOOR 16 2858 SCHIFFDORF 3

Suche Tauschpartner für PD und Seka/Devpac-Sources. KEINE RAUBKOPIEN! Schreibt an A. Signer, Büelacherweg 22, CH-8306 Brüttisellen, Tel. +041 1 833 12 78

Stellenmarkt

■ ■ TOP PROGRAMMIERER (ASSEMBLER) AUF AMIGA/ST GESUCHT! WIR (PROGRAMMIERER, GRAFIKER, MUSIKER) SUCHEN PROGRAMMIERER FÜR SPIELE, PROGRAMMIERUNG. MURAD M'BARKI, BERGSTR. 218, 4370 MARL

Geschäftsverbindungen

Wer kann meine Demo P.D. Serie "INTRO-VISIONS" im Ausland verteilen. Bitte nur zuverlässige Leute. Tel. 02421/76604

Das ist Ihre Chance...

Schon eine Kleinanzeige bringt oftmals großen Erfolg und hilft, neue Kontakte zu knüpfen. Nutzen Sie unser Angebot und profitieren Sie von der Tatsache, daß unsere Zeitschrift

»Amiga DOS«

jeden Monat von zigtausend Computer-Interessierten gelesen wird. Möchten Sie etwas verkaufen, tauschen, oder suchen Sie das »Tüpfelchen auf dem i« - dann sollten Sie eine Karte ausfüllen und an unseren Verlag absenden. Ihre Annonce erscheint dann in der nächsterreichbaren Ausgabe. Wir möchten ausdrücklich darauf hinweisen, daß wir keine Anzeigen veröffentlichen, aus denen ersichtlich ist, daß es sich hierbei um Veräußerungen von Raubkopien handelt.

Des weiteren machen wir darauf aufmerksam, daß indizierte Computerspiele nicht in Form von Anzeigen beworben werden dürfen.

Listings



```

1:  /*
2:  JOKE.c
3:
4:  Beispiel für einen DOS-Handler mit minimalem
5:  Funktionsumfang
6:
7:  by Oliver Wagner
8:  (c) 1990 DMV
9:  Sprache: Lattice C
10:
11: */
12:
13:
14: #include <proto/exec.h>
15: #include <proto/dos.h>
16: #include <string.h>
17: #include <libraries/filehandler.h>
18:
19: /*
20: Die "Jokes". Die Liste ist beliebig erweiterbar, m
21: uß aber mit einer 0 beendet werden
22: */
23:
24: char *jokes[]={
25:     "Der Tod ist die Methode, mit der Dir das Leben sa
26: gt, daß Du gefeuert bist...\n",
27:     "Alle vier Jahre wird mir immer so demokratisch um
28: 's Herz...\n",
29:     "Killroy wasn't here...\n",
30:     "God Save The Queen,8,1\n",
31:     "Ein Schnelli am Morgen bringt Kummer und Sorgen..
32: \n",
33:     "Ein Freak ist, wer nur noch 64 statt 69 will...\n
34: ",
35:     "Auf der Mauer, auf der Lauer, sitzt ein kleiner H
36: onecker...\n",
37:     "Liberte, Egalite, Kamillenteen",
38:     "Des Menschen Wille ist sein Amiga\n",
39:     "Two beer or not two beer this is the question\n",
40:     0};
41:
42: short jokecount=0;
43:
44: /* Protos der Routinen aus misc.c */
45:
46: extern void returnpacket(struct DosPacket*,struct Proc
47: ess*,long,long);
48: extern struct DosPacket *getpacket(struct Process*);
49:
50: /* Der eigentliche Handler */
51: void joke_handler(void)
52: {
53:     struct Process *hproc;
54:     struct DosPacket *packet;
55:     struct DeviceNode *devnode;
56:     struct FileHandle *fh;
57:
58:     short running=TRUE;
59:     long opencount=0;
60:     int jokelen,readlen;
61:     char *currentjoke;
62:
63:     /* Handler initialisieren, Startup-Packet annehmen */
64:     hproc=(struct Process*)FindTask(0);
65:     packet=getpacket(hproc);
66:     devnode=(struct DeviceNode*)BADDR(packet->dp_Arg3)
67: ;
68:     devnode->dn_Task=&hproc->pr_MsgPort;
69:
70:     /* und Startup-Packet zurückgeben - Damit läuft
71: der Handler
72: */
73:     returnpacket(packet,hproc,DOSTRUE,0);
74:
75:     /* Hauptschleife des Handlers */
76:     while(running) {
77:         packet = getpacket(hproc);
78:         switch(packet->dp_Type) {
79:
80:             /* "File" öffnen */
81:
82:             case ACTION_FINDUPDATE:
83:             case ACTION_FINDINPUT:
84:             case ACTION_FINDOUTPUT:
85:                 opencount++;
86:                 /* FileHandle-Struktur */
87:                 fh=(struct FileHandle*)BADDR(packet->dp_Arg1);
88:                 /* Nicht-Interaktives File */
89:                 fh->fh_Port=0;
90:                 /* wird bei Read usw. übergeben */
91:                 fh->fh_Args=(long)fh;
92:                 /* Zeiger auf den Joke */
93:                 fh->fh_Arg2=(long)jokes[jokecount++];
94:                 /* Falls Ende der Liste, neu von vorne */
95:                 if(!jokes[jokecount]) jokecount=0;
96:                 /* Packet zurückgeben */
97:                 returnpacket(packet,hproc,DOSTRUE,0);
98:                 break;
99:
100:             /* File schließen... */
101:
102:
103:

```

Listing Joke.C

```

104:         case ACTION_END:
105:             /* Falls kein "Öffner", Handler beenden */
106:             if(--opencount==0) running=0;
107:             returnpacket(packet,hproc,DOSTRUE,0);
108:             break;
109:
110:             /* Vom "Joke" lesen */
111:
112:             case ACTION_READ:
113:                 /* FileHandle, bei Open eingetragen */
114:                 fh=(struct FileHandle*)packet->dp_Arg1;
115:                 /* Zeiger auf den Joke des Handles */
116:                 currentjoke=(char*)fh->fh_Arg2;
117:                 /* Noch fehlende Zeichen Bis zum Ende */
118:                 jokelen=strlen(currentjoke);
119:                 /* Angeforderte Zeichen */
120:                 readlen=packet->dp_Arg3;
121:                 /* Falls mehr Zeichen angefordert, als lieferbar,
122: entsprechend kürzen */
123:                 if(readlen>jokelen) readlen=jokelen;
124:                 /* Text kopieren */
125:                 if(readlen) memcpy((char*)packet->dp_Arg2,currentjok
126: e,readlen);
127:                 /* Zeiger weitersetzen */
128:                 fh->fh_Arg2+=readlen;
129:                 /* Packet zurückschicken */
130:                 returnpacket(packet,hproc,readlen,0);
131:                 break;
132:
133:             /* Schreiben leider nicht möglich */
134:
135:             case ACTION_WRITE:
136:                 /* Schreibfehler */
137:                 returnpacket(packet,hproc,DOSFALSE,ERROR_DISK_WRITE_
138: PROTECTED);
139:                 break;
140:
141:             /* Unbekanntes Packet */
142:
143:             default:
144:                 returnpacket(packet,hproc,DOSFALSE,ERROR_ACTION_NOT_
145: KNOWN);
146:                 break;
147:         }
148:     }
149:
150:     /* DevNode löschen, Handler beendet */
151:     devnode->dn_Task=FALSE;
152: }
153: }

```

```

1:  /*
2:  MISC.c
3:  -----
4:  DOS-Packet-Routinen grob angelehnt an eine ältere
5:  Routinensammlung von Phillip Lindsay
6:
7:  diese Version von Oliver Wagner
8:  (c) 1990 DMV
9:  Sprache: Lattice C
10:
11: */
12:
13: #include <proto/exec.h>
14: #include <proto/dos.h>
15:
16: /* Packet zurückgeben, mit entsprechenden Returncodes
17: */
18: void returnpacket(struct DosPacket *packet, struct Pro
19: cess *p, long res1, long res2)
20: {
21:     struct Message *msg;
22:     struct MsgPort *replyport;
23:
24:     /* Return-Codes setzen */
25:     packet->dp_Res1=res1;
26:     packet->dp_Res2=res2;
27:     /* ReplyPort holen */
28:     replyport=packet->dp_Port;
29:     /* Zeiger auf die Exec-Message des Packets */
30:     msg=packet->dp_Link;
31:     /* Packet-Port zurücksetzen */
32:     packet->dp_Port=&p->pr_MsgPort;
33:     /* Message und Packet verbinden */
34:     msg->mn_Node.ln_Name=(char *)packet;
35:     msg->mn_Node.ln_Succ=NULL;
36:     msg->mn_Node.ln_Pred=NULL;
37:     /* und Message abschicken */
38:     PutMsg(replyport,msg);
39: }
40:
41: struct DosPacket *getpacket(struct Process *p)
42: {
43:     struct MsgPort *port;
44:     struct Message *msg;
45:
46:     /* Port unseres Process */
47:     port = &p->pr_MsgPort;
48:     /* Auf Nachricht warten */
49:     WaitPort(port);
50:     /* Nachricht abholen */
51:     msg = GetMsg(port);
52:     /* Packet extrahieren und zurückgeben */
53:     return((struct DosPacket *)msg->mn_Node.ln_Name);
54: }

```

Listing Joke.C

PD-WORKSHOP

Messefieber

Schaut man sich im Moment in den Redaktionen der verschiedensten Amiga-Zeitschriften um, sieht man nur noch hektische Redakteure/-innen durch die "geheiligten" Hallen rennen. Wo gibt es noch die allerheißesten Nachrichten zu ergattern, so daß man sich fast die Finger verbrennt? Auch unsere Redaktion blieb von diesem Fieber nicht verschont – nein, in hektischer Betriebsamkeit werden die letzten topaktuellen Meldungen in den Redaktionsräumen der AMIGA DOS von den zur Zeit allzu "geplagten" Redakteuren und -innen zusammengetragen. Und warum???

Schlagwort: Amiga'90 in Köln, der Welt größte Amiga-Messe, steht vor der Tür.

Auch den vielen, vielen PD-Händlern, die auch auf dieser Messe in Scharen vertreten sein werden, bricht so langsam aber sicher der Schweiß aus. Nicht vor Angst, eher vor der vielen Kopiererei, damit Ihnen als Publikum wieder viel geboten werden kann. Superprogramme zum kleinen Preis ist wie immer die Devise der seriösen PD-Händler. Auch die AMIGA DOS befindet sich mit einem eigenen Stand auf der Messe. Wir laden Sie herzlich ein, mit uns ein wenig Spaß in der heute manchmal so tristen Welt zu haben.

Besuchen Sie uns auf Stand 307 in Halle 12.

Viel Spaß bei der Lektüre dieses Workshops wünscht Ihnen

Vera Brinkmann

Vera Brinkmann
Redakteurin AMIGA DOS



Bezugsquellen

Die hier vorgestellten Programme wurden uns freundlicherweise von den Firmen

- A.P.S. Electronic
Sonnenborstel 31
3071 Steimbke
Tel.: 05026/1700

und

- Herrmanns & Kommelter
Vom-Bruck Platz 45
4150 Krefeld 1
Tel.: 02151/399833

zur Verfügung gestellt. Diese beiden Firmen haben die entsprechenden Disketten vorrätig. Weitere Vertrieber können Sie der Anbieterliste in der PD-Werkzeugkiste entnehmen.

Inhalt

Msh – MS-DOS-Daten für alle!

Offenes System	S. 86
Installation	S. 86
Hinweise	S. 86
Disketten und Bootblock	S. 86
Arbeit mit Msh	S. 87
Anderes Formatieren	S. 87
Diskettenname	S. 87

MyBBS – ein Zerberus-Netz-Point

Punkte verbinden	S. 88
Point, aber wie und woher?	S. 88
Session	S. 89

Schnapp zu!

Mit der Schere in den Screen	S. 90
Ready for takeoff	S. 90
Klickern mit der Maus	S. 90
Bilder klickern	S. 91
Schnappen mit Parametern	S. 91

Themen

Der PD-Workshop in der AMIGA DOS informiert Sie diesmal ausführlich zu den Programmen

■ »Msh« – ein Programm von Olaf Seibert – erleichtert Ihnen die Arbeit mit DOS-Formaten ganz erheblich.

■ »MyBBS« von Frank Schummertz ist ein Mailbox-Programm und fast in jeder Mailbox downzuladen.

■ »Snap« speichert den Bildschirm wie ein Notizblock und ist von Mikael Karlsson.

Literatur

Hinweise zu PD-Programmen finden Sie noch in folgenden Diskettenmagazinen:

- [1] AmigaJUICE, erscheint alle 4 bis 6 Wochen,
Herausgeber: Holger Lubitz,
Vertrieb: Herrmanns&Kommelter
- [2] GetiT, erscheint monatlich,
Herausgeber: B. Rönn,
Vertrieb: Herrmanns&Kommelter
- [3] AUGS, erscheint monatlich,
Herausgeber: Amiga User Group Switzerland,
Vertrieb: Herrmanns&Kommelter
- [4] Newsflash, erscheint alle 4 bis 6 Wochen,
Herausgeber: R. Fontaine,
Vertrieb: A.P.S. Electronic

Wer an seinem Arbeitsplatz an einem MS-DOS-Rechner arbeiten muß, der hat wohl schon des öfteren einen Wunsch verspürt: bestimmte Daten einfach mit nach Hause zu nehmen und dort auf seinem Amiga in aller Ruhe diese Daten zu bearbeiten und am nächsten Tag wieder auf den PC zu überspielen. Bislang gab es einige Hilfsprogramme, die PC-Disketten lesen und Dateien kopieren konnten. Doch waren das alles eigenständige Programme, so daß man Textdateien beispielsweise erst kopieren und umformatieren mußte, bevor sie in die heimische Textverarbeitung eingelesen werden konnten. Ein Beispiel für ein solches Programm ist »DOS-2-DOS« von Central Coast Software. Jetzt gibt es ein Shareware-Programm, das Sie in die Lage versetzt, MS-DOS-Disketten auf jedem Amiga mit den normalen CLI-Befehlen zu lesen, zu beschreiben, zu formatieren, zu löschen... — »Msh«.

Offenes System

Da der Amiga seine "offene Systemarchitektur" (OSA) jedoch nicht nur im Hardware-Bereich, sondern auch im Betriebssystem konsequent einhält, dürfte es eigentlich nicht unmöglich sein, einen Ersatz für das »trackdisk.device« zu schreiben, das statt dessen die Diskettenlaufwerke bedient und MS-DOS-Disketten lesen und beschreiben kann. Mit den Programmen »PC-Handler« von Mayer-Verlag und »Cross-Dos« von Consultron kamen solche Hilfsmittel in kommerzieller Form auf den Markt. Nachdem das hier vorgestellte und auf Fish-Disk 327 als Shareware veröffentlichte Programm »Msh« bei einem Wettbewerb eines bekannten Amiga-Magazins abgelehnt wurde, ist dieses Hilfsmittel nun auch kostengünstig für jeden Amiga-Benutzer zu haben. »Msh« verarbeitet MS-DOS-Disketten im 3,5-Zoll- oder 5,25-Zoll-Format — letzteres

Ottmar Röhrig

Msh — MS-DOS-Daten für alle!

Kommerzielle Programme wie »Cross-Dos« oder der »PC-Handler« haben sich des Problems schon angenommen, MS-DOS-Daten auf dem Amiga zu lesen.

natürlich nur, wenn Sie auch ein 5,25-Zoll-Laufwerk mit 40- (seltener) oder 80-Track (Standard) an den Amiga angeschlossen haben. Durch die hardwaremäßige Beschränkung, die die 3,5-Zoll-Laufwerke des Amiga auferlegen, sind zwar keine 1,44-Zoll-MByte-Disketten von OS/2 zu lesen, doch kommt das Programm zumindest mit den am meisten verbreiteten Formaten mit 8, 9 oder 10 Sektoren und 40 oder 80 Tracks zurecht. Für weitere Versionen von »Msh« ist das Lesen von Disketten von VAX- und PDP11-Minarechnern geplant. Erfreulicherweise wird jetzt schon die Arbeit mit MS-DOS-Harddisks unterstützt. Einzige Voraussetzung dafür ist, daß diese mit 12- oder 16-Bit-FATs arbeiten, was jedoch der Normalfall ist. FAT ist unter MS-DOS die sogenannte »FileAllocationTable«, die angibt, welche Dateien welche Sektoren der Diskette bereits belegt haben. Aus Sicherheitsgründen ist die FAT auf einer MS-DOS-Diskette oder -Festplatte meist mehrmals abgespeichert.

Installation

Die Bedienung von »Msh« ist bei all den Eigenschaften jedoch weitaus einfacher, als Sie vielleicht vermuten. Den größten Umstand wird wohl die Installation bereiten, deshalb

hier eine Schritt-für-Schritt-Anleitung für dieses Vorhaben. Als Voraussetzung gilt hier, daß Sie mit einer Originalkopie der Fish-Disk 327 arbeiten. Die Installation ist nur über den CLI ausführbar, kann jedoch von geübten Benutzern auch mit einem Dateikopierprogramm und einem Editor ausgeführt werden. Tabelle 1 gibt Aufschluß darüber, wo welche Dateien hingehören.

Damit sind alle Dateien an ihrem richtigen Platz. Sollten irgendwelche Fehlermeldungen auftauchen, so ist auf Ihrer Zieldiskette höchstwahrscheinlich nicht genügend Platz für die Programme, die insgesamt ungefähr 54 kByte beanspruchen. Löschen Sie nicht benötigte Dateien, und beginnen Sie die Installation von neuem.

Damit ist der Vorgang jedoch noch nicht beendet. Der sogenannten Mountlist, in der die Definition von neuen Geräten für das AmigaDOS abgelegt werden muß, muß nämlich noch ein Eintrag über das Gerät »MSH:« hinzugefügt werden. Laden Sie also die Datei »devs:mountlist« in Ihren bevorzugten Editor, und begeben Sie sich an das Ende der Datei. Dort tippen Sie die Zeilen aus Tabelle 2 ein und speichern den Text unter gleichem Namen als ASCII-Datei ab.

Hinweise

Beim Abtippen sind einige spezielle Bereiche zu beachten. So muß hinter »Unit« die Zahl stehen, die mit dem Laufwerk korrespondiert, auf dem MS-DOS-Disketten gelesen werden sollen (0 für das inter-

ne, 1 für das erste externe, 2 für das zweite externe, 3 für das dritte externe Laufwerk). Haben Sie keine Speichererweiterung, so sollte hinter »BufMemType« keine 1, sondern eine 3 stehen. Für die Spezialisten unter Ihnen: Die Handler und Device-Driver müssen auch nicht unbedingt in »L:« und »DEVS:« abgespeichert werden. Stehen sie jedoch in anderen Verzeichnissen, so muß der Eintrag hinter »FileSystem« und »Device« entsprechend angepaßt werden!

Auch der Gerätenamen kann von »MSH:« auf irgendeinen anderen Namen geändert werden, der jedoch mit einem Doppelpunkt enden muß. Falls Sie den Namen ändern, so muß im Laufe des Artikels anstatt »MSH:« überall der von Ihnen gewählte Name eingesetzt werden.

Disketten und Bootblock

Damit ist die Installation endlich abgeschlossen, und die Arbeit mit MS-DOS-Disketten kann beginnen. Eine Voraussetzung müssen die Disketten übrigens erfüllen: Sie müssen mindestens mit der MS-DOS-Version 3.1 formatiert worden sein! Das kommt daher, daß das MS-DOS-Betriebssystem erst ab dieser Version einen Bootblock mit Informationen über das Format der Diskette schreibt. Die Einträge »LowCyl«, »Surfaces« und »BlocksPerTrack« in der Mountlist werden nämlich nur beim ersten Zugriff auf die Diskette benötigt, um den Bootblock zu finden, der die tatsächlichen Informationen über den Aufbau der Diskette enthält.

Das ist im Übrigen der Grund, warum auch mit dem Eintrag »HighCyl=79«, der die Anzahl der Spuren auf 80 festlegt, Disketten mit nur 40 Tracks einwandfrei gelesen werden können — ähnlich verhält es sich mit den »BlocksPerTrack«. Ähnlich nur deshalb, weil es wichtig ist, genau die richtige Anzahl Sektoren pro Track dort anzugeben, wenn Sie mit dem CLI-Befehl »DiskCopy« eine Diskette im MS-DOS-Format kopieren möchten. »DiskCopy« liest nämlich verständlicherweise nicht erst den Bootblock einer Diskette, um sich über

- 1) copy AmigaLibDisk327/Msh/c c:
- 2) copy AmigaLibDisk327/Msh/l l:
- 3) copy AmigaLibDisk327/Msh/devs/messydisk.device devs:

Tabelle 1. Anweisung, in welches Verzeichnis die Dateien zu kopieren sind

deren Aufbau zu informieren, da ihm ja normalerweise die Informationen in der Mountlist genügen.

Nachdem alle Vorbereitungen getroffen sind, können Sie das "neue" MS-DOS-Laufwerk mit dem Befehl

»mount msh:«

anmelden. Wird nun eine Diskette in das Laufwerk gelegt, untersucht der Amiga automatisch, ob es sich um eine AmigaDOS-Diskette handelt (dann wird die Kontrolle an das »trackdisk.device« weitergegeben) oder ob es eine MS-DOS-Diskette ist (dann wird die Diskette von »messydisk.device« bearbeitet), was die Meldung »DFx:BAD« auf der Workbench hervorruft. Diese ist jedoch normal und verschwindet, sobald Sie die Diskette wieder aus dem Laufwerk nehmen.

Arbeit mit »Msh«

Legen Sie eine MS-DOS-Diskette ein, so kann auf ihr ganz normal gearbeitet werden. Dateien kopieren mit »copy«, löschen mit »delete«, anschauen mit »type«, Verzeichnisse erstellen mit »makedir«, wechseln mit »cd« und so weiter. Die Workbench ermöglicht bei Dateien die Extension »info«. Da diese Extensionen aber bei Laufwerken, die unter »Msh« benutzt werden, nicht erlaubt sind, ist es nicht empfehlenswert, die Workbench unter »Msh« laufen zu lassen. Auch von jeder beliebigen Applikation kann das MS-DOS-Laufwerk direkt angesprochen werden, indem man beispielsweise statt »DH0:« einfach »MSH:«

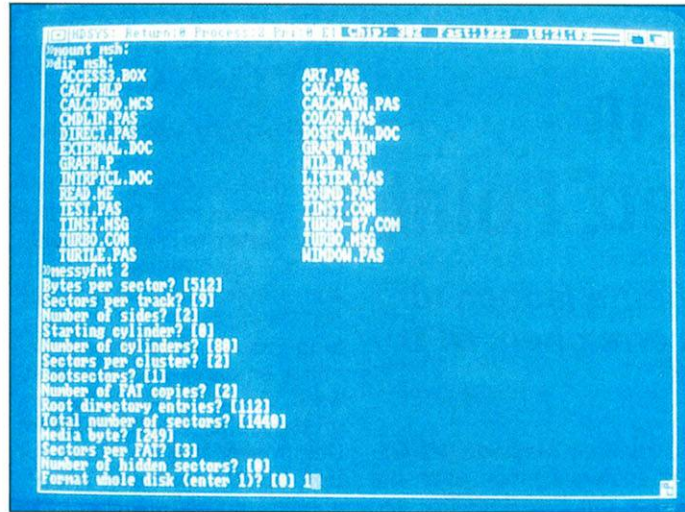


Bild 1. Das Programm »messyfmt« im Einsatz

eingibt. Lediglich zur Formatierung von MS-DOS-Disketten auf einem Amiga ist nicht das normale »format« des CLI zu verwenden. Ein spezielles Programm »messyfmt« übernimmt diese Aufgabe. Es wird mit der Syntax

`messyfmt <Unit> [DeviceName]`

aufgerufen. Als »Unit« muß die Nummer des Laufwerkes eingesetzt werden, ebenso wie schon beim Mountlist-Eintrag »Unit«. Als »DeviceName« kann, muß aber nicht, ein anderer Name als »messydisk.device« stehen. Normalerweise wird jedoch kein zweiter Parameter verwendet.

Anderes Formatieren

Nun fragt »messyfmt« nach und nach einige Parameter ab, die meistens mit [Return] bestätigt werden können – das übernimmt die in Klammern

angegebene Zahl (siehe dazu auch Bild 1). Die Frage »Format whole disk?« bezieht sich darauf, ob die ganze Diskette oder nur das Root-Verzeichnis und die FAT formatiert werden sollen. Soll der Formatierungsvorgang tatsächlich gestartet werden, so ist die letzte Frage mit »42« zu beantworten (Arthur Dent läßt grüßen...).

Zwei weitere Hilfsprogramme sind für die Verwaltung von MS-DOS-Laufwerken gedacht. »die« läßt ein entsprechendes Laufwerk mit als Parameter folgendem Namen »sterben«, meldet es also ab (um es vollkommen abzumelden, ist jedoch noch ein »assign MSH: remove« notwendig). »ignore« mit darauffolgender Unit-Nummer unterdrückt die ständige Überprüfung von CRC-Checksummen, was es unter Umständen noch ermöglicht, defekte Dateien zu lesen.

Auf Folgendes müssen Sie übrigens bei der Arbeit mit MS-DOS-Disketten noch achten: Dateinamen sind anders aufgebaut als unter AmigaDOS. Grundsätzlich gibt es nur Großbuchstaben, die zudem nur 8 Zeichen plus eine mit einem Punkt getrennte 3-Zeichen-Erweiterung lang sein dürfen. Außerdem dürfen die Zeichen »Leerzeichen«, »*«, »?«, »<«, »>«, »/«, »=«, »|«, »:«, »\« und »<« nicht vorkommen. Sollte dies aus Versehen doch der Fall sein, so ändert »messydisk.device« den Dateinamen automatisch, so daß eigentlich nie etwas schiefgehen kann.

MS-DOS-Disketten tragen übrigens ein sogenanntes »Label«, das den Diskettenamen

angibt. Dieses Label wird normalerweise beim Formatieren mit angegeben, was jedoch mit »messyfmt« nicht möglich ist. Deshalb sei es unbedingt angeraten, nach dem Formatieren – nachdem die Diskette sowieso aus dem Laufwerk genommen und wieder eingelegt werden muß – den Befehl »relabel« zu verwenden, um der Diskette einen eigenen Namen und eine Uhrzeit zu geben. Wird das nicht gemacht, so erhält die Diskette automatisch den Namen »Unnamed« und das Erstellungsdatum 01-Jan-80. In Verbindung mit »Msh« können beispielsweise auch versteckte Dateien angezeigt werden. Daneben lassen sich Datei-Attribute und das eben erwähnte Erstellungsdatum einer Diskette einfach mit den CLI-Befehlen »protect« und »setdate« ändern.

Diskettenname

Im Vertriebs-Paket von »Msh« wird übrigens der komplette Sourcecode in C mitgeliefert. Sollten Sie Programmierer sein, so können Sie das Programm auch auf ganz spezielle Bedürfnisse anpassen – eine spezielle Anleitung mit dem Namen »dev.man« ist ausschließlich für Entwickler geschrieben und berichtet ausführlich über die Implementation und die Funktionen des »messydisk.device«. Damit dürfte der Arbeit mit diesem guten Utility nichts mehr im Wege stehen.

Weitere Informationen liefert ein recht umfangreiches Referenz-Manual, das auch auf der Diskette zu finden ist und viele weitere Probleme lösen kann. Mit diesem Utility wird dem Anwender ein Werkzeug an die Hand gegeben, das Qualität zum kleinen Preis liefert.

(vb)

```
/* Mountlist-Eintrag für ein MS-DOS-Laufwerk */
MSH: FileSystem = I:MessyFileSystem
Device = messydisk.device
Unit = 2
Flags = 0
LowCyl = 0
HighCyl = 79
Surfaces = 2
BlocksPerTrack = 9
Buffers = 5
BufMemType = 1
BootPri = 0
Stacksize = 4096
Priority = 5
GlobVec = -1
#
```

Tabelle 2. Dieser Eintrag wird in der Datei »devs:mountlist« vorgenommen und als ASCII-Datei abgespeichert

PD-Info

Programm: Msh
Merkmal: Shareware
Autor: Olaf Seibert
 Beek 5
 5815 CS Merselo
 Niederlande
Quelle: Fish 327

Oliver Wagner

MyBBS – Ein Zerberus-Netz Point

Früher umständlich, heute jedoch für nahezu jedermann zu erreichen. DFÜ wird populär.

In den Ursprüngen der DFÜ-Zeit war jede Mailbox ein einzelnes System, abgeschottet vom Rest der Welt. Wer überregionale Informationen sammeln wollte, mußte Mailboxen außerhalb seines Nahbereichs anrufen – bei 300 Baud eine sehr kostenintensive Methode.

Irgendwann schlossen sich dann verschiedene Mailboxen zu Datennetzen zusammen. Um einen überregionalen Datenaustausch anzubieten, tauschen die Mailboxen die Bretter untereinander aus: Schreibt ein Benutzer eine Nachricht in einem Netzbrett, ist diese am nächsten Tag in jeder dem Netz angeschlossenen Box zu lesen.

Durch diese Methode des Datentransfers können auch die Kosten der Box-Betreiber niedrig gehalten werden, da ja jede Box ihren "Tauschpartner" mit etwas Glück im Nahbereich oder Fernzone 1 hat. Auch betreiben Sysops Boxen meist mit High-Speed-Modems, die zwar unerreich hohe Datentransferraten, aber auch ebenso hohe Anschaffungskosten als Merkmal aufweisen.

Dieser Informationsfluß hat aber auch einen Nachteil: Die Zeit, die jeder User »online« in der Box verbringt, um sämtliche Nachrichten zu lesen und zu beantworten, steigt immer mehr an. So kostet dies dem Enduser wieder mehr Telefon-

gebühren und die Box ist unnötig lange von einem Einzelnen belegt.

Punkte verbinden

Aus diesem Grund entstand das System der Points: Der User bestellt bei seiner Box die Bretter, die ihn interessieren und erhält diese auf Anfrage als Datenpaket. Zur Auswertung dieses Datenpakets benötigt er dann die Point-Software. Diese Software besteht grob gesehen aus zwei Teilen: Dem Mailer, der die Daten zur Box schickt und dort wieder abholt, und dem Nachrichteneditor, der im Prinzip wie eine Box funktioniert, aber eben nicht »online«, sondern nur mit den per Point-Paket angekommenen Daten. Der Vorteil ist, daß der User nur noch für die Zeit des Mailens »online« ist. Dabei geht dieser Vorgang vollkommen automatisch und so mit maximaler Geschwindigkeit vor sich. In der Regel kommt man, je nach Anzahl der bestellten Bretter, immer mit einer Gebühreneinheit aus.

Unter DFÜ-Freaks gehört das Zerberus-Netz zu den beliebtesten Datennetzen überhaupt und wird von vielen Mailboxen

angeboten. So gibt es auch für den Amiga eine Z-Netz-Point-Software: »MyBBS« von Frank Schummert. Das Paket ist Shareware und kann seit kurzem in praktisch jeder Z-Netz-Box downgeloadet werden.

Point, aber wie und woher?

Der erste Schritt auf dem Weg zum eigenen Point besteht in einer Bitte an den Sysop der Box, einen solchen Point für den zukünftigen Benutzer einzurichten. Der Sysop möchte dann den Namen des Points (ist später belanglos), das Passwort und das verwendete Archivierungsprogramm wissen. (Auf dem Amiga sollte man nach Möglichkeit immer ARC/ARC benutzen). Als User stellt man dann noch in der Box seinen Point als Vertreter ein, damit auch persönliche Nachrichten den Weg in den Point finden. Heißt der Point zum Beispiel "OLLI", ist als Vertreter "SYSOP@OLLI.ZER" einzutragen.

Damit sind die externen Arbeiten bereits abgeschlossen. Als nächstes muß man sich die Software installieren. Festplattenbesitzer erzeugen ein Unterverzeichnis mit Namen »ZERBERUS« und fügen die Zeile

ASSIGN ZERBERUS: DHO:ZERBERUS

in ihre »s:start-up-sequence« ein. Diskettenbenutzer formatieren einfach eine Diskette mit dem Namen »ZERBERUS«.

»MyBBS« wird auf die Diskette beziehungsweise in dieses Unterverzeichnis kopiert. Das MyBBS-Paket besteht insgesamt aus den fünf Programmen »Mailer«, »Import«, »Export«, »Recover«, »XPRTTransmit«, einer Bibliothek »xprzmodem.library« sowie verschiedenen Konfigurationsdateien (»config.zer«, »areas.zer«, »modem.cfg«). Die Bibliothek ist selbstverständlich in das Verzeichnis »Libs« der Bootdiskette bzw. Bootpartition zu kopieren.

Zusätzlich muß man sich noch die Archivierungsprogramme »ARC« und »PKXARC« besorgen, die aber weit verbreitet und praktisch überall downzuloaden sind. »ARC« ist auch auf Fish 70 zu finden. Die meisten DFÜ-Freaks werden diese Programme sowieso schon längst haben.

```
MyBBS V1.5c                                     © by Frank Schummert 13
8      8219 T 08.06 G.HOOFFACKER@INFIN Morde in Peru (bis 13.7.)
9      6520 T 08.06 G.HOOFFACKER@INFIN Mutter von Ermordeten bedroht
10     1907 T 10.06 BILBO@HOT.ZER      re: Tja, Bilbo, das war wohl nix!
11     2432 T 11.06 K.FISCHER@KBBS.ZER WIDERSTAND GEGEN BONNER GROßMACHTKURS
12     1766 T 08.06 D.LOEBNER@LINK-HH. re^x:
13      322 T 08.06 M.WOYDT@LINK-HH.ZE rere spd verlassen
14      576 T 11.06 RO1%BFC@TBX.ZER    A: re: Lafontaine
15    18329 T 11.06 RO1%BFC@TBX.ZER    RE:WOHNUNGSNOT->ENDENER
Keine weiteren Nachrichten gefunden
```

/Z-NETZ/POLITIK:> BRE ZAA

/Z-NETZ/AMIGA/ALLGEMEIN:> I *

Neue Nachrichten in Brett /Z-NETZ/AMIGA/ALLGEMEIN :

Nr.	Byte	Typ	Datum	Absender	Betreff
1	278	T	09.06	OLIVER WAGNER	re: Amiga Point Prg.
2	7256	T	09.06	OLIVER WAGNER	The Musical Enlightenment
3	1345	T	08.06	4-TEA-2@MULI.ZER	Re:re: Re:Amiga-PC
4	629	T	10.06	OLIVER WAGNER	re: Re:re: Re:Amiga-PC
5	361	T	17.09	MARC SCHROEER@TB. itoa()	
6	324	T	09.06	O.BARTHEL@A-LINK-H	MSH & ST-Disketten
7	446	T	09.06	C.SCHREIER@INFINET	Re:Re:Fish-Disks
8	342	T	08.06	TOMCAT@BDB.ZER	Re:Re:RE: CMI & Co

<<< Abbruch >>>

/Z-NETZ/AMIGA/ALLGEMEIN:> █

Informationen lassen sich über Bretter mit »MyBBS« abrufen


```
; Name des Sysops, für private Post
OLIVER_WAGNER=Zerberus:Privat
; Verzeichnis für unzustellbare Nachrichten
/UNZUSTELLBAR=Zerberus:Unz
; Die Verzeichnisse für die einzelnen Bretter
/Z-NETZ/AMIGA/ALLGEMEIN=Zerberus:ZAA
/Z-NETZ/AMIGA/PROGRAMMIEREN=Zerberus:ZAP
/Z-NETZ/AMIGA/BINAER=Zerberus:ZAB
/Z-NETZ/AMIGA/VIREN=Zerberus:ZAV
/Z-NETZ/POLITIK=Zerberus:ZP
```

Tabelle 1. Beispiel für eine Datei »\UNZUSTELLBAR«

Dann geht es an die Konfiguration. In der Datei »areas.zer« trägt man die bestellten Brett-namen mit den dazugehörigen Unterverzeichnissen ein. Daneben sind zwei besondere Bretter zu erwähnen: Eines trägt den Namen des Sysops, dort wird private Post eingelagert. In dem anderen mit Namen »/UNZUSTELLBAR« legt die Software jede Nachricht ab, die unzustellbar ist, das heißt, die in keine andere Rubrik eingeordnet werden konnte. Ein Beispiel für eine solche Datei können Sie Tabelle 1 entnehmen. Natürlich muß man die angegebenen Unterverzeichnisse auch mit »Makedir« anlegen. Die Datei »modem.cfg« enthält Informatio-

nen über das verwendete Modem. In der Regel sind nur die Felder »SERVER« (Name der Serverbox), »POINT« (Name des eigenen Points), »PASSWORD« und »DIAL« (»atdp« mit Nummer der Serverbox) zu ändern. In »config.zer« trägt man die Konfiguration für das Mailboxprogramm »MyBBS« ein. Wie eine Beispiel-Konfiguration aussehen könnte, entnehmen Sie bitte Tabelle 2. Natürlich bietet »MyBBS« noch wesentlich mehr Konfigurationsmöglichkeiten, auf die in diesem Beispiel aber nicht eingegangen werden soll. Sie sind einfach nicht wichtig genug. Im Zweifelsfall sei immer auf die Anleitung zum Programm verwiesen.

```
; Farben
FRONT=240
BACK=0
ROUTE=JA
; der zu verwendende Editor
EDITOR=ed
; Text, der beim simulierten "Logon" ausgegeben wird
LOGON=Zerberus:logon.txt
; Wird vom Nachrichteneditor an jede Nachricht angefügt
ORIGIN=Olli's Point ^O^ Kein Kohl, kein Krenz, kein Benz
; Name des Points
POINT=OLLI
; Name der Serverbox
SERVER=TTB
; Umlautwandlung
UMLAUTE=1
```

Tabelle 2. Die Datei »config.zer« könnte so aussehen

Noch ein Wort zum Editor: Prinzipiell kann jeder Editor benutzt werden, der sich vom CLI aus starten läßt. Dabei darf sich das Programm aber nicht vom CLI abkoppeln, so daß beispielsweise das Programm »Cygnus-Ed« nicht mit »MyBBS« funktioniert. Als letztes muß noch die Datei »areas.index« erzeugt werden. Das funktioniert recht einfach durch den Aufruf des Programms »RECOVER«. Fertig ist die Installation.

Session

Eine Z-Netz-Session läuft dann wie folgt ab: Man ruft das Programm »MAILER« auf, das automatisch die Box aufruft, die geschriebenen Nachrichten abschickt und neue abholt. Diese Nachrichten werden dann mit Aufruf von »IMPORT-a« (-a steht für ARC als verwendeter Archiver) entpackt und einsortiert. Mit »MyBBS« kann man dann in den Nachrichten nach Lust und Laune stöbern, beantworten und selber welche schreiben. Das Programm funktioniert genau wie eine Z-Netz-Mailbox, so daß hierzu keine weiteren Erläuterungen notwendig sind. Danach werden die neugeschriebenen Nachrichten mit »EXPORT-a« für den Versand fertig gemacht und beim nächsten Mailen abgeschickt. Hat man Nachrichten gelöscht, muß man noch »RECOVER« zum Neusortieren der Marken aufrufen. Soweit, so wunderbar. Wie kann man aber nun die einzelnen Bretter bestellen und wieder abbestellen? Dazu gibt es in jeder Serverbox einen simulierten User namens »MAPS«. Im »Betreff« einer Nachricht an diesen User wird der Befehl angegeben, in der Nachricht selbst eventuelle Parameter. Eine genaue Liste aller Befehle erhält man, wenn man eine

Nachricht mit »Betreff HILFE *« an »MAPS« schickt. Bretter bestellt man mit dem Befehl »ADD«, wobei im Nachrichtentext die zu bestellenden Bretter angegeben werden: »/Z-NETZ/AMIGA*« bestellt beispielsweise alle Amiga-Bretter, »/Z-NETZ/POLITIK« das Politik-Brett. Der User muß allerdings selbst darauf achten, ob er alle bestellten Bretter auch bei »areas.zer« eingetragen hat, ansonsten landen die Nachrichten im Brett »/UNZUSTELLBAR«. Analog zu »ADD« bestellt »DEL« Bretter wieder ab. Weitere MAPS-Befehle findet man im oben erwähnten Hilfsfile. Beachten Sie bitte, das »MyBBS« Shareware ist. Falls Sie also einen Point mit der Software installiert haben, erwartet der Autor Frank Schummertz eine Sharegebühr von 30,- DM. In Anbetracht der Leistungsfähigkeit des Systems ist dieser Preis bestimmt nicht zu hoch gegriffen. Außerdem spart man diese Summe durch die gewaltig gesunkenen Telefonkosten schnell wieder ein. (vb)

PD-Info

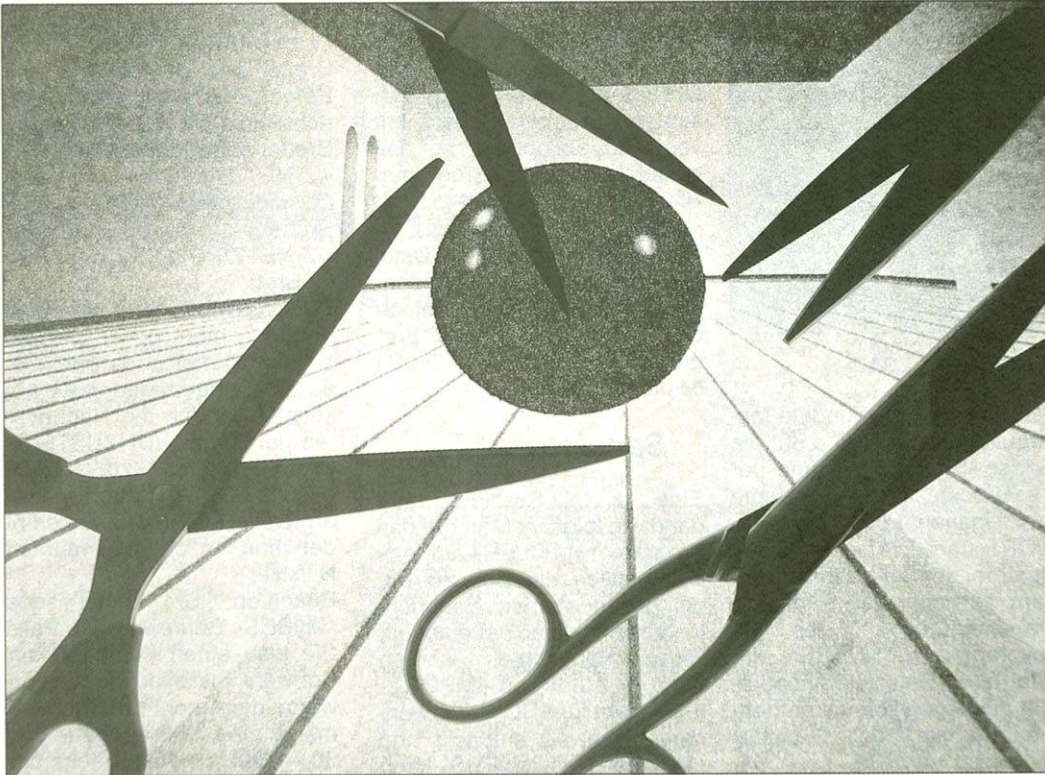
Programm: MyBBS

Merkmal: Shareware

Der Autor erwartet einen Obolus von 30,- DM als Sharegebühr.

Autor: Frank Schummertz

Quelle: Kann in praktisch jeder Z-Netz-Box downgeloadet werden



Ottmar Röhrig

Schnapp zu!



“Was ich gerade auf dem Bildschirm sehe, ist genau das, was ich gleich tippen werde...”

Warum sollte man etwas, was schon auf dem Bildschirm zu sehen ist, noch einmal eintippen?

Diese Frage führte den Autor des Programms »Snap« zur Entwicklung des nun in der Version 1.4 vorliegenden Hilfsmittels. Zu der genannten Situation kommt es beispielsweise bei jedem, der über bestimmte Anwendungen am Computer schreibt und Bildschirm-Ausschnitte (Text oder Grafik) in seinen Text mit aufnehmen möchte. So wird man beispielsweise gezwungen, die Ausgabe eines CLI-Befehls in die Textverarbeitung erneut einzutippen. Auch kann auf diese Art und Weise eine (zuvor bereits eingegebene) lange Kommandozeile einfach in einem anderem CLI-Fenster ausführen lassen, ohne daß man Sie noch einmal extra eingeben muß. In diesem CLI-Fenster kann sie ja nicht mittels der Taste [Cursor-Hoch]

wiederholt werden, da jedes CLI seinen eigenen History-Buffer besitzt.

Mit der Schere in den Screen?

Ein anderes Anwendungsgebiet ist ein Text, eine Fehlermeldung oder ein einfacher Grafikausschnitt, von dem Sie wissen, daß er später noch benötigt wird. Bisher mußte man einen solchen Bereich abschreiben oder abmalen. Mit »Snap« kann der Bereich markiert werden und ist ab sofort als eigenes Fenster auf der Workbench zu sehen. Unabhängig davon, wie sich der markierte Bereich daraufhin verändert, wirkt das Fenster als eine Art Notizblock, der die Grafik oder den Text immer zur Zeit des Schnapp-Vorgangs darstellt. Für die spätere Archivierung kann ein solcher Ausschnitt auch als IFF-Grafik abgespeichert werden.

Lediglich eine Vorarbeit ist für die Arbeit mit »Snap« zu leisten: die Datei »clipboard.device« muß im Verzeichnis »Devs:« vorhanden sein. Bei der Installation des Programms müssen Sie nur die Datei »Snap« in das von Ihnen gewünschte Verzeichnis ihrer Arbeitsdiskette kopieren. Das Programm wird vom CLI mittels der Eingabe **Snap** gestartet und entkoppelt sich automatisch vom aufrufenden CLI-Fenster, weshalb ein Run oder Runback nicht notwendig ist. Beenden können Sie »Snap« durch »Snap -Q« oder »Snap QUIT«

Nach dem Programmstart steuern verschiedene Tasten das Programm und seine Funktionen. Es gibt grundsätzlich zwei verschiedene Arten, einen Bereich zu »snappen«. Entweder möchte man einen Grafikausschnitt »merken« oder einen Text in ein anderes Programm übertragen. Beim

»Textschnappen« kommt ausschließlich die linke Amiga-Taste ins Spiel. Drücken Sie diese versuchsweise einmal und halten Sie sie fest. Nun fährt man mit der Maus auf das erste zu erkennende Zeichen und drückt die linke Maustaste. Ohne diese loszulassen, kann die Maus nun bewegt werden, was einen rechteckigen Bereich mit einem sich bewegenden Rahmen aus kleinen Strichen öffnet.

Ready for takeoff

Alles, was sich innerhalb dieses Rahmens befindet, wird von »Snap« in einen Zwischenspeicher (den sogenannten Buffer) übertragen, sobald man die linke Maustaste losläßt. Haben Sie es sich während des Markierungsvorgangs anders überlegt und wollen nun doch keinen Text »snappen«, so können Sie auch erst die linke Amiga-Taste und dann die linke Maustaste loslassen, was den Erkennungsvorgang von »Snap« abbricht und den Inhalt des Buffers nicht berührt. Vorteilhaft, wenn Sie sich beispielsweise gerade erinnern, daß im Buffer noch ein Text stehen müßte, der noch gebraucht wird. Ist jedoch die Markierung erfolgreich beendet worden, aktivieren Sie beispielsweise das Fenster Ihrer Textverarbeitung, in die der eben geschnappte Text eingegeben werden soll. Nun einfach die linke Amiga-Taste und [I] gedrückt, ein wenig gewartet und schon ist der Text wieder zu sehen! »Snap« fügt übrigens (bis auf die letzte Zeile) automatisch nach jeder Zeile ein [Return] ein, damit die Formatierung gewahrt bleibt. Auf Wunsch kann auch automatisch ein String an den Anfang und das Ende jeder Zeile eingefügt werden. Wie dieser String aussieht, wird mit den weiter unten erläuterten Parametern definiert. Um den Textbuffer auf diese Art modifiziert auszugeben, muß neben der linken Amiga-Taste auch noch die Taste [Shift] gedrückt werden.

Klickern mit der Maus

Da es jedoch unzählige verschiedene Anwendungsbereiche für die Texterkennung mit

»Snap« gibt, haben Sie drei weitere Möglichkeiten, den Beginn eines Bereichs zu markieren. Die Vorgehensweisen sind dabei immer dieselben: linke Amiga-Taste während der gesamten Operation gedrückt halten und währenddessen mit der Maus einen Bereich markieren. Doch ist die Form des markierten Bereichs bei jeder dieser Möglichkeiten anders. Klickt man in dem oben beschriebenen Vorgang jedoch nicht nur ein-, sondern zweimal (beim zweiten Klick jedoch nicht loslassen, sondern festhalten und in Ruhe die Maus bewegen, bis der Bereich korrekt markiert ist) mit der Maus, wenn man sich gerade über dem Anfang des zu schnappenden Bereichs befindet, so wird automatisch der Wort-orientierte Modus aktiviert. Hierbei muß der zu markierende Bereich nicht unbedingt rechteckig sein, sondern kann in einer beliebigen Zeile und Spalte anfangen, über mehrere Zeilen hinweggehen und wiederum in einer anderen Spalte aufhören. All dies geschieht jedoch Wort-orientiert, der markierte Bereich wird also niemals mitten in einem Wort aufhören.

Ähnlich wie das Wort-orientierte Schnappen funktioniert das Zeichen-orientierte. Hierzu ist die linke Amiga-Taste zu drücken, ein kurzer Mausklick auf den Anfang des Bereiches anzubringen und die Markierung mit wiederum gedrückter linker Maustaste zu speichern. Die letzte Alternative zur Bereichsmarkierung besteht im zeilenorientierten Schnappen. Diese ähnelt sehr der zuerst besprochenen Variante, mit dem Unterschied, daß es sich beim markierten Bereich immer um ein Rechteck mit der Breite des aktiven Fensters handelt. Für diesen Markierungsmodus muß nach dem Druck auf die linke Amiga-Taste dreimal mit der linken Maustaste auf den Anfang des Bereichs geklickt werden, bevor man mit gedrückter Maustaste den Bereich umrahmen kann.

Um jedoch auch noch eine zweite Möglichkeit für das Ausgeben des Textbuffers zu haben (erstere ist der gleichzeitige Druck auf die linke Amiga-Taste und [I]), kann man auch die linke Amiga-Taste drücken und währenddessen die rechte

Maustaste klicken. Auch hier ist jedoch darauf zu achten, daß das Fenster, in das der Text vom Computer »eingegen« werden soll, zuvor mit der Maus aktiviert wurde.

Bilder klicken

Möchten Sie sich nicht auf Text beschränken, sondern jeden Bildschirmbereich zwischen speichern, so sind Sie auf die Grafikschnapp-Funktion von »Snap« angewiesen. Eine Grafik kann natürlich nicht als Eingabe an ein Programm weitergegeben werden, außer Sie speichern sie als IFF-Grafik ab und laden diese in einem Malprogramm wieder ein. Doch kann es nützlich sein, sich Bereiche beliebiger Bildschirme (also nicht nur des Workbench-Bildschirms) zu merken und auf Mausklick zur Neuansicht zur Verfügung zu haben. Das Grafikschnappen funktioniert ausschließlich mit der rechten Amiga-Taste und ist auch ansonsten nicht so umfangreich wie das Textschnappen. Einfach die rechte Amiga-Taste drücken und festhalten, mit gedrückter linker Maustaste den Grafikbereich markieren und die Maustaste danach loslassen. Auch hier kann der Vorgang abgebrochen werden, wenn Sie zuerst die rechte Amiga-Taste loslassen. Ist die Markierung abgeschlossen, erscheint ein Fenster auf dem Workbench-Bildschirm, der als Inhalt den markierten Bereich aufweist. Dieses Fenster erscheint immer auf der Workbench, egal auf welchem Intuition-Screen die Grafik markiert wurde. Unterschei-

den sich die Farbpaletten der Workbench und des Original-Bildschirms, von dem die Grafik ursprünglich herstammte, so kann diese zeitweise angeglichen werden. Drücken Sie einfach innerhalb des aktivierten Grafik-Fensters von »Snap« die linke Maustaste und halten Sie sie fest. Solch ein – nennen wir es einmal »Notiz-Fenster« – kann wie gewöhnlich durch einen Klick auf dessen Schließ-Gadget wieder geschlossen werden. Erst wenn alle Notiz-Fenster geschlossen sind, kann »Snap« auch auf die anfangs beschriebene Art wieder aus dem Speicher entfernt, also beendet werden.

Innerhalb der Titelzeile des Notiz-Fensters befindet sich jedoch noch ein weiteres Gadget, das recht ungewöhnlich ist und mittels einer kleinen Diskette repräsentiert wird. Es ist direkt links neben den Tiefen-Gadgets angebracht und ist zuständig für das Abspeichern des angezeigten Grafikausschnitts als IFF-Grafik. Klickt man einmal darauf, öffnet sich ein neues Kontroll-Fenster von »Snap«. Dort kann der Dateiname eingegeben werden, unter dem die Grafik abgespeichert werden soll. Dies geschieht, wenn man entweder ein weiteres Mal auf das Disk-Gadget klickt oder das Save-Gadget des Kontroll-Fensters anwählt. Ist dieses Fenster bereits geöffnet (was auch durch die Tastenkombination linke Amiga-Taste und [W] geschehen kann), erscheint es natürlich nicht noch einmal.

Die Art und Weise, in der der Dateiname im Kontroll-Fenster

geändert werden kann, ist übrigens davon abhängig, ob »Snap« beim Start die Funktionsbibliothek »arp.library« im Verzeichnis »Libs:« gefunden hat. Ist das der Fall, öffnet sich bei einem Klick auf das Dateinamen-Gadget ein Datei-Requester, der die Eingabe des Dateinamens erlaubt. Wurde die »arp.library« von »Snap« nicht gefunden (diese Library ist Public Domain und auf vielen PD-Disketten und in Amiga-Mailboxen zu finden), ist das Dateinamen-Gadget im Kontrollfenster ein einfaches String-Gadget, in das ein Text eingegeben werden kann. Dies sollte natürlich hier eine gültige AmigaDOS-Pfad/Name-Kombination sein. Das Gadget »Transparent Color« gibt die Zahl des Farbregisters an, das als Hintergrundfarbe abgespeichert werden soll. Fast immer ist dies 0, muß also nicht geändert werden.

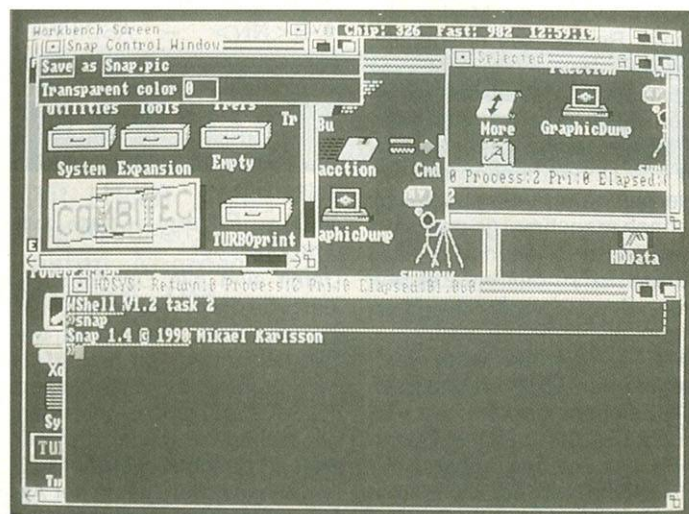
Schnappen mit Parametern

Dem Programm können verschiedene Parameter gleich zu Beginn mit auf den Weg gegeben werden. Mit diesen kann jeder Benutzer »Snap« auf die eigenen Bedürfnisse hin konfigurieren. Die Syntax von »Snap« lautet:

```
Snap -p<NN> -t<XX>
-g<XX> -i<XX> -w<XX>
-P<str> -A<str> -c<NN>
-l<NN> -a< > -x -X -E
-u<N> -r -R -j -b<XX>
-o<NN> -C<NN>
-B<NN> -Q
```

wobei <NN>/<N> eine dezimale und <XX>/< > eine hexadezimale Zahl repräsentieren; <str> steht für String und bezeichnet eine Zeichenkette beliebiger Länge. All das sieht auf den ersten Blick viel komplizierter aus, als es tatsächlich ist. Wir werden uns jeden einzelnen Parameter getrennt vornehmen.

• -p<NN> – Hiermit wird die Priorität des Input-Handlers gesetzt. Sie beträgt normalerweise 51 und ist somit um eins höher, als der normale Input-Handler von Intuition. Meist ist dies ausreichend. Zeigen jedoch bestimmte Tastenkombinationen keine Wirkung, so wird diese Kombination vom Input-Handler eines anderen Programms abgefangen. Da-



Bilder lassen sich im IFF-Format speichern

her kann es notwendig sein, die Priorität zu erhöhen.

●-t<XX> – Der Qualifier-Code der Taste, die das Textschnappen einleitet. Voreingestellt ist der Wert 0040, was der linken Amiga-Taste entspricht.

●-g<XX> – Der Qualifier-Code der Taste, die das Grafik-schnappen einleitet. Voreingestellt ist der Wert 0080, was der rechten Amiga-Taste entspricht.

●-i<XX> – Der RAW-Key-Code der Taste, die zusammen mit der linken Amiga-Taste den Inhalt des Textbuffers wieder ausgibt. Voreingestellt ist der Wert 17, was der Taste [I] entspricht. Andere Werte sind 34 für die Taste [V] oder 15 für [Z].

●-w<XX> – Der RAW-Key-Code der Taste, die zusammen mit der linken Amiga-Taste das Kontroll-Fenster von »Snap« öffnet. Normalerweise 11, was [W] entspricht.

●-P<str> – Die Zeichenkette, die bei der modifizierten Ausgabe des Textbuffers (wenn [Shift] gleichzeitig zur linken Amiga-Taste und [I] gedrückt wird) jeder Zeile vorangestellt wird. Voreingestellt ist »>«.

●-A<str> – Die Zeichenkette, die jeder Zeile aus dem Textbuffer nachgestellt wird (bevor das obligatorische [Return] eingefügt wird), wenn Sie eine modifizierte Ausgabe des Textbuffers anstrengen.

●-c<NN> – Zeit in Mikrosekunden, die »Snap« zwischen jedem Buchstaben wartet, wenn der Textbuffer ausgegeben wird. Dies kann notwendig sein, wenn das den Text empfangende Programm die eingehenden Daten nicht schnell genug verarbeiten kann. Voreingestellt ist keine Wartezeit, also ein Wert von 0.

●-l<NN> – Wie das vorherige Beispiel, nur wird hier die Zeit in Mikrosekunden angegeben, die hinter jedem [Return] von »Snap« gewartet werden soll.

●-a<> – Eine hexadezimale 16-Bit-Zahl, die das Muster des sich bewegenden Rahmens angibt, den »Snap« während jeder Markierung anzeigt. Abgeschaltet wird die Bewegung mit dem Wert FFFF (alle Pixel immer an). Voreingestellt ist ein Wert von 7777. Setzt man den Wert gleich 0000, so verwendet »Snap« für jede der vier oben erwähnten Markie-

rungsmethoden einen unterschiedlich aussehenden Rahmen.

●-x – Mit diesem Schalter wird das Verhalten von »Snap« etwas verändert. Von nun an wird der geschnappte Text direkt nach Loslassen der linken Maustaste in das zuvor aktivierte Fenster ausgegeben. Das heißt, man muß beispielsweise die Textverarbeitung vorher aktivieren, bevor man im CLI die Ausgabe eines CLI-Befehls markiert, die in den Text übernommen werden soll. Dieses Verhalten ist von Computern der Firma Xerox übernommen worden.

●-X – Schaltet das Xerox-ähnliche Schnappen des Textes wieder aus.

●-e – Fängt Programmroutinen, die für »Snap« gefährlich werden könnten, automatisch ab, solange die Textschnapp-Taste (normalerweise linke Amiga-Taste) gedrückt ist. Diese Option wird automatisch eingeschaltet, wenn der Parameter »-x« gegeben wurde.

●-E – Der Gegenbefehl zu »-e« schaltet die Patch-Routine wieder ab. Dies ist die Voreinstellung.

●-j – Mit diesem Parameter veranlassen Sie »Snap« beim Textschnappen am Ende jeder Zeile das [Return] wegzulassen, wenn das letzte Zeichen der ersten Zeile kein Leerzeichen ist. Dies trifft jedoch nur zu, wenn der Text nicht mit einer rechteckigen Markierung definiert wurde.

●-J – Schaltet das Verhalten von »-j« wieder ab. Voreingestellt.

●-u<N> – <N> ist entweder 0 oder 1. Voreingestellt ist 0, was beim Textschnappen als ersten Modus die Markierung eines Rechteckes ermöglicht. Wird 1 angegeben, hat man beim ersten Mausklick gleich den Zeichen-orientierten Markierungsmodus vorliegen.

●-r – Unterstrichene Zeichen werden näher untersucht, was jedoch manchmal (je nach verwendetem Zeichensatz) dazu führen kann, daß ein Zeichen von »Snap« falsch oder überhaupt nicht erkannt wird.

●-R – Werden beim Textschnappen auch unterstrichene Zeichen markiert, so versucht »Snap« nach der Option »-R« die Zeichen anhand der Pixel oberhalb der Unterstreichung zu erkennen. Dies ist bei einigen Zeichensätzen not-

wendig, da die Unterstreichungsroutine des Betriebssystem-Fehler aufweist.

●-b<XX> – Ist die Bitplane-Maske, die angibt, in welcher Farbe der Rahmen beim Textschnappen gezeichnet wird. Voreingestellt ist der Wert 1, bei 2 wird der Rahmen schwarz, bei 3 orange (bezogen auf ein Textschnappen auf der Workbench mit den Standard-Farben).

●-o<NN> – Gibt die Anzahl der Pixel vom rechten oberen Rand jedes Notiz-Fenster an, ab der das Disk-Gadget gezeichnet wird. Voreingestellt sind 52 Pixel, die das Gadget direkt neben den Tiefen-Gadgets anordnen. Ein anderer Wert kann notwendig sein, wenn man andere Programme gleichzeitig laufen läßt, die ebenfalls eigene Gadgets an den Rahmen eines Fensters anbringen.

●-C<NN> – Ist die Anzahl der Fenster, bei denen sich »Snap« merkt, daß aus Ihnen bereits Text geschnappt wurde. Dies ist insofern wichtig, da Sie normalerweise das Textschnappen nur auf einem Zeichen, nicht auf einem Leerzeichen beginnen können. Wurde jedoch bereits Text erfolgreich aus einem Fenster geschnappt, so kann der Markierungsvorgang beim zweiten Mal auch auf einem Leerzeichen erfolgen. Voreingestellt ist der Wert 10, was für »Snap« bedeutet, sich die Werte für die letzten 10 Fenster – aus denen Text geschnappt wurde – zu merken. Wird dieser Parameter verändert, während »Snap« bereits läuft, so erhöht sich die Bufferzahl um den angegebenen Wert.

●-B<NN> – ASCII-Wert des Zeichens, das eingesetzt wird, wenn »Snap« beim Textschnappen ein Zeichen nicht erkennen kann. Voreingestellt ist 63, was dem Fragezeichen entspricht.

●-Q – Entfernt »Snap« aus dem Speicher.

Damit sind alle Optionen erklärt, die die persönliche Konfiguration von »Snap« für die Fortgeschrittenen unter Ihnen problemlos ermöglichen sollten. Doch gibt es noch einige weitere Punkte, die bei »Snap« beachtet werden sollten. Diese sollen hier in der Art einer Liste dargestellt werden, da sie oft Ursache für das

Nicht-Starten bestimmter Funktionen sind:

1. Nachdem »Snap« gestartet wurde, kann es jederzeit erneut mit neuen Parametern aufgerufen werden. Dabei ändern sich nur die Einstellungen mit den explizit angegebenen Parametern. Es wird also nicht alles auf die Voreinstellung zurückgesetzt.

2. »Snap« arbeitet mit allen nicht-proportionalen Zeichensätzen, die nicht höher oder breiter als 16 Pixel sind. Es wird also nicht nur der Topaz-8-Font der Workbench korrekt gehandhabt.

3. »Snap« erkennt beim Textschnappen zwar auch Zeichen, die invertiert, fett oder unterstrichen sind. Bei Zeichen in Kursiv-Schrift oder bei Zeichen, die nicht in der normalen Farbe dargestellt sind, verweigert »Snap« die Arbeit.

4. Außer den unter dem Parameter »-C<NN>« aufgeführten Ausnahmen, sollte das Textschnappen immer auf einem Zeichen, nicht auf einem Leerzeichen erfolgen, da »Snap« ansonsten keine korrekten Koordinaten der Zeichen berechnen kann.

5. »Snap« unterstützt alle eingestellten Tastaturbelegungen.

6. Hat man neben »Snap« auch nach das PD-Hilfsprogramm MachII laufen, so darf der Parameter »-e« nicht eingegeben werden!

7. »Snap« unterstützt auch »AREXX«. Über die Möglichkeiten informiert ein Extra-Teil der Anleitung sowie ein mitgeliefertes Beispielprogramm.

8. »Snap« ist nicht PD, sondern Shareware. Verwenden Sie es häufiger, so unterstützen Sie den Shareware-Gedanken durch Überweisung eines finanziellen Obolus an den Programmierer, dessen Adresse Sie in der Anleitungsdatei finden.

(vb)

PD-Info

Programm: Snap
Merkmal: Shareware
Der Autor erwartet einen Obolus von \$10.
Autor: Mikael Karlsson
Quelle: Fish 326



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	6S	1.490,- (DM 213,-)
Amiga 5.25 Zoll Laufwerk, Bus/Ein- u. Ausschalter, 40/80 Tr.	6S	1.990,- (DM 285,-)
Amiga 3.5 Zoll Laufwerk, Bus/Ein- u. Ausschalter	6S	1.490,- (DM 213,-)
Amiga 2000, 2 MB Memory	6S	4.990,- (DM 713,-)
Amiga 2000 XT-Karte inkl. Laufwerk	6S	5.990,- (DM 856,-)
Amiga 2000 XT-Turbo mit 8 MHz	6S	1.590,- (DM 227,-)
Amiga Ersatzmaus	6S	590,- (DM 85,-)
Control-Center 500 (Top-Styling!)	6S	1.490,- (DM 213,-)
Digi View 4.0 Gold	6S	2.690,- (DM 385,-)
Genlock Pal Ver 1.3	6S	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

Druckfehler und Preisänderungen vorbehalten.



**AMIGA -- BTX
MULTITERM
!!! PRO !!!**

Der absolute BTX Software-Decoder.

**Kopierschutz passe', jetzt kommt:
Project D**

Disketten Utility der Sonderklasse

Das Original Kopierprogramm direkt aus USA.
Mit Update-Service.

Preis:		
Multiterm an A-Koppler / Modern	125,-DM	Incl. deutscher Anleitung
Multiterm Pro	149,-DM	
Multiterm an Post-Modern DBT03	195,-DM	DIGI-View 4.0
Multiterm Pro	229,-DM	Pagestream 1.8
Sep. Interface Amiga-DBT-03 nur	75,-DM	Pagesetter II
		weitere Software zu Superpreisen !!!!
		Kickstart ROM 1.3

Das richtige Modem !!!!

Best 2400 Plus (Btx-fähig)*	429,-DM	Neuheiten !!!
Discovery 2400C*	359,-DM	Interne PC-Karte f. Amiga 500
U.S. Robotics HST 14.400*	1560,-DM	Interne 20 MB Festplatte
U.S. Robotics Dual Standard*	2390,-DM	
Weitere Modelle Robotics/Lightspeed etc. auf Anfrage		Für XT / AT-Karten ...
* Betrieb am Postnetz der BRD und West-Berlin bei Strafe verboten		Flugsimulator 4.0
512K Speicher mit Uhr f. A500	189,-DM	Die neueste Version von Microsoft
Info über MS-DOS Rechner und weitere Neuheiten anfordern.		Direkt aus USA.

M. Kirschbaum Medienberatung

Schubertstr. 3, 4320 Hattingen
Tel. + Btx: 02324/82249 --- Fax: 02324/83722



Inh.
P.Engels

512K Ram für A-500 139,50 DM
Mega-Bit-Technologie, Accu-Uhr, abschaltbar

Disklaufwerk extern 179,00 DM
Qualitätslaufwerke, abschaltbar & durchgeführter Bus

128K RAM-Karte für XT-Karte: 178,00 DM
kurze Karte mit 128K Ram. Somit haben Sie 640K DosRam und können z.B. Ventura, F&A und D-Base-IV anwenden

XT-Multifunktions Karte: 299,00 DM
Multi-IO für XT-Karte mit 128K Ram Ser. Par. & Uhr

A2000 ALF2 Autoboot Filecards:

20MB:888,-DM, 30MB:968,-DM, 50MB:1268,-DM

A500 ALF2 Autoboot Festplatten:

20MB:1068,-DM, 30 MB:1168,-DM, 40MB:1448,-DM

File-Cards für XT & AT-Karte:

30 MB : 648,-DM, 50 MB : 948,-DM

Fordern Sie unsere kostenlose Preisliste an

SKY-Ware, Postf.1225, 5358 Bad Münstereifel
Telefon: 02253/2667 (btx)

Wir produzieren Bestseller

Amiga 500

**Speichererweiterung bis 1,8 MB
akkugepufferte Uhr
abschaltbar, Industriequalität**

512 KB + Uhr für nur DM 129.-

Externes Laufwerk für alle Amigas DM 179.-

Tröps + Hierl Computertechnik GmbH * 5040 Brühl * Jordanstr.3

Amiga 2000

**EPEX der Epromexpress
Im Amiga Magazin
6/89 getestet**

komplett inkl. Software DM 348.-

Tel.: 02232/45018



NORDSOFT die Amiga Profis!!!!

Ihr Amiga Profi für:

- * Reparaturen
- * Versand/PD
- * Entwicklung
- * Ersatzteile

2800 Bremen 21
Heidelbergstr. 75
Tel.: 0421-611430
Fax: 0421-6166912

NEU! NEU! NEU!
Boing Mouse(optisch)
3 Tasten, Microschalter
Mousepad 198,-

2 MB Aufrüstsatz
Supra 288,-

Jin Mouse (optical) 149,-
Jin Mouse normal 99,-

NEU! XCOPY-PRO 99,-

VGA-Karte für XT/AT-Karte, Auflösung 800 x 600 249,-

Drucker Fujitsu DL-1100
Testsieger! 1098,-

2MB Aufrüstsatz für
A590! 288,-

2 MB Supra Ram für
A2000 689,-

AMIGA 500 incl.
Appetizer 789,-!

Public-Domain !
** 5000 Disks! **

**Katalogdisk gegen
5 DM anfordern!**

Besuchen Sie uns
auf der AMIGA 90!
Halle 10 Stand
Nr. 1305 !!

Matthias Kühn

Schau an!

Das Programm »look«, das wir Ihnen an dieser Stelle präsentieren, ist ein Ersatz für das bekannte Programm »more«.

Jedoch kann »look« eine ganze Menge mehr als »more«. Zum einen scrollt der Text bei »look« auf- und abwärts, was die Übersicht über den Text wesentlich verbessert. Auch ganz- und halbseitiges Scrollen wird von »look« beherrscht, und mit einem Proportional-Gadget an der Seite weiß man genau, wo im Text man sich befindet. Außerdem läßt sich mit ihm auch schnell eine bestimmte Stelle im Text anwählen. Suchen nach Texten mit oder ohne Groß- oder Kleinschreibungsunterscheidung ist mit »look« keine Schwierigkeit.

Jedoch ist die herausragendste Fähigkeit von »look«, daß man sich damit alle Files, egal ob es sich um Text oder um binäre Dateien handelt, ansehen kann. »look« filtert nämlich alle Zeichen, die nicht durch seinen Filter passen, heraus und zeigt nur die Zeichen, die man sehen will. »look« läßt einem die Wahl zwischen fünf Filtern:

- 1.) alle Zeichen, die keine Steuerzeichen sind,
- 2.) Zahlen
- 3.) Buchstaben
- 4.) Zahlen und Buchstaben
- 5.) ASCII-Zeichen

Wenn einem diese Filter nicht genügen, kann man sich leicht weitere programmieren, da

alle Filter eigenständige Routinen sind, auf die mit einem Zeiger verwiesen wird. Weiterhin kann man bei »look« einstellen, daß erst ab einer bestimmten Anzahl von hintereinander aufgetretenen gefilterten Zeichen diese gezeigt werden, das heißt, gibt man als Zahl eine Drei ein, werden nur Worte länger als zwei Zeichen angezeigt. Dies ist dann nützlich, wenn man sich eine binäre Datei ansieht, in der man zum Beispiel nach Lösungen für Spiele, Paßwörter, Fehlermeldungen, Meldungen von Linkviren oder Hinweise zur Benutzung sucht. Mit »look« bleibt einem kein Text mehr verborgen.

Die Bedienung des Programms

Das Programm selbst ist recht einfach zu handhaben. Eine Online-Hilfe kann man durch Druck der Taste [Help] aufrufen, falls man mal was vergessen hat. »look« wird nur über die Tastatur bedient, da keine komplizierten Eingaben zu machen sind. Die Cursor-Tasten [auf] und [ab] bewegen den Text jeweils eine Zeile hoch oder runter, mit gedrückter Taste [Alt] eine halbe Seite, mit gedrückter Taste [Ctrl] eine

ganze Seite. Drückt man [Cursor up] und [Shift], springt der Text an den Anfang, bei [Cursor down] und [Shift] ans Ende. Beim Drücken der Taste [S] wird ein kleines Fenster mit einem aktivierten String-Gadget geöffnet, in das man den zu suchenden Text eingibt. Nach der Eingabe wird der Text ab der obersten Zeile der Anzeige gesucht und, falls er gefunden wurde, auch in der obersten Zeile angezeigt; wurde er nicht gefunden, wird dies in einem »Windowtitle« gemeldet. Bei jedem Drücken der Taste [N] wird nun nach weiteren Stellen gesucht, an denen dieser Text steht. Das Drücken der Taste [I] bewirkt das gleiche wie die Taste [S], nur daß nun zwischen Groß- und Kleinschreibung kein Unterschied mehr gemacht wird. Damit die Umgewöhnung von »more« auf »look« nicht zu schwer fällt, wird nach Drücken der Leertaste eine Seite vorwärts, nach Drücken der Taste [Backspace] eine Seite rückwärts gescrollt. Die Taste [Return] läßt die Anzeige um eine Zeile nach oben gehen. Jetzt fehlt nur noch die Taste [Q], mit der man das Programm beenden kann, dies ist aber auch mit dem Close-Gadget des Fensters möglich. Gestartet wird »look« vom CLI aus, dabei hat die Kommandozeile folgendes Aussehen:

```
Look Source [TO Destination]
[#saden]
```

»Source« ist die Datei, die angezeigt werden soll, und »Destination« ist die Datei, in die der gefilterte Text geschrieben wird. Bei Angabe einer Destination wird kein Fenster geöffnet, ansonsten ist das Fenster so groß wie der Workbench-Screen. Im Titel steht der Name der Quelldatei, damit man weiß, was man sich gerade ansieht. Die Optionen wer-

den mit einem »-« eingeleitet und können danach in beliebiger Reihenfolge angegeben werden. Das #-Zeichen bedeutet eine einstellige Zahl, und erst beim Auftreten von mindesten ebenso vielen gefilterten Zeichen werden diese ausgegeben.

Bei Angabe von »s« wird der ASCII-Filter angeschaltet, »a« schaltet den Buchstabenfilter ein und »d« den Nummernfilter. Bei Angabe mehrere Filter wird die Gesamtmenge der Filter gebildet, zum Beispiel werden bei »ad« sowohl Buchstaben als auch Zahlen zugelassen. Das »c« steht für ein beliebiges Zeichen, welches ausgegeben wird, wenn ein nicht gültiges Zeichen auftritt. Damit wird eine Trennung von Texten deutlich, die in der Anzeige zwar direkt hintereinander stehen, in der Datei aber getrennt sind. Die Angabe von »n« bewirkt, daß Return-Codes nicht mehr berücksichtigt werden, dadurch paßt mehr Text in das Fenster.

Hinweise zum Compilieren von »look«

Das Programm wurde mit dem Lattice C Compiler Version 5.04 entwickelt und kann mit jedem ANSI-Compiler compiliert werden.

Da »Aztec C« aber leider noch nicht ANSI-kompatibel ist, bleibt nur der Compiler von Lattice. Compiliert und gelinkt wird mit folgender Eingabe:

```
lc -cuf rr -rr -v -O Look
blink FROM LIB:c.o+Look.o TO
Look LIB LIB:lcr.lib
```

Danach wird »look« im Verzeichnis »C« untergebracht. So, das war's; Tschüß und viel Spaß mit »look«.

(jb)

Listings

```
1: /*****
2:
3: :Programm. Look.c
4: :Contents. Anzeigen eines Textfiles oder nur ASCII Z
5: :eichen
6: :Author. Matthias Kuehn [mat]
7: :Copyright. AmigaDOS
8: :Language. ANSI - C
9: :Translator. Lattice C 5.04
10: :Imports. lcr.lib
11: :History. Intuition Graphics
12: :V1.0 [mat] 3.Maer.1990
```

Listing: Look.c

```
12:
13: *****
14:
15: #include <stdio.h>
16: #include <string.h>
17: #include <stdlib.h>
18: #include <ctype.h>
19: #include <exec/types.h>
20: #include <exec/memory.h>
21: #include <intuition/intuition.h>
22: #include <intuition/intuitionbase.h>
23: #include <intuition/screens.h>
24: #include <proto/exec.h>
25: #include <proto/dos.h>
26: #include <proto/intuition.h>
```

Listing: Look.c




```

27: #include <proto/graphics.h>
28:
29: #include "Look.h"
30:
31: /*          D E F I N E S          */
32:
33: #define REV          NULL
34: #define TOP          10
35: #define DOWN         4
36: #define LEFT         3
37: #define RIGHT        18
38: #define HEIGHT       (win->Height - TOP - DOWN)
39: #define WIDTH        (win->Width - LEFT - RIGHT)
40:
41: #define SIZE         512
42:
43: #define QLSHIFT      0x8001
44: #define QRSHIFT      0x8002
45: #define QCTRL        0x8008
46: #define QALT          0x8010
47: #define QALTB        0x8020
48: #define QKEY         0x10
49: #define IKEY         0x17
50: #define SKEY         0x21
51: #define HKEY         0x25
52: #define NKEY         0x36
53: #define SPACE        0x40
54: #define BACKSPACE    0x41
55: #define RETURN       0x44
56: #define HOCH         0x4c
57: #define RUNTER       0x4d
58: #define HELP         0x5f
59:
60: #define HXPOS        110
61: #define HYPOS        20
62: #define hPrint(t)    Move(hRp,HXPOS,hy);Text(hRp,t,
sizeof(t)-1);hy+=9
63:
64: /*          P R O T O S          */
65:
66: void domessage(struct IntuiMessage *msg);
67: void domenu(int menu, int item, int subitem);
68: void parseCLI(int argc, char **argv);
69: void openall(void);
70: void closeall(int rclim);
71: BOOL everyFilter(char);
72: BOOL asciiFilter(char);
73: BOOL alphaFilter(char);
74: BOOL digitFilter(char);
75: BOOL alnumFilter(char);
76: void aufab(void);
77: void fillDown(void);
78: void fillUp(void);
79: void fillPage(void);
80: void fillNew(void);
81: void key(USHORT qu, USHORT rk);
82: void hoch(void);
83: void runter(void);
84: void help(void);
85: struct Window* openwindow(SHORT x, SHORT y, SHORT w, S
HORT h,
86: char* name, ULONG flags, ULONG IDCMP_flags, struct Gad
get* gadget);
87: void search(void);
88: void next(void);
89: void onlyWrite(void);
90:
91: /*          G L O B A L S          */
92:
93: BOOL (*filter)(char); /* Filterfunktion */
94: BOOL ascii, alpha, digit, full, setWt,
95: up, nreturn, sign, sinc, sens;
96: BPTR fin, fout; /* Source, Destination */
97: LONG fsize, fpos, /* File */
98: first[64], last[64], /* Filepos. je Zeile */
99: bNr, /* momentane Block Nummer
100: */
101: bNrMax, /* maximale Block Nummer
102: */
103: sPos, sNr; /* Suchposition */
104: BYTE inBuffer[SIZE], /* Buffer */
105: outBuffer[180];
106: SHORT inBlast, /* " - Fuellung */
107: inBpos, outBpos, /* " - Position */
108: winZ, winS, /* Zeile (Spalte) */
109: countZ, /* Zeilenanzahl zeigen */
110: countF, appear, /* von count */
111: add; /* Erkennt RETURN beim
Richtungswechsel */
112: char space = ' ', count = '0',
113: title[100],
114: Help[] =
115: "Usage : Look Source [TO Destination] [-#sacdn
116: ]\n"
117: "      Source : File to read from\n"
118: "      Destination : File to write to\n"
119: "      # : Digit. Look if min. # Matches\n"
120: "      s : Look only ascii\n"
121: "      a : Look only alpha\n"
122: "      d : Look only digit\n"
123: "      c : Charakter to sign Space\n"
124: "      n : Do no Carriage Return\n"
125: "      h1[] = 'Shift - UP : Look at first line.'\n"
126: "      h2[] = 'Shift - DOWN : Look at last line.'\n"
127: "      h3[] = 'Ctrl - UP : One page up.'\n"
128: "      h4[] = 'Ctrl - DOWN : One page down.'\n"
129: "      h5[] = 'Alt - UP : Half page up.'\n"
130: "      h6[] = 'Alt - DOWN : Half page down.'\n"
131: "      h7[] = 'UP : One line up.'\n"
132: "      h8[] = 'DOWN : One line down.'\n"
133: "      h9[] = 's : Search for string.'\n"
134: "      h10[] = 'i : Search for string cas
e insensitive.'

```



Listing: Look.c

```

132: h11[] = "      n : Search for next strin
g.",
133: h12[] = "      h : ",
134: h13[] = "      HELP : Show this page.",
135: h14[] = "      q : Quit.",
136: h15[] = "      *** Now some 'more' commands: ***",
137: h16[] = "      SPACE : One page down.",
138: h17[] = "      BS : One page up.",
139: h18[] = "      RETURN : One line down.",
140: h19[] = "      h20[] = "      h21[] = "      Press RETURN to go on. ***";
141:
142:
143:
144: struct IntuitionBase *IntuitionBase;
145: struct GfxBase *GfxBase;
146: struct Window *win;
147: struct Screen *screen;
148: struct RastPort *rp;
149:
150: /*          M A I N          P R O G R A M          */
151:
152: void main(int argc, char **argv)
153: {
154: struct IntuiMessage *msg;
155: SHORT knob;
156:
157: parseCLI(argc, argv);
158: appear = countF = (SHORT)(count - '0');
159:
160: /***** Filter schalten *****/
161: knob = ascii*100 + alpha*10 + digit*1;
162: switch(knob)
163: {
164: case 0: filter = everyFilter; break;
165: case 1: filter = digitFilter; break;
166: case 10: filter = alphaFilter; break;
167: case 11: filter = alnumFilter; break;
168: default: filter = asciiFilter; break;
169: } /* switch knob */
170:
171: if(fout) onlyWrite();
172: openall();
173:
174: NewModifyProp(&aufab_gad, win, 0, AUTOKNOB|FREEVERT|K
NOBHIT, 0, 0, 0,
175: (long)MAXBODY/(bNrMax+1), 1);
176: sprintf(title, "Look(ing at %s) by Matthias Kuehn for
AmigaDOS (c)1990" argv[1]);
177: SetWindowTitles(win, title, (char*)-1);
178:
179: /***** erste Seite zeigen *****/
180: fillPage();
181:
182: /***** MAIN LOOP *****/
183: while(TRUE)
184: {
185: if(countZ) /* noch Zeilen anzuzeigen */
186: {
187: if(up) fillUp();
188: else fillDown();
189: countZ--;
190: while (msg = (struct IntuiMessage*) GetMsg(win->Use
rPort))
191: domessage(msg);
192: } /* if countZ */
193: else
194: {
195: NewModifyProp(&aufab_gad, win, 0, AUTOKNOB|FREEVERT
|KNOBHIT, 0,
196: (long)bNr*MAXPOT/(bNrMax+1), 0L, (long)MAXBODY/(bN
rMax+1), 1);
197: WaitPort(win->UserPort);
198: while (msg = (struct IntuiMessage*) GetMsg(win->Use
rPort))
199: domessage(msg);
200: } /* else full */
201:
202: } /* while */
203:
204: } /* main() */
205:
206:
207: /*          D O M E S S A G E          */
208:
209: void domessage(struct IntuiMessage *msg)
210: {
211: ULONG class;
212: USHORT code, qualifier;
213: struct Gadget *gadget;
214:
215: class = msg->Class;
216: code = msg->Code;
217: qualifier = msg->Qualifier;
218: gadget = (struct Gadget*)msg->IAddress;
219: ReplyMsg((struct Message *)msg);
220:
221: switch(class)
222: {
223: case GADGETUP : aufab();
break;
224: case RAWKEY : key(code, qualifier);
break;
225: case NEWSIZE : fillNew();
break;
226: case CLOSEWINDOW : closeall(0);
break;
227: } /* switch */
228:
229: } /* domessage() */
230:
231:
232:
233: /*          O P E N A L L          */
234:
235: void openall(void)
236: {

```

Listing: Look.c

LISTING

```

239: if(!IntuitionBase = (struct IntuitionBase *)OpenLibr
ary("intuition.library",REV))
240: {
241:     puts("No intuition.library !!!");
242:     closeall(20);
243: }
244:
245: if(!GfxBase = (struct GfxBase *)OpenLibrary("graphic
s.library", REV))
246: {
247:     puts("No graphics.library !!!");
248:     closeall(20);
249: }
250:
251: /***** OPEN WINDOW *****/
252: win = openwindow(0, 0,
253:     IntuitionBase->ActiveScreen->Width,
254:     IntuitionBase->ActiveScreen->Height,
255:     NULL,
256:     | WINDOWSIZE | WINDOWDEPTH | NOCAREREFRESH,
257:     NEWSIZE | RAWKEY | GADGETUP | CLOSEW
INDOW,
258:     &aufab_gad);
259: if(win == NULL)
260: {
261:     puts("No Window !!!");
262:     closeall(10);
263: }
264: rp = win->RPort;
265:
266: /* openall() */
267:
268: * C L O S E A L L *
269:
270: void closeall(int rclim)
271: {
272:
273:     if(fIn) Close(fIn);
274:     if(fOut) Close(fOut);
275:     if(win) CloseWindow(win);
276:     if(GfxBase) CloseLibrary((struct Library *)GfxBase);
277:     if(IntuitionBase) CloseLibrary((struct Library *)Int
uitionBase);
278:     exit(rclim);
279: } /* closeall() */
280:
281:
282: /* P A R S E C L I *
283: * wertet die Commandline aus */
284:
285: void parseCLI(int argc, char **argv)
286: {
287:     SHORT i, j;
288:     BPTR lock;
289:     static struct FileInfoBlock *info;
290:
291:     /***** Hilfe *****/
292:     if((argc==1) || (argv[1][0]!='?'))
293:     {
294:         puts(Help);
295:         closeall(0);
296:     } /* argc == 1 */
297:
298:     /***** SOURCE *****/
299:     if(info = (struct FileInfoBlock*)AllocMem(sizeof(struct
FileInfoBlock), MEMF_PUBLIC))
300:     {
301:         if(lock = Lock(argv[1], SHARED_LOCK))
302:         {
303:             Examine(lock, info);
304:             fSize = info->fSize;
305:             bNrMax = fSize/SIZE;
306:             UnLock(lock);
307:             fIn = Open(argv[1], MODE_OLDFILE);
308:             /* if lock */
309:             else
310:                 printf("Can't open File : %s\n", argv[1]);
311:             FreeMem((void*)info, sizeof(struct FileInfoBlock));
312:             /* if info */
313:             else
314:                 puts("Out of Memory !!!");
315:
316:             if(!lock || !info) closeall(10);
317:
318:             /***** ein Argument nur *****/
319:             if(argc == 2) return;
320:
321:             /***** Destination *****/
322:             if(!(!strcmp(argv[2], "TO"))) && (argc>3))
323:                 if(!fOut = Open(argv[3], MODE_NEWFILE))
324:                 {
325:                     printf("Can't open File : %s\n", argv[3]);
326:                     closeall(10);
327:                     /* if !fOut */
328:
329:                     /***** Options *****/
330:                     for(i=2; i<=argc; i++)
331:                     {
332:                         if(argv[i][0] != '-') continue;
333:
334:                         for(j=1; j<strlen(argv[i]); j++)
335:                         {
336:                             if(isdigit(argv[i][j]))
337:                             {
338:                                 count = argv[i][j];
339:                                 continue;
340:                             } /* if digit */
341:
342:                             switch(argv[i][j])
343:                             {
344:                                 case 'a' : alpha = TRUE; break;
345:                                 case 's' : ascii = TRUE; break;
346:                                 case 'd' : digit = TRUE; break;

```

Listing: Look.c

```

347:     case 'n' : nreturn = TRUE; break;
348:     default : space = argv[i][j];
349:     } /* switch */
350:     } /* for j */
351:     } /* for i */
352:
353: } /* parseCLI() */
354:
355: /* F I L T E R *
356: * Funkt. auf die Filter zeigt um von inBuffer->outBuff
er */
357:
358: BOOL everyFilter(char c){ return((BOOL)((31<c&c<128) |
((160<c&c<255)&&c!=215&&c!=247))); } /* everyFilter */
359:
360: BOOL asciiFilter(char c){ return((BOOL)((31<c&c<128) |
(96<c&c<123) | (c==32))); } /* asciiFilter */
361:
362: BOOL alphaFilter(char c){ return((BOOL)((64<c&c<91) |
(96<c&c<123) | (c==32))); } /* alphaFilter */
363:
364: BOOL digitFilter(char c){ return((BOOL)((47<c&c<58) |
(96<c&c<123) | (c==32))); } /* digitFilter */
365:
366: /* A U F A B *
367: * verschiebt den Ausschnitt der im Fenster gezeigt wir
d */
368:
369: void aufab(void)
370: {
371:     bNr = (((struct PropInfo*)aufab_gad.SpecialInfo)->Ve
rtPot*(long)(bNrMax-1)/MAXPOT);
372:     sinc = TRUE;
373:     inBpos = 0;
374:     fillPage();
375:     /* aufab() */
376:
377:     /***** F I L L D O W N *
378:     * bringt die naechste Zeile Zeichen ins Fenster */
379:
380:     void fillDown(void)
381:     {
382:         LONG fPos, lPos;
383:         SHORT zeile, i, spalte;
384:         char c;
385:
386:         /***** Position fuer den Anfang der Zeile errechnen *****/
387:         fPos = bNr*SIZE + inBpos;
388:
389:         /***** outBuffer loeschen *****/
390:         memset((void*)outBuffer, 0x20202020, sizeof(outBuffer
));
391:
392:         /***** Spalte berechnen *****/
393:         spalte = WIDTH / 8;
394:
395:         /***** outBuffer fuellen *****/
396:         for(outBpos=0; outBpos<spalte; inBpos++)
397:         {
398:             /***** inBuffer ganz ausgewertet ? *****/
399:             if(inBpos == inBlast)
400:             {
401:                 if(bNr == bNrMax) /* Wir waren schon im letzten */
402:                     if(outBpos) return;
403:                     else break;
404:                 /***** inBuffer einlesen *****/
405:                 inBlast = Read(fIn, inBuffer, SIZE);
406:                 inBpos = 0;
407:                 bNr++; /* OK ?, dann BNr erhoehen */
408:                 if(inBlast == -1)
409:                 {
410:                     printf("Error : %d\n", IoErr());
411:                     closeall(10);
412:                     /* if -1 */
413:                     /* if == */
414:
415:                     /***** Filtern *****/
416:                     c = inBuffer[inBpos];
417:                     if(!nreturn)
418:                     {
419:                         if(c == '\n')
420:                         {
421:                             inBpos++;
422:                             break;
423:                         } /* if \n */
424:                         if(c == '\t')
425:                         {
426:                             outBpos += 4;
427:                             continue;
428:                         } /* if \t */
429:                         /* if !nreturn */
430:
431:                         if(filter(c))
432:                         {
433:                             appear--;
434:                             if(!appear)
435:                                 inBpos -= countF; /* Sind digit Zeichen da */
436:                                 /* zurueck auf das erste. */
437:                             /* if appear */
438:                             else
439:                             {
440:                                 outBuffer[outBpos++] = c;
441:                                 sign = TRUE;
442:                             } /* else appear */
443:                             /* if filter */
444:                             else
445:                             {
446:                                 if(sign) /* wurde bereits ein space gesetzt? */
447:                                     sign = FALSE;
448:                                 outBuffer[outBpos++] = space;
449:                                 /* if !sign */

```



Listing: Look.c

LISTING

```

450:     appear = countF;
451: }/* else filter */
452: }/* for outBpos */
453:
454: /* Fileposition fuer das Ende der Zeile errechnen */
455: lPos = bNr*SIZE + inBpos - 1;
456:
457: /***** Zeile errechnen *****/
458: zeile = HEIGHT/8;
459:
460: /***** Scrollen *****/
461: for(i=0; i<zeile-1; i++)
462: {
463:     first[i] = first[i+1];
464:     last[i] = last[i+1];
465: }/* for i */
466: ScrollRaster(rp, 0, 8, LEFT, TOP+2, win->Width-RIGHT,
467: TOP+zeile*8+1);
468: Move(rp, LEFT, TOP+zeile*8);
469: Text(rp, outBuffer, spalte);
470:
471: /***** Filepositionen eintragen *****/
472: for(i<64; i++)
473: {
474:     first[i] = fPos;
475:     last[i] = lPos;
476: }/* fillDown() */
477:
478: /* F I L L U P
479: * bringt die vorhergehende Zeile Zeichen ins Fenster*/
480:
481: void fillUp(void)
482: {
483:     LONG          fPos, lPos;
484:     SHORT         zeile, i, spalte;
485:     char          c;
486:
487:     /** Position fuer das Ende der Zeile errechnen **/
488:     lPos = bNr*SIZE + inBpos + add;
489:     add = 0;
490:
491:     /***** outBuffer loeschen *****/
492:     memset((void*)outBuffer, 0x20202020, sizeof(outBuffer));
493:
494:     /***** Spalte berechnen *****/
495:     spalte = WIDTH / 8;
496:
497:     /***** outBuffer fuellen *****/
498:     for(outBpos=spalte-1; outBpos>=0; inBpos--)
499:     {
500:         /***** inBuffer ganz ausgewertet ? *****/
501:         if(inBpos == -1)
502:         {
503:             if(!bNr) /* Sind wir schon am ANFANG */
504:             if(outBpos == spalte) return;
505:             else break;
506:             /** inBuffer einlesen **/
507:             Seek(fin, (--bNr)*SIZE, OFFSET BEGINNING);
508:             inBlast = Read(fin, inBuffer, SIZE);
509:             inBpos = inBlast - 1;
510:             if(inBlast == -1)
511:             {
512:                 printf("Error : %d\n", IoErr());
513:                 closeall(10);
514:             }/* if -1 */
515:             }/* if == */
516:
517:         /***** Filtern *****/
518:         c = inBuffer[inBpos];
519:         if(!nreturn)
520:         {
521:             if(c == '\n')
522:             {
523:                 inBpos--;
524:                 break;
525:             }/* if \n */
526:             if(c == '\t')
527:             {
528:                 outBpos -= 4;
529:                 continue;
530:             }/* if \t */
531:             }/* if !nreturn */
532:
533:         if(filter(c))
534:         {
535:             if(appear)
536:             {
537:                 appear--;
538:                 if(!appear)
539:                 inBpos += countF; /* Sind digit Zeichen da */
540:                 }/* if appear */
541:                 else
542:                 {
543:                     outBuffer[outBpos--] = c;
544:                     sign = TRUE;
545:                 }/* else appear */
546:                 }/* if filter */
547:                 else
548:                 {
549:                     if(sign) /* wurde bereits ein space gesetzt? */
550:                     {
551:                         sign = FALSE;
552:                         outBuffer[outBpos--] = space;
553:                     }/* if !sign */
554:                     appear = countF;
555:                 }/* else filter */
556:                 }/* for outBpos */
557:
558:     /** Fileposition fuer den Anfang der Zeile errechnen */
559:     fPos = bNr*SIZE + inBpos + 1;
560:

```

Listing: Look.c



OASE
Die deutsche Softwarequelle

**AMIGA
Köln 90**
Wir stellen aus
09.-11. November 1990
Halle 12, Stand 216

OASE # 101 FIBU deluxe + (benötigt 1 MB !)

Suchen Sie schon lange eine leistungsstarke Finanzbuchhaltung? Jetzt gibt es die neueste Version der bereits über tausendfach bewährten "FIBU deluxe". Nicht nur der sagenhafte Preis von nur DM 49,- für das Komplettpaket (incl. deutschem Handbuch) setzt neue Maßstäbe, auch die Leistungsstärke:

- FIBU deluxe + verwaltet bis zu 2000 Konten aller Art
- FIBU deluxe + ist mandantenfähig (frei wählbare Währung)
- FIBU deluxe + gibt Ihnen AfA-Vorschläge für jedes Ihrer Anlagekonten
- FIBU deluxe + erstellt und druckt umfangreiche Auswertungslisten
- FIBU deluxe + bietet Ihnen einfache Maus- oder Tastaturbedienung
- FIBU deluxe + erstellt Bilanzen, Journale, Kassenberichte oder GuV
- FIBU deluxe + beinhaltet eine FAKTURA mit direkter Verbuchung

nur sagenhafte **DM 49,-**

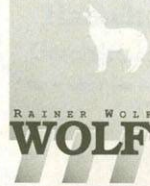
OASE # 109 STEUER 1990

Endlich gibt es das neue Steuerprogramm 1990 mit sämtlichen neuen Steuerrichtlinien der Steuerreform. Dieses Programm erstellt Ihre komplette Lohn- und Einkommensteuererklärung 1990. Natürlich auch mit der aktuellen Lohnsteuertabelle für 1991. Es werden 99% aller Normal- und Sonderfälle abgedeckt. Mit jährlichem Updateservice!

nur vorbildliche **DM 59,-**

Versandkosten: Vorkasse DM 3,- (Ausl. DM 6,-) / Nachnahme: DM 7,- (Ausl. 15,-)

SOFTWARE - WERBUNG - DESIGN



WOLF Software & Design

Deipe Stegge 187
4420 Coesfeld

Telefon: 02541/2874
Telefax: 02541/71172

Software für Amiga Computer made in Germany

Golden Tool Serie

Das unentbehrliche Sammelwerk für alle Amigabesitzer. Im stabilen DIN A5 Ordner finden nützliche Utilities, neben der ausführlichen Dokumentation in deutscher Sprache, Platz. Abgeschlossene Serien werden durch sofort anwendbare Software ergänzt, z.B. IFF-Serie mit Show Programm sowie IFF Bilderdiskette. Erstmals mit spielend einfacher Installationsroutine, die alle Geräte, z.B. Harddisk, unterstützt.

Ausgabe 1 (2 Disketten) im A5 Ordner erhältlich.

Virusdetektor Professional

Neben den altbewährten Funktionen haben wir die neuesten Virenarten erkennbar und eliminierbar gemacht. Nun werden Copy-, Link-, Bootblock- und Fileviren erkannt und beseitigt. Selbst die "trojanischen Pferde" sind nicht unsterblich. Ob Sie Ihre Harddisk oder Ihre Diskette von Viren reinigen wollen, der Virusdetektor steht Ihnen von Anfang an zur Seite. Das ausführliche Handbuch führt Sie in die Virenproblematik ein. Damit Sie wieder ruhig schlafen können.

Update ebenfalls lieferbar.

Alle unsere Produkte erhalten Sie im Fachhandel, z.B. bei Conrad Elektronik.

Händlernachweis in der Schweiz bei:
Swisoft AG ; Postfach 3109; CH-2501 Biel
Telefon : 32231833

Für weitere Informationen sind wir für Sie da:

DIT

Dienstleistungen und Informationstechnologie
Musfeldstraße 1 - 4200 Oberhausen 12
Telefon : 0208 605645

LISTING



```

561:  /***** Zeile errechnen *****/
562:  zeile = HEIGHT/8;
563:
564:  /***** Filepositionen eintragen *****/
565:  for(i=63; i>zeile-1; i--)
566:  {
567:    first[i] = first[zeile-1];
568:    last[i] = last[zeile-1];
569:  } /* for i */
570:
571:  /***** Scrollen *****/
572:  for(; i>0; i--)
573:  {
574:    first[i] = first[i-1];
575:    last[i] = last[i-1];
576:  } /* for i */
577:  first[0] = fPos;
578:  last[0] = lPos;
579:  ScrollRaster(rp, 0, -8, LEFT, TOP+2, win->Width-RIGHT
, TOP+zeile*8+1);
580:  Move(rp, LEFT, TOP+8);
581:  Text(rp, &outBuffer[outBpos+1], spalte);
582:
583:  } /* fillUp() */
584:
585:  /*          K E Y          */
586:  /*          Tastatur auswerten          */
587:
588:  void key(USHORT rk, USHORT qu)
589:  {
590:    if(!(rk & 0x80))          /* nur Key press */
591:    {
592:      if(setWt)
593:      {
594:        setWt = FALSE;
595:        SetWindowTitles(win, title, (char*)-1);
596:      } /* if setWt */
597:      switch(qu)
598:      {
599:        case QRSHIFT :
600:        case QLSHIFT :
601:          switch(rk)
602:          {
603:            case HOCH :          /***** ANFANG *****/
604:              inBpos = 0;
605:              bNr = 0;
606:              fillPage();
607:              break;
608:            case RUNTER :        /***** ENDE *****/
609:              bNr = bNrMax;
610:              inBpos = 0;
611:              sinc = TRUE;
612:              fillPage();
613:              break;
614:            } /* switch rk */
615:          break;
616:        case QCTRL :
617:          switch(rk)
618:          {
619:            case HOCH :          /***** SEITE HOCH *****/
620:              countZ = HEIGHT/8 - 1;
621:              hoch();
622:              break;
623:            case RUNTER :        /***** SEITE RUNTER *****/
624:              countZ = HEIGHT/8 - 1;
625:              runter();
626:              break;
627:          } /* switch rk */
628:          break;
629:        case QRLT :
630:        case QLALT :
631:          switch(rk)
632:          {
633:            case HOCH :          /***** HALBE SEITE HOCH *****/
634:              countZ = HEIGHT/16;
635:              hoch();
636:              break;
637:            case RUNTER :        /***** HALBE SEITE RUNTER *****/
638:              countZ = HEIGHT/16;
639:              runter();
640:              break;
641:          } /* switch rk */
642:          break;
643:        default :
644:          switch(rk)
645:          {
646:            case HOCH :          /***** EINS HOCH *****/
647:              hoch();
648:              countZ = 1;
649:              break;
650:            case RUNTER :        /***** EINS RUNTER *****/
651:              runter();
652:              countZ = 1;
653:              break;
654:            case BACKSPACE :      /***** SEITE HOCH *****/
655:              hoch();
656:              countZ = HEIGHT/8 - 1;
657:              break;
658:            case SPACE :         /***** SEITE RUNTER *****/
659:              runter();
660:              countZ = HEIGHT/8 - 1;
661:              break;
662:            case HELP :          /***** HILFE *****/
663:            case HKEY :
664:              help();
665:              break;
666:            case QKEY :          /***** BYE - BYE *****/
667:              closeall(0);
668:              break;
669:            case SKEY :          /***** SUCHEN *****/
670:              sens = TRUE;
671:              search();
672:              break;
673:            case IKEY :          /***** SUCHEN INSENSITIVE *****/

```

Listing: Look.c

```

675:    sens = FALSE;
676:    search();
677:    break;
678:    case NKEY :          /***** SUCHEN NAECHSTES *****/
679:      snr = first[1]/SIZE;
680:      sPos = first[1]-SIZE*snr;
681:      Seek(fin, snr*SIZE, OFFSET BEGINNING);
682:      inBlast = Read(fin, inBuffer, SIZE);
683:      next();
684:      break;
685:    } /* switch rk */
686:    } /* switch qu */
687:    } /* if rk */
688:    } /* key() */
689:
690:    /*          H O C H          */
691:    /*          vorbereiten fuer Zeile hoch scrollen          */
692:
693:  void hoch(void)
694:  {
695:    LONG          tNr;
696:
697:    if(!up)
698:    {
699:      up = TRUE;
700:      tNr = first[0]/SIZE;
701:      inBpos = first[0]-SIZE*tNr - 1;
702:      if(tNr != bNr)
703:      {
704:        bNr = tNr;
705:        Seek(fin, bNr*SIZE, OFFSET BEGINNING);
706:        inBlast = Read(fin, inBuffer, SIZE);
707:      } /* if != */
708:      if(inBuffer[inBpos] == '\n')          /* damit das cr am
Ende der ersten Zeile nicht beachtet wird */
709:      {
710:        inBpos--;
711:        add = 1;
712:      } /* if \n */
713:    } /* if !up */
714:    } /* hoch() */
715:
716:    /*          R U N T E R          */
717:    /*          vorbereiten fuer Zeile runter scrollen          */
718:
719:  void runter(void)
720:  {
721:    LONG          tNr;
722:
723:    if(up)
724:    {
725:      up = FALSE;
726:      tNr = last[(HEIGHT/8)-1]/SIZE;
727:      inBpos = last[(HEIGHT/8)-1]-SIZE*tNr + 1;
728:      if(tNr != bNr)
729:      {
730:        bNr = tNr;
731:        Seek(fin, bNr*SIZE, OFFSET BEGINNING);
732:        inBlast = Read(fin, inBuffer, SIZE);
733:      } /* if != */
734:    } /* if up */
735:    } /* runter() */
736:
737:    /*          H E L P          */
738:    /*          Hilfstext ausgeben          */
739:
740:  void help(void)
741:  {
742:    struct Window          *hWin;
743:    struct RastPort          *hRp;
744:    struct IntuiMessage          *msg;
745:    LONG          hy = HYPOS;
746:    ULONG          class;
747:    USHORT          code;
748:
749:    /***** Oeffnen *****/
750:    hWin = openwindow(0, 0,
IntuitionBase->ActiveScreen->Width,
IntuitionBase->ActiveScreen->Height
751:    ,
752:    "HELP for Look",
753:    ACTIVATE | WINDOWCLOSE | NOCAREREFR
754:    ESH,
755:    RAWKEY | CLOSEWINDOW,
756:    NULL);
757:
758:    if(!hWin)
759:    {
760:      SetWindowTitles(win, "Can't open Window to help !!!"
, (char*)-1);
761:      setWt = TRUE;
762:      return;
763:    } /* if ! hWin */
764:    hRp = hWin->RPort;
765:    SetAPen(hRp, 1);
766:
767:    /***** Rumkritzeln *****/
768:    hPrint(h1);
769:    hPrint(h2);
770:    hPrint(h3);
771:    hPrint(h4);
772:    hPrint(h5);
773:    hPrint(h6);
774:    hPrint(h7);
775:    hPrint(h8);
776:    hPrint(h9);
777:    hPrint(h10);
778:    hPrint(h11);
779:    hPrint(h12);
780:    hPrint(h13);
781:    hPrint(h14);
782:    hPrint(h15);
783:    hPrint(h16);
784:    hPrint(h17);

```

Listing: Look.c



```

786: hPrint(h18);
787: hPrint(h19);
788: hPrint(h20);
789: hPrint(h21);
790:
791: /***** Warten *****/
792: do
793: {
794:   WaitPort(hWin->UserPort);
795:   msg = (struct IntuiMessage*)GetMsg(hWin->UserPort);
796:   class = msg->Class;
797:   code = msg->Code;
798:   ReplyMsg((struct Message*)msg);
799:   while((class!=CLOSEWINDOW) && ((class!=RAWKEY)|| (code
    !=RETURN)));
800:
801: /***** Schließen *****/
802: CloseWindow(hWin);
803:
804: }/* help() */
805:
806: /*          F I L L P A G E          */
807: /*          fuellt das Fenster mit Zeichen          */
808:
809: void fillPage(void)
810: {
811:   LONG          fPos, lPos;
812:   SHORT         zeile, i, spalte;
813:   char          c, *ptr;
814:
815: /***** Vorbereitung treffen *****/
816: Seek(fIn, bNr*SIZE, OFFSET BEGINNING);
817: inBlast = Read(fIn, inBuffer, SIZE);
818: up = FALSE;
819:
820: if(!return && sinc)
821: {
822:   sinc = FALSE;
823:   ptr = memchr(inBuffer, '\n', SIZE);
824:   if(ptr)
825:     inBpos = ptr - inBuffer + 1;
826:   else
827:     inBpos = 0;
828: }/* if !return */
829:
830: /***** Fenster saeuern *****/
831: SetAPen(rp, 0);
832: RectFill(rp, LEFT, TOP+1, win->Width-RIGHT, TOP+(HEIGHT/8
    )+8+2);
833: SetAPen(rp, 1);
834:
835: /***** Spalte berechnen *****/
836: spalte = WIDTH / 8;
837:
838: /***** Zeile errechnen *****/
839: zeile = HEIGHT/8;
840:
841: /***** Das ganze Fenster fuellen *****/
842: for(i=0; i<zeile; i++)
843: {
844:   /* Position fuer den Anfang der Zeile errechnen */
845:   fPos = bNr*SIZE + inBpos;
846:
847:   /***** outBuffer loeschen *****/
848:   memset((void*)outBuffer, 0x20202020, sizeof(outBuffer));
849:
850:   /***** outBuffer fuellen *****/
851:   for(outBpos=0; outBpos<spalte; inBpos++)
852:   {
853:     /***** inBuffer ganz ausgewertet ? *****/
854:     if(inBpos == inBlast)
855:     {
856:       if(bNr == bNrMax) /* Wir waren schon im letzten */
857:         break;
858:       /* inBuffer einlesen */
859:       inBlast = Read(fIn, inBuffer, SIZE);
860:       inBpos = 0;
861:       bNr++;
862:       if(inBlast == -1) /* OK ?, dann BNR erhoehen */
863:       {
864:         printf("Error : %d\n", IoErr());
865:         closeall(10);
866:       }/* if -1 */
867:     }/* if == */
868:
869:     /***** Filtern *****/
870:     c = inBuffer[inBpos];
871:     if(!return)
872:     {
873:       if(c == '\n')
874:       {
875:         inBpos++;
876:         break;
877:       }/* if '\n */
878:       if(c == '\t')
879:       {
880:         outBpos += 4;
881:         continue;
882:       }/* if '\t */
883:     }/* if !return */
884:
885:     if(filter(c))
886:     {
887:       if(appear)
888:       {
889:         appear--;
890:         if(!appear)
891:           inBpos -= countF; /* Sind digit Zeichen da */
892:         /* zurueck auf das erste. */
893:       }/* if appear */
894:       else
895:       {
896:         outBuffer[outBpos++] = c;
897:         sign = TRUE;
898:       }/* else appear */

```

Listing: Look.c

Mac Soft Amiga Shop

Hard- & Software Schulungen

Public Domain

über 7000 Disketten ! Ständig aktuell !

Public-Domain-Software incl. Diskette ab 1,50 DM

***** 24 Stunden Bestell- und Versand - Service *****

Ladenlokal mit Fachberatung, Reparatur und Entwicklung
in eigener Werkstatt mit Schnellservice

Hardware-Zusammenstellung nach Ihren Vorstellungen

Kurse für Einsteiger und Insider in unseren Schulungsräumen

Highlights : unsere thematische Super-Serie mit vielen

exklusiven Programmen nur bei uns - 8 DM

z.B.: Anti-Virus-Disk, Übersetzer, Grafik, Powerbench uvm

2 Katalog-Disks (= 5 normale Disks) in **DEUTSCH** anfordern 7,-

Telefon 0231/ 516010

BTX Bestellservice * mac soft amiga #

Hannöversche Str. 82 4600 Dortmund 1

Geschäftszeiten: Mo-Fr 10-13 u 15-19 Uhr Sa 10-15 Uhr

Warum nicht gleich bei uns?

TOP - Aktualität bei der PD!

Alle gängigen Serien und das Neueste von ACS über Fish bis Taifun.
Weit über 9000 Disketten! Top-aktuell z.B.: Berlin PD (-45), Docs by Dr.
Knox (-19). **Neu: PD top-aktuell** in unserer **Mailbox** (030-3213436)!!!
2 deutsche Katalogdisks mit vielen Infos und Tips: **DM 5,-** incl. Versand

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 (außer Fish, Kickstart) **2.80 DM / 3.5" 2DD**
von uns zusammengestellt
zum Abo-Preis von nur **DM 25,-** !! **1.80 DM / 5.25" 2DD**

TOP - Qualität bei der Hardware!

512 KB RAM für A500, Uhr + Abschalter	DM 119,-
3.5" LW extern, Chinon oder NEC, mit Bus	DM 175,-
5.25" LW extern, TEAC, mit Bus, 40/80 Tr.	DM 225,-
A580 plus 2 MB für A500, davon 0.5 für 1MB Chip-Ram	DM 529,-
Megamix 2 MB/8 MB für A2000, voll autokonfigurierend	DM 569,-
Supra 4 MB/8MB für A2000, voll autokonfigurierend	DM 879,-
Supra Filecard 32 MB A2000, SCSI, umfangr. Softw. + Anl.	DM 999,-
Supra Filecard mit Quantum 40 MB für A2000, 19 ms	DM 1169,-
Supra Filecard 60 MB für A2000, SCSI, 24 ms, dt. Anleit.	DM 1279,-
Supra Filecard 84 MB für A2000, SCSI, 24 ms, dt. Anleit.	DM 1439,-
Supra Filecard mit Quantum 105 MB für A2000, 19 ms	DM 1839,-
Supra Drive XP 500 Harddisk, SCSI, bis 8 MB on Board	a. Anfr.

Harddisks für A500 ab DM790,-; individuelle Konfigurationen auch mit
anderen Controllern - z.B. Trumcard- möglich. Eigener Reparaturservice.

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

Gabriele von Thienen

Handeln mit neuen Medien

Postfach 100 648, 1000 Berlin 10

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


```

898:  }/* if filter */
899:  else
900:  {
901:      if(sign) /* wurde bereits ein space gesetzt? */
902:      {
903:          sign = FALSE;
904:          outBuffer[outBpos++] = space;
905:      }/* if !sign */
906:      appear = countF;
907:      }/* else filter */
908:      }/* for outBpos */
909:
910:  /* Fileposition fuer das Ende der Zeile errechnen */
911:  lPos = bNr*SIZE + inBpos - 1;
912:
913:  /***** Scrollen *****/
914:  Move(rp, LEFT, TOP+8+i*8);
915:  Text(rp, outBuffer, spalte);
916:
917:  /***** Filepositionen eintragen *****/
918:  first[i] = fPos;
919:  last[i] = lPos;
920:  }/* for i */
921:  }/* fillPage() */
922:
923:  /* F I L L N E W
924:  *   füllt bei Veraenderungen erneut das Fenster
925:  */
926:  void fillNew(void)
927:  {
928:      SetAPen(rp, 0);
929:      RectFill(rp, 2, 12, win->Width-18, win->Height-2);
930:      SetAPen(rp, 1);
931:      up = FALSE;
932:      bNr = first[0]/SIZE;
933:      inBpos = first[0]-SIZE*bNr;
934:      Seek(fIn, bNr*SIZE, OFFSET BEGINNING);
935:      inBlast = Read(fIn, inBuffer, SIZE);
936:
937:      fillPage();
938:  }/* fillNew() */
939:
940:  /* O P E N W I N D O W
941:  *   erleichtert das Oeffnen eines Fensters
942:  */
943:  struct Window* openwindow(SHORT x, SHORT y, SHORT w, S
HORT h,
944:  char* name, ULONG flags, ULONG IDCMP_flags, struct Gad
get* gadget)
945:  {
946:      struct NewWindow nw;
947:
948:      nw.LeftEdge = x;
949:      nw.TopEdge = y;
950:      nw.Width = w;
951:      nw.Height = h;
952:      nw.DetailPen = 0;
953:      nw.BlockPen = 1;
954:      nw.Title = name;
955:      nw.Flags = flags;
956:      nw.IDCMPFlags = IDCMP_flags;
957:      nw.Screen = NULL;
958:      nw.Type = WBENCHSCREEN;
959:      nw.FirstGadget = gadget;
960:      nw.CheckMark = NULL;
961:      nw.BitMap = 0;
962:      nw.MinWidth = 100;
963:      nw.MinHeight = 30;
964:      nw.MaxWidth = 640;
965:      nw.MaxHeight = 256;
966:
967:      return( OpenWindow(&nw) );
968:  }/* openwindow() */
969:
970:  /* S E A R C H
971:  *   Eing. eines Strings und suchen ab der moment. Positi
on
972:  */
973:  void search(void)
974:  {
975:      struct Window sWin;
976:      struct IntuiMessage *msg;
977:      ULONG class;
978:      USHORT code;
979:
980:      /***** Oeffnen *****/
981:      sWin = openwindow((SHORT)((IntuitionBase->ActiveScre
e
n->Width-250)/2),
982:      (SHORT)((IntuitionBase->ActiveScre
e
n->Height-25)/2),
983:      250, 25,
984:      "Search for what ?",
985:      ACTIVATE | WINDOWCLOSE | NOCAREREF
986:      ESH | WINDOWDRAG, CLOSEWINDOW | GADGETUP,
987:      &string_gad);
988:
989:      if(!sWin)
990:      {
991:          SetWindowTitles(win, "Can't open Window to Search !!
", (char*)-1);
992:          setWt = TRUE;
993:          return;
994:      }/* if ! sWin */
995:      ActivateGadget(&string_gad, sWin, 0L);
996:
997:      /***** Warten *****/
998:      do
999:      {
1000:          WaitPort(sWin->UserPort);
1001:          msg = (struct IntuiMessage*)GetMsg(sWin->UserPort);
1002:          class = msg -> Class;
1003:          code = msg -> Code;
1004:          ReplyMsg((struct Message*)msg);
1005:      }while((class!=CLOSEWINDOW) && (class!=GADGETUP));

```

Listing: Look.c

```

1006:
1007:  /***** Schließen *****/
1008:  CloseWindow(sWin);
1009:  sNr = first[0]/SIZE;
1010:  sPos = first[0]-SIZE*sNr;
1011:  Seek(fIn, sNr*SIZE, OFFSET BEGINNING);
1012:  inBlast = Read(fIn, inBuffer, SIZE);
1013:  next();
1014:  }/* search() */
1015:
1016:  /* N E X T
1017:  *   naechsten String suchen
1018:  */
1019:  void next(void)
1020:  {
1021:      char *ptr, *lPtr, *uPtr, error[100];
1022:
1023:      /***** SUCHEN *****/
1024:      ptr = &inBuffer[sPos] - 1;
1025:      while(TRUE)
1026:      {
1027:          /* erstes Vorkommen suchen egal ob groß oder klein */
1028:          lPtr = memchr(++ptr, (int)tolower(string_sbuf[0]), S
IZE-sPos);
1029:          uPtr = memchr(ptr, (int)toupper(string_sbuf[0]), SIZ
E-sPos);
1030:          ptr = min(lPtr, uPtr);
1031:          if(sens)
1032:          {
1033:              if(!strncmp(ptr, string_sbuf, strlen(string_sbuf)))
1034:                  break;
1035:          }/* if sens */
1036:          else
1037:          {
1038:              if(!strnicmp(ptr, string_sbuf, strlen(string_sbuf))
)
1039:                  break;
1040:          }/* else sens */
1041:          /**** naechster Block ****/
1042:          if(!ptr)
1043:          {
1044:              if(sNr == bNrMax)
1045:                  break;
1046:              inBlast = Read(fIn, inBuffer, SIZE);
1047:              sPos = 0;
1048:              ptr = inBuffer;
1049:              sNr++;
1050:              if(inBlast == -1)
1051:              {
1052:                  printf("Error : %d\n", IoErr());
1053:                  closeall(10);
1054:              }/* if -1 */
1055:          }/* if !ptr */
1056:          }/* while TRUE */
1057:
1058:          if(ptr)
1059:          {
1060:              /***** ANZEIGEN *****/
1061:              bNr = sNr;
1062:              if(!return)
1063:              {
1064:                  for(; ptr>inBuffer; ptr--)
1065:                      if(*ptr == '\n') break;
1066:                  inBpos = ptr - inBuffer + 1;
1067:                  }/* if !return */
1068:                  fillPage();
1069:              }/* if ptr */
1070:              else
1071:              {
1072:                  /***** RUECKSTELLEN *****/
1073:                  sprintf(error, "String \"%s\" not found", string_sbu
f);
1074:                  SetWindowTitles(win, error, (char*)-1);
1075:                  setWt = TRUE;
1076:                  Seek(fIn, bNr*SIZE, OFFSET BEGINNING);
1077:                  inBlast = Read(fIn, inBuffer, SIZE);
1078:                  }/* else ptr */
1079:                  }/* next() */
1080:
1081:  /* O N L Y W R I T E
1082:  *   filtern von Source nach Destination
1083:  */
1084:  void onlyWrite(void)
1085:  {
1086:      char c;
1087:
1088:      inBlast = Read(fIn, inBuffer, SIZE);
1089:      while(TRUE)
1090:      {
1091:          /***** outBuffer loeschen *****/
1092:          memset((void*)outBuffer, 0x20202020, sizeof(outBuffe
r));
1093:
1094:          /***** outBuffer fuellen *****/
1095:          for(outBpos=0; outBpos<76; inBpos++)
1096:          {
1097:              /***** inBuffer ganz ausgewertet ? *****/
1098:              if(inBpos == inBlast)
1099:              {
1100:                  if(bNr == bNrMax) /* Wir waren schon im letzten */
1101:                      if(!outBpos) closeall(10); /* ENDE */
1102:                      else break; /* noch ei
ne Zeile schreiben */
1103:                  inBlast = Read(fIn, inBuffer, SIZE);
1104:                  inBpos = 0;
1105:                  bNr++; /* OK ?, dann BNR erhoeihen */
1106:                  if(inBlast == -1)
1107:                  {
1108:                      printf("Error : %d\n", IoErr());
1109:                      closeall(10);
1110:                  }/* if -1 */
1111:                  }/* if == */
1112:                  }/* filtern *****/
1113:
1114:

```

Listing: Look.c

LISTING

```

1115: c = inBuffer[inBpos];
1116: if(!return)
1117: {
1118:   if(c == '\n')
1119:   {
1120:     inBpos++;
1121:     break;
1122:   }
1123:   /* if '\n */
1124:   if(c == '\t')
1125:   {
1126:     outBpos += 4;
1127:     continue;
1128:   }
1129:   /* if '\t */
1130:   if(filter(c))
1131:   {
1132:     if(appear)
1133:     {
1134:       appear--;
1135:       if(!appear)
1136:       {
1137:         inBpos -= countF; /* sind digit Zeichen da */
1138:         /* zurueck auf das erste. */
1139:       }
1140:       else
1141:       {
1142:         outBuffer[outBpos++] = c;
1143:         sign = TRUE;
1144:       }
1145:       /* else appear */
1146:       /* if filter */
1147:     }
1148:     else
1149:     {
1150:       if(sign) /* wurde bereits ein space gesetzt? */
1151:       {
1152:         sign = FALSE;
1153:         outBuffer[outBpos++] = space;
1154:       }
1155:       /* if !sign */
1156:       appear = countF;
1157:       /* else filter */
1158:     }
1159:     /* for outBpos */
1160:     /****** RETURN anhaengen und schreiben *****/
1161:     outBuffer[outBpos++] = '\n';
1162:     if(Write(out, outBuffer, outBpos) == -1)
1163:     {
1164:       printf("Error : %d\n", IoErr());
1165:       closeall(10);
1166:     }
1167:     /* if -1 */
1168:     /* while TRUE */
1169:     /* onlyWrite() */

```



```

1: /*****
2:
3: :Programm. LookGad.h
4: :Contents. Gadgets für Look.c
5: :Author. Matthias Kühn [mat]
6: :Copyright. AmigaDOS
7: :Language. ANSI - C
8: :Translator. Lattice C 5.04
9: :AmigaLibs. Intuition
10:
11: *****/
12:
13: /* STRUCT = */
14:
15: struct Image aufab_knobim;
16: struct PropInfo aufab_prop =
17: {
18:   AUTOKNOB | FREEVERT,
19:   0, 0, 0, 100,
20:   0, 0, 0, 0, 0, 0
21: };
22: struct Gadget aufab_gad =
23: {
24:   NULL,
25:   -15, 10, 16, -19,
26:   GADGIMAGE | GRELRIGHT | GRELHEIGHT,
27:   RELVERIFY | RIGHTBORDER,
28:   PROPGADGET,
29:   (APTR)&aufab_knobim,
30:   NULL, NULL, NULL,
31:   (APTR)&aufab_prop,
32:   NULL, NULL
33: };
34:
35: SHORT string_Pairs[] = { 0, 0, 30*8+1, 0, 30*8+1, 9, 0, 9,
36:   0, 0 };
37: struct Border string_bord = {
38:   -1, -1, 1, 3, JAMI,
39:   5, string_Pairs, NULL
40: };
41: UBYTE string_ubuf[70];
42: UBYTE string_sbuf[70];
43: struct StringInfo string_txstr = {
44:   string_sbuf, string_ubuf,
45:   0, 70, 0,
46:   0, 0, 0, 0, 0, 0, 0, NULL
47: };
48: struct Gadget string_gad = {
49:   NULL,
50:   4, 12, 30*8, 9,
51:   GADGCOMP, RELVERIFY, STRGADGET,
52:   (APTR)&string_bord,
53:   NULL, NULL, NULL,
54:   (APTR)&string_txstr,
55:   NULL, NULL
56: };

```

Listing: Look.h



Funkbilder mit dem PC AMIGA, ATARI, 64/128er

Fernschreiben, Morsen und Fax sowie Radio-Kurzwellen-Nachrichten aller Art. Haben Sie schon einmal das Piepsen von Ihrem Radio auf dem Bildschirm sichtbar gemacht? Hat es Sie schon immer interessiert, wie man Wetterkarten, Meteosat-Bilder, Wetternachrichten, Presseagenturen, -Botenschaftsdienste usw. auf dem Computer sichtbar macht? Ja? -, dann lassen Sie sich ein Info schicken.

Angebote für Empfang und Sendung
248,- DM (64/128er) bis 398,- DM (PC)

Bitte Info Nr. 18 anfordern. Telefon 0 50 52 / 60 52



Fa. Peter Walter, BONITO
Gerichtsweg 3, D-3102 Hermannsburg



SEHR GUT,*
jetzt noch besser!

Digi Tiger II
der ganz besondere
Videodigitizer

Mathildenstr. 12 3000 Hannover 91
05 11 / 62 98 25

KLAUS D. TUTE
Soft-Art- und Hardware

- Jetzt in der Version 2.0
- 2 - 64 Farben-Modus
- optimierter HAM-Modus
- Farbbilder in 25 - 85 s, alle Auflösungen
- SW-Bilder in 1 - 5 s
- integrierter RGB-Splitter
- Hotline und Update-Service
- Infos gratis, Demodiskette für 10 DM Schein

**weiterhin
nur 698,-**

* siehe Kickstart 5/90

DONAU-SOFT

24 h-Schnellversand

Ihr Amiga-PD-Partner

● ab 2,50 DM ●

Alle gängigen Serien sind
lieferbar

Einzeldisk	4,50 DM
ab 10 Disk	4,— DM
ab 50 Disk	3,50 DM
ab 100 Disk	3,30 DM
ab 200 Disk	3,— DM
bei Serienabnahme:	ab 2,50 DM

Preise incl. 3,5" DD-Disks
— Mit Qualitätsgarantie —

Wir kopieren nur mit doppeltem Verify.
Alle Disks sind:
— 100 % Virus- und Error frei
— etikettiert.

Leerdisketten 3,5" 2DD von	
Sentinel	ab 1,15 DM
Sony	ab 1,50 DM

3 ausführliche Katalogdisketten
mit Kurzbeschreibung aller
Programme gegen 10,— DM
(V-Scheck/Briefmarken) anfordern!

gratis zu unseren Katalogen:
Viruskiller, CLJ-Wizard + Turbo Backup

Das große Amiga-PD-Handbuch
Band I-IV + alle 42 Disks
+ 3 Katalogdisketten
(Einzelpreise erfragen) **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.	155,— DM
Floppy 3 1/2" ext.	abschaltbar 189,— DM
Floppy 5 1/4" ext.	mit allen Extras 249,— DM

+ DM 5,— bei Vorkasse, + DM 8,— bei Nachnahme
Ausland: + DM 10,— (nur Vorkasse)

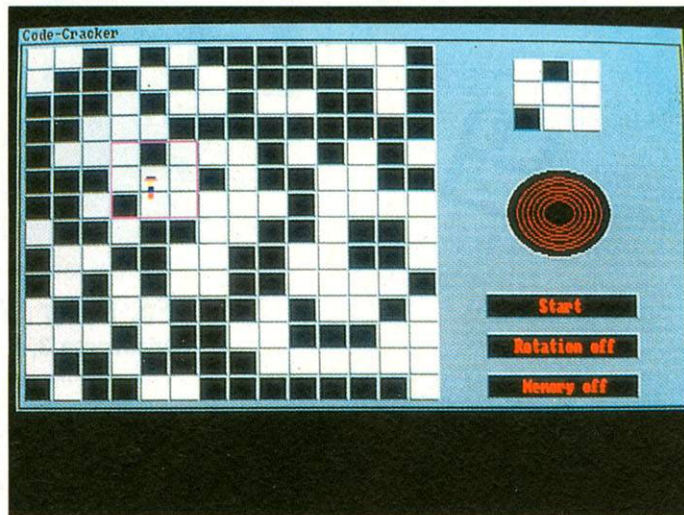
MAIK HAUER

Postfach 1401, 8858 Neuburg Fax: 08431/49800
Tel.: 08431/49798 (bis 22 Uhr) BTX: *Donau-Soft #...

Wer denkt, intelligente und fesselnde Spiele lassen sich nur mit Compiler-Sprachen erstellen, kennt »Code-Cracker« noch nicht. Programmiert in Amiga-Basic, fesselt es den Spieler für lange Zeit auf den Sitz und ist gleichzeitig die richtige Medizin gegen lange, verregnete Herbstabende.

Bei diesem Spiel werden Sie von einem intelligenten, sprechenden Computer herausgefordert. Sie sollen versuchen, in einer vorgegebenen Zeitspanne einen Ausschnitt aus einem Leuchtfeld wiederzufinden. Schaffen Sie dies, so müssen Sie sich erneut, aber in kürzerer Zeit beweisen. Als besondere Vorzüge weist dieses Spiel eine hohe Geschwindigkeit und eine gute Sprachausgabe auf.

Nach dem Start des Programms müssen Sie warten, bis der Computer das Spielfeld fertig generiert hat. Nun wandelt sich die Schrift der Schalter von Rot auf Grün. Der Computer gibt sein »OK«. Dies signalisiert, daß die Schalter nun mit der Maus betätigt werden können. Die beiden unteren Schalter beeinflussen den Schwierigkeitsgrad: »Rotation on« bedeutet, daß der Computer den Ausschnitt eventuell um 90 Grad, 180 Grad oder 270 Grad um sein Mittelfeld dreht. Dies verlangt von Ihnen bei der Suche ein gutes Vorstellungsvermögen. Steht der untere Schalter auf »Memory on«, so wird der Ausschnitt nach dem Start nur kurz für die Dauer von 3 Signaltönen gezeigt. Bei komplizierten Mustern verlangt das ein gutes Gedächtnis. Natürlich können Sie »Rotation« und »Memory« auch kombinieren. Betätigen Sie den Startschalter, so zeigt



Oliver Gumtau

Knacken Sie den Code!

Ein recht kniffliges Denkspiel für AmigaBasic-Fans – Code-Cracker.

der Computer sofort sein Codefeld, den 3x3 Felder großen Ausschnitt rechts oben, und das Spiel beginnt. Die Schrift der Felder ändert sich wieder in Rot, das heißt, der Schwierigkeitsgrad kann im Spiel nicht geändert werden. Mit der Maus steuern Sie eine rotviolette Umrahmung mit dem Format des Ausschnitts, was das Suchen erheblich erleichtert.

Los geht's

Sie müssen den Mauszeiger nur auf das Mittelfeld des ge-

wünschten Ausschnitts richten. Sind Sie der Meinung, daß Sie den gesuchten Ausschnitt gefunden haben, so betätigen Sie den linken Mausknopf und lassen ihn dort bitte wieder los. Liegen Sie falsch, so ertönt ein Piepton, und der Bildschirm blitzt auf. Hatten Sie recht, so bringt der Computer eine Erfolgsmeldung und generiert ein neues Spielfeld.

Danach beginnt sogleich ein neues Spiel, in dem Sie weniger Zeit zur Verfügung haben. Es ist auch erlaubt, daß Sie einen Ausschnitt wählen, den der Computer nicht meinte,

der aber dennoch richtig ist, weil er mehrmals auftaucht.

Die Zeit im Nacken

Der einzige Haken ist natürlich die Zeitbegrenzung. Das etwas eigenwillige Chronometer sehen Sie rechts in der Mitte. Die sechs roten Ringe leuchten der Reihe nach von innen nach außen auf, wobei jedesmal ein »Pling« ertönt. Ist ein imaginärer siebenter Ring erreicht, so wird Ihr gesamtes Spiel beendet. Der Computer zeigt den Ausschnitt im Codefeld mit einer blinkenden Umrahmung an und gibt einen Kommentar zu Ihrer Spielweise ab. Dieser richtet sich nach der erreichten Runde und dem Schwierigkeitsgrad. »Rotation« geht jeweils mit vier, »Memory« mit zwei zusätzlichen Runden in die Wertung ein, sofern Sie wenigstens die erste Runde geschafft haben. Das Results-Fenster müssen Sie durch Anklicken des Close-Gadgets schließen, damit ein neues Spielfeld generiert wird und Sie erneut starten können. Noch ein Tip: Finden Sie das Muster nicht, das die gelben Leuchtfelder des Ausschnitts bilden, so ist dieses sicher nicht abgegrenzt genug. Suchen Sie dann einmal nach dem Muster, das die schwarzen Lücken bilden!

(vb)

AMIGA DOS Info

Hallo Amiga-Basic-Programmierer, falls Sie auch ähnlich gute Ideen zu Spielen haben, lassen Sie sie nicht in Schubladen verkümmern, sondern schicken Sie sie uns zu.

Listings

```
1:  *****
2:  *                CODE - CRACKER                *
3:  *                by Oliver Gumtau                *
4:  *                (c) 1990 DMV Verlag              *
5:  *                Sprache: AMIGA BASIC            *
6:  *                *****
7:
8:  SCREEN 2,640,200,3,2
9:  WINDOW 2,"Code-Cracker",16,2
10: PALETTE 0,0,0:PALETTE 1,.47,.6,.53
11: PALETTE 2,0,0,0:PALETTE 3,.27,.4,.33
12: PALETTE 4,.73,.87,.8:PALETTE 5,.47,0,0
13: PALETTE 6,1,0,0:PALETTE 7,1,0,.8
14: DEFINT a-y
15: RANDOMIZE TIMER
16:
17: DIM Cu(710),Fe(710)
18: CLS:LINE (0,0)-(1,38),7,b:LINE (1,0)-(83,0),7
19: LINE (82,0)-(83,38),7,b:LINE (0,38)-(82,38),7
20: GET (0,0)-(83,38),Cu:CLS
21: DIM Feld(13,13),Code(2,2):Zeitschalt=8:Lev=0
```

Listing: CODE-CRACKER

```
22:
23: PAINT (0,0),1
24: FOR i=0 TO 8:READ Stimme(i):NEXT i
25: DATA 80,0,170,0,21000,64,7,0,0
26: ON ERROR GOTO Fehler
27: SAY " ":WINDOW 2
28: ON ERROR GOTO 0
29: SAY "KOW9D KRAE5KAXR -",Stimme
30: SAY "- AX2KSEH7PT DHAX CHAE4LIHNJ.",Stimme
31:
32: FOR i=0 TO 13:FOR j=0 TO 13
33:   x=4+j*28:y=2+i*13
34:   GOSUB Leuchtfeld:NEXT j,i
35:
36: FOR i=0 TO 2: FOR j=0 TO 2
37:   x=468+j*28:y=10+i*13
38:   GOSUB Leuchtfeld
39: NEXT j,i
40:
41: CIRCLE (510,90),50,4
42: PAINT (510,90),0,4
43:
44: x=440:y=130:GOSUB Schalter
45: POKEW WINDOW(8)+36,490:POKEW WINDOW(8)+38,139
```

Listing: CODE-CRACKER





```

46: COLOR 6:PRINT "Start"
47: y=150:GOSUB Schalter:y=170:GOSUB Schalter
48:
49: Neu:
50: FOR i=40 TO 15 STEP -5
51:   CIRCLE (510,90),i,5:NEXT i
52:   FOR z=1 TO 0 STEP -.007
53:     PALETTE 2,z,z,z*.6
54:   NEXT z
55:
56:   IF Lev>0 THEN Feld
57:   i=INT(RND*3)
58:   IF i=0 THEN SAY "WEY7T AX LIHTUL MOW4MAXNT PLIY2Z.",Stimme
59:   IF i=1 THEN SAY "JAH7ST AX MIH4NIHT PLIY2Z.",Stimme
60:   IF i=2 THEN SAY "WAH7N MOW4MAXNT PLIY2Z.",Stimme
61:
62:   Feld:
63:   FOR i=0 TO 13:FOR j=0 TO 13
64:     Feld(i,j)=INT(RND*2)
65:     x=4+j*28:y=2+i*13:f=Feld(i,j)*2
66:     GOSUB Inneres:NEXT j,i
67:
68:     x1=INT(RND*12):y1=INT(RND*12)
69:     FOR i=0 TO 2:FOR j=0 TO 2
70:       Code(i,j)=Feld(y1+i,x1+j)
71:       x=468+j*28:y=10+i*13:f=Code(i,j)*2
72:       GOSUB Inneres:NEXT j,i
73:       IF Lev>0 THEN Rotation
74:       FOR z=0 TO 1 STEP .007
75:         PALETTE 6,1-z,z,0
76:       NEXT z
77:       SAY "OW7 KEY3.",Stimme
78:
79:   Anzeige:
80:   POKEW WINDOW(8)+36,466:POKEW WINDOW(8)+38,159
81:   IF rflag=0 THEN PRINT "Rotation off"
82:   IF rflag THEN PRINT "Rotation on "
83:   POKEW WINDOW(8)+36,474:POKEW WINDOW(8)+38,179
84:   IF mflag=0 THEN PRINT "Memory off";
85:   IF mflag THEN PRINT "Memory on ";
86:   WHILE MOUSE(0)<>0:WEND
87:
88:   Mausabfrage:
89:   WHILE MOUSE(0)=0:WEND
90:   xm=MOUSE(1):ym=MOUSE(2)
91:   IF xm<444 OR xm>585 THEN Mausabfrage
92:   IF ym<130 AND ym<142 THEN Rotation
93:   IF ym>150 AND ym<162 THEN IF rflag THEN rflag=0:GOTO
94:   Anzeige:ELSE rflag=1:GOTO Anzeige
95:   IF ym>170 AND ym<182 THEN IF mflag THEN mflag=0:GOTO
96:   Anzeige:ELSE mflag=1:GOTO Anzeige
97:   GOTO Mausabfrage
98:
99:   Rotation:
100:  i=INT(RND*5)
101:  IF i=0 THEN SAY "GEH3T REH9DIY.",Stimme
102:  IF i=1 THEN SAY "AETE9NSHAXN.",Stimme
103:  IF i=2 THEN SAY "PRIHPAESAX FAO GEY7M.",Stimme
104:  IF i=3 THEN SAY "WAATSH AW9T.",Stimme
105:  IF i=4 THEN SAY "LEHTS GOW7.",Stimme
106:  IF rflag=0 THEN Memory
107:  R=INT(RND*4)
108:  IF R=0 THEN Memory
109:  IF R=1 THEN FOR i=0 TO 2:FOR j=0 TO 2:Ro(i,j)=Code(2
110:  -j,i):NEXT j,i
111:  IF R=2 THEN FOR i=0 TO 2:FOR j=0 TO 2:Ro(i,j)=Code(2
112:  -i,2-j):NEXT j,i
113:  IF R=3 THEN FOR i=0 TO 2:FOR j=0 TO 2:Ro(i,j)=Code(2
114:  -2-i):NEXT j,i
115:  FOR i=0 TO 2:FOR j=0 TO 2
116:    Code(i,j)=Ro(i,j)
117:    x=468+j*28:y=10+i*13:f=Code(i,j)*2
118:    GOSUB Inneres:NEXT j,i:ERASE Ro
119:
120:  Memory:
121:  FOR z=0 TO 1 STEP .008
122:    PALETTE 2,z,z,z*.6
123:    IF Lev=0 THEN PALETTE 6,z,1-z,0
124:  NEXT z
125:  IF mflag=0 THEN Start
126:  FOR i=1 TO 2
127:    SOUND 440,5,200,0:FOR t=0 TO 3000
128:      NEXT t,i:SOUND 440,18,220,0
129:    FOR i=0 TO 3800:NEXT i
130:    FOR i=0 TO 2:FOR j=0 TO 2
131:      x=468+j*28:y=10+i*13:f=0:GOSUB Inneres
132:    NEXT j,i
133:
134:  Start:
135:  k=10:x=6:y=6:GOSUB Cursor
136:  i=MOUSE(0):Fin=0
137:  ON TIMER (Zeitschalt) GOSUB Zeit
138:  TIMER ON
139:  Schleife:
140:  IF Fin THEN Timeout
141:  IF MOUSE(0)>0 THEN Pruefen
142:  xm=INT((MOUSE(1)-34)/28):ym=INT((MOUSE(2)-15)/13)
143:  IF xm=x AND ym=y THEN Schleife
144:  xm=xm+(xm<0)*xm-(xm>11)*(11-xm)
145:  ym=ym+(ym<0)*ym-(ym>11)*(11-ym)
146:  IF xm=x AND ym=y THEN Schleife
147:  GOSUB Cursorweg:x=xm:y=ym:GOSUB Cursor
148:  GOTO Schleife
149:
150:  Zeit:
151:  k=k+5:IF k >= 45 THEN Fin=1:RETURN
152:  CIRCLE (510,90),k,6:SOUND WAIT
153:  FOR t=100 TO 10 STEP -10
154:    SOUND 1046.5,5,t,0
155:  NEXT t:SOUND RESUME:RETURN
156:
157:  Cursor:
158:  GET (4+x*28,2+y*13)-(87+x*28,40+y*13),Fe
159:  PUT (4+x*28,2+y*13),Cu,OR
160:  RETURN

```

Listing: CODE-CRACKER

```

156:
157: Cursorweg:
158: PUT (4+x*28,2+y*13),Fe,PSET
159: RETURN
160:
161: Leuchtfeld:
162: LINE (x,y)-(x+1,y+12),3,b
163: LINE (x+1,y)-(x+27,y),4
164: LINE (x+27,y)-(x+26,y+12),4,b
165: LINE (x,y+12)-(x+26,y+12),3
166: RETURN
167: Inneres:
168: LINE (x+2,y+1)-(x+25,y+11),f,bf
169: RETURN
170:
171: Schalter:
172: LINE (x,y)-(x+1,y+12),3,b
173: LINE (x+1,y)-(x+145,y),4
174: LINE (x+145,y)-(x+144,y+12),4,b
175: LINE (x,y+12)-(x+144,y+12),3
176: LINE (x+2,y+1)-(x+143,y+11),0,bf
177: RETURN
178:
179: Pruefen:
180: IF x=x1 AND y=y1 THEN Treffer
181: a=0:b=0:c=0:d=0
182: FOR i=0 TO 2:FOR j=0 TO 2
183:   IF Code(i,j)=Feld(y+i,x+j) THEN a=a+1
184:   IF Code(2-j,i)=Feld(y+i,x+j) THEN b=b+1
185:   IF Code(2-i,2-j)=Feld(y+i,x+j) THEN c=c+1
186:   IF Code(j,2-i)=Feld(y+1,x+j) THEN d=d+1
187: NEXT j,i
188: IF rflag=0 THEN IF a=9 THEN Treffer:ELSE BEEP:GOTO Sc
189: hleife
190: IF a=9 OR b=9 OR c=9 OR d=9 THEN Treffer
191: BEEP:GOTO Schleife
192:
193: Fehler:
194: WINDOW 2:CLS
195: PALETTE 0,1,0,0:FOR i=0 TO 2000:NEXT i
196: PALETTE 0,0,0,0
197: IF ERR=5 THEN
198:   LOCATE 10,15:COLOR 6
199:   PRINT "Das Programm benoetigt die Datei narrator.dev
200:   ice"
201:   LOCATE 12,20:PRINT "im devs-Verzeichnis der Bootdisk
202:   ete !"
203:   ELSE
204:   RESUME
205:   END IF
206:   FOR i=0 TO 32000:NEXT i
207:   WINDOW 1:END
208:
209: Timeout:
210: TIMER OFF
211: FOR i=0 TO 1000:NEXT i
212: GOSUB Cursorweg
213: SAY "/HAA9 TAY7M IHZ AW8T.",Stimme
214: FOR i=0 TO 2:FOR j=0 TO 2
215:   x=468+j*28:y=10+i*13:f=Code(i,j)*2
216:   GOSUB Inneres:NEXT j,i
217:   x=x1:y=y1
218:   FOR i=1 TO 10
219:     GOSUB Cursor
220:     FOR j=15 TO 40 STEP 5
221:       CIRCLE (510,90),j,6:NEXT j
222:     GOSUB Cursorweg
223:     FOR j=40 TO 15 STEP -5
224:       CIRCLE (510,90),j,5:NEXT j,i
225:     i=Lev-2*mflag*(Lev>0)-4*rflag*(Lev>0)
226:     We$="YOHR GEY3M WAAZ "
227:     IF i=0 THEN We$=We$+"RIHDIH5KYUWLAXS.":Te$="ridiculo
228:     us !"
229:     IF i=1 THEN We$=We$+"MIH8ZERAXBUL.":Te$="miserable."
230:     IF i=2 OR i=3 THEN We$=We$+"NAA7T VEHI7RIY KWIH4K.":T
231:     e$="not very quick."
232:     IF i=4 OR i=5 THEN We$=We$+"AE4VERIHJ.":Te$="average
233:     ."
234:     IF i=6 OR i=7 THEN We$=We$+"NAAT BAE4D.":Te$="not ba
235:     d."
236:     IF i=8 OR i=9 THEN We$=We$+"RIH8PEH2KTAXBUL.":Te$="r
237:     espectable."
238:     IF i=10 OR i=11 THEN We$=We$+"VEH7RIY KWIH4K.":Te$="
239:     very quick."
240:     IF i=12 OR i=13 THEN We$=We$+"FAEINTAE7STIHK.":Te$="
241:     fantastic !"
242:     IF i=14 OR i=15 THEN We$=We$+"IH1NKRH7DAXBUL.":Te$=
243:     "incredible !"
244:     IF i>16 THEN We$=We$+"AH1N BIY9TAXBUL.":Te$="unbeata
245:     ble !!"
246:   WINDOW 3,"Results:",(71,83)-(327,123),8,2
247:   LOCATE 2,3:COLOR 4
248:   PRINT "You reached Level";Lev+1
249:   LOCATE 4,3
250:   PRINT "Your game was ";Te$
251:   SAY We$,Stimme
252:   WHILE WINDOW(7)<>0:WEND
253:   WINDOW 2
254:   FOR i=0 TO 1000:NEXT i
255:   Lev=0:Zeitschalt=8:GOTO Neu
256:
257: Treffer:
258: TIMER OFF
259: FOR i=0 TO 1000:NEXT i
260: i=INT(RND*3)
261: IF i=0 THEN SAY "YUWV KRAE5KT MAY KOW1D.",Stimme
262: IF i=1 THEN SAY "DHAE4TS AOL RAY8T.",Stimme
263: IF i=2 THEN SAY "YUW WIH8N DHIHS GEY4M.",Stimme
264: a=x:b=y
265: FOR i=0 TO 2:FOR j=0 TO 2
266:   x=468+j*28:y=10+i*13:f=Code(i,j)*2
267:   GOSUB Inneres:NEXT j,i
268:   FOR i=0 TO 8000:NEXT i
269:   x=a:y=b:GOSUB Cursorweg
270:   Lev=Lev+1:Zeitschalt=Zeitschalt*.75
271:   IF Zeitschalt<.1 THEN Zeitschalt=.1
272:   GOTO Neu

```

Listing: CODE-CRACKER

Postkarte genügt!

Das große AMIGA-DOS-Preis ausschreiben

Anlässlich der Messe "Amiga Expo '90" in Köln haben wir uns von der Redaktion für Sie, liebe Leser, etwas ganz Besonderes ausgedacht, nämlich ein großes Preis ausschreiben.

Die Teilnahme ist ganz einfach. Alles, was Sie tun müssen, ist uns eine Postkarte mit Ihrem Namen und Ihrer Adresse zu schicken. Jede Karte, die ausreichend frankiert, mit der vollständigen Adresse versehen und bis zum Einsendeschluß in den Briefkasten geworfen wird, nimmt garantiert an unserer Verlosung teil.

Fehlen nur noch die Preise, und die gibt es massenweise. Damit Sie wissen, was Ihnen im Falle des Gewinnens droht, hier nun eine Aufstellung:

- Von der Firma Digital Marketing fünf »Intro-Maker« und 20 Poster von letzter Repro-Party
- Von A.P.S.-Electronic drei »Power Packer 3.0« sowie diverse Soundtracker- und Intro-Maker-Programme

- Von Data Becker fünf Exemplare von »Beckertext 2« sowie fünf Shareware-Bücher
- Von Gigatron eine Festplatte Arriba HD20, eine Speichererweiterung 2 MByte für den A2000(neu), eine MiniMax-Speichererweiterung 2MB
- Von Jin Tech eine optische Maus, eine Speichererweiterung für den A 500
- Von Alcomp ein EPROMmer mit EPROM-Bank
- Von Pulsar eine kabellose Maus, das Spracherkennungsprogramm »Vorec One«, sowie 10 Programme »Cross-Dos«
- Von 3State eine 2MB-Karte für A500 (A580 Plus)
- Von HK-Computer eine Erweiterung 2,5 MByte für A500 sowie eine Speichererweiterung 1MB, fünf "Amiga-Bremsen" und fünf »Power-Fire«
- Von TKR ein 2400-Baud-Modem und dreimal das Programm »MultiTerm PRO«
- Von Roßmüller eine Speichererweiterung 2MB für A500, sowie ein Tornado-Board, ebenfalls für A500
- Von IrseeSoft gibt es dreimal das Programm »TurboPrint Professional«
- Die Firma Diezemann stiftet einen Digitizer mit RGB-Splitter

Turtle
Byte

AL
ALCOMP

3-State
Computertechnik

GIGATRON®

ROSSMÖLLER
COMPUTER TECHNIK

profitteam

MAXON
computer gmbh

bsc büroautomation

TKR

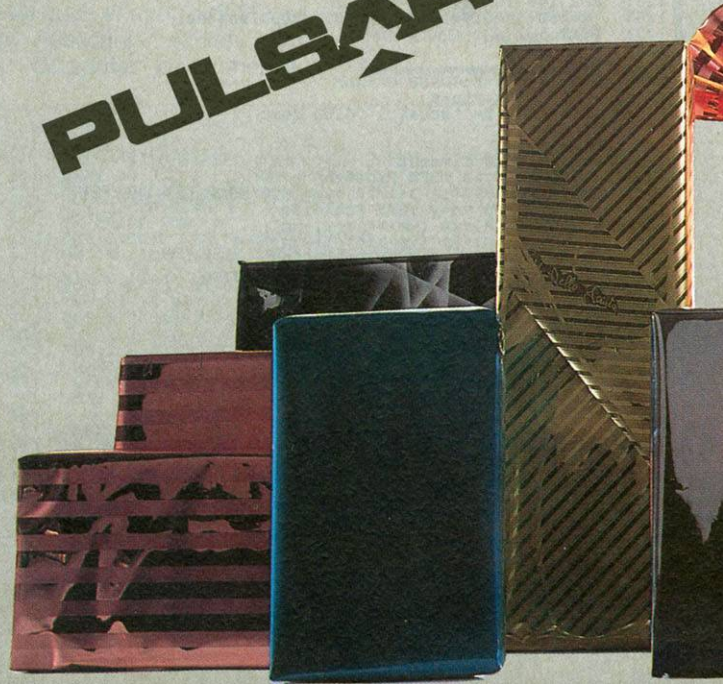
PULSAR

GTI



KCS®

DATA BECKER



WETTBEWERB

- Von der Firma Miky Wenngatz gibt es einen MW 500-Umbausatz und Joystickumschalter
- Die Firma United Software spendet zehn »Might&Magic II« und T-Shirts
- Von der Firma Rushware gibt es 20 Softwarepakete
- Die Firma KCS spendet sechs PC-Power-Boards
- Von der Firma Epson gibt es einen Drucker LQ 400
- Maxon spendiert fünf KickPascal
- Von Profiteam Software gibt es fünfmal das Programm »Soundfactory«
- Die Firma Reisware spendet einen Schwarzweiß-Handyscanner
- Von Merkens EDV gibt es ein Genlock »MiniGen«
- Vom technicSupport-Verlag gibt es ein Goldpaket, einen AmigaDOS-Lernkurs, einmal die PD-Schatztruhe, ein Exemplar des »Desktop Publishing mit Amiga« und einmal »Digitalisieren mit Amiga«
- Die Firma Turtle Byte spendet 50 Software-Produkte aus ihrer Programmpalette
- Von BSC gibt es eine Multi-IO-Karte, einen Alf3-SCSI-Controller und die THI Tools
- HS&Y stellt ihre Produkte »AMTrack«, »Tiger Cub« und »Video Page« zur Verfügung

- Die Firma Eurosystems spendet fünf Action Replay
 - Von Gold Vision gibt es zehn Exemplare des Programms »VectorTrace«
 - GTI stiftet 5 mal »XCopy Professional«
 - Jochheim beteiligt sich mit einer Erweiterung für A2000 2MB
 - Tröps&Hierl stellt eine Erweiterung 1,8 MB bereit
- Also, wer bei diesen Preisen nicht mitmacht, ist selbst schuld. Wir jedenfalls von der AMIGA DOS würden am liebsten dabei sein, doch wie immer gilt auch hier: Mitarbeiter des DMV-Verlages dürfen nicht teilnehmen, der Rechtsweg ist ausgeschlossen. Doch nun heißt es für Sie schnellstens die Karte ausfüllen und einsenden an den

DMV-Verlag
Redaktion AMIGA DOS
Kennwort: Postkarte genügt!
Postfach 250
D-3440 Eschwege

Einsendeschluß ist der 10.11.1990, es gilt das Datum des Poststempels. Die Gewinner werden automatisch von uns benachrichtigt. Bleibt uns von der Redaktion nur noch, Ihnen, liebe Leser, viel Glück zu wünschen.

(jb)



EUROSYSTEMS



IrseeSoft SPCS

Merkens EDV

technicSupport

JIN TECH ELECTRONICS CORP.

EPSON

HS&Y

Digital Marketing
 Software Herstellung und Vertrieb

Wir bedanken uns bei allen hier abgebildeten Firmen für die Bereitstellung der Preise.

VIDEOTECHNIK DIEZEMANN



Andreas Polk

Freie Fahrt für 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

Die Definition eigener Datentypen kann die Arbeit mit Modula2 und seinen spezifischen Datentypen sehr vereinfachen.

In diesem Kursteil behandeln wir weitere Datentypen wie Aufzählungstypen, Mengen, Records und Felder. Ein eigener Datentyp kann nur nach dem Import von Modulen und vor der Variablendeklaration geschehen. Dieser Programmteil wird eingeleitet durch das Schlüsselwort »TYPE«. Anschließend kommt eine beliebige Anzahl von Typendefinitionen, die immer folgende Syntax hat:

```
neuerTyp = alterTyp;
```

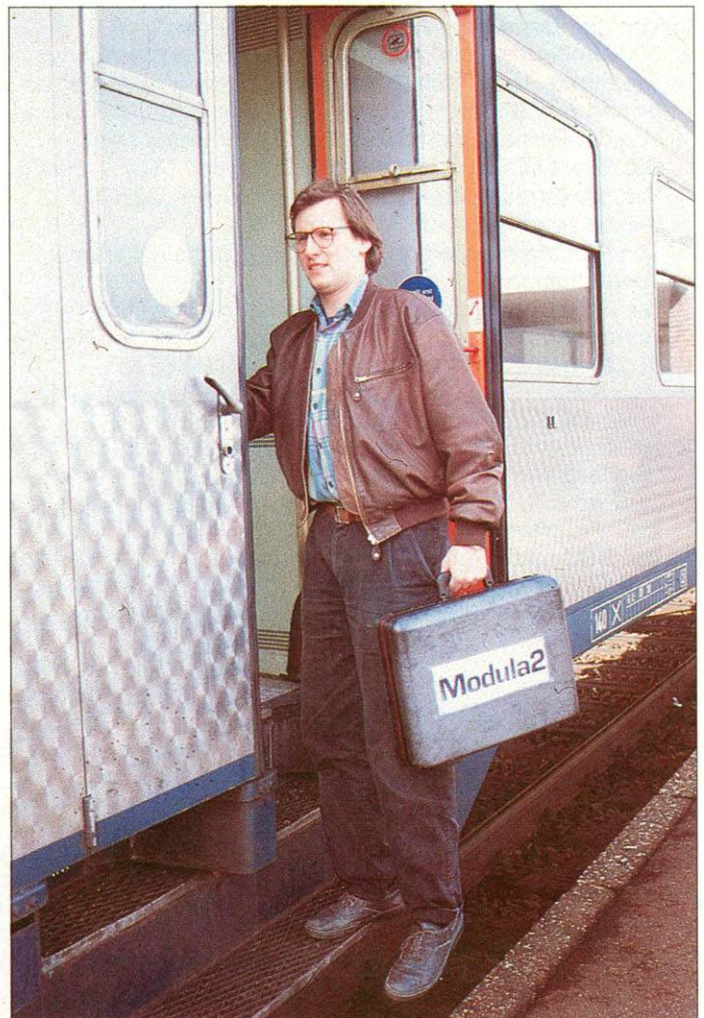
Die Anweisung »TYPE« wird bei der Verwendung eines neuen Typen fast immer benötigt. Dazu wird in der Definition nur der neue Typ an Stelle eines Basistyps angegeben. Kommen wir nun zu den neuen Variablentypen. Als erstes sei hier der Unterbereichstyp aufgeführt. Bei vielen Anwendungen kann man schon vorausagen, welchen Wertebereich eine Variable annehmen kann. Definiert man beispielsweise eine Variable, die die Tage eines Monats aufnehmen soll, so sollte diese Variable nur

Werte zwischen 1 und 31 annehmen können, denn den 32. eines Monats gibt es ja bekanntlich nicht.

Bisher benutzten Sie für diese Aufgabe eine Variable vom Typ »CARDINAL«. Allerdings kann eine solche Variable auch Werte, die größer als 31 sind und natürlich den Wert 0 annehmen. An dieser Stelle setzt der Unterbereichstyp ein. Ein Unterbereich ist ein Variablentyp, der eine Teilmenge eines schon bestehenden Variablentyps annehmen kann. Zur Definition eines Unterbereiches ist folgende Syntax nötig:

```
VAR Unterbereich : Variablentyp[untereGrenze..obereGrenze]
```

»Unterbereich« ist der Name des neuen Typs, Variablentyp der Basistyp, also der Typ, aus dem der Unterbereich gebildet wird. Innerhalb der eckigen Klammern müssen dann die Grenzen dieses Unterbereichstyps angegeben werden, dazwischen zeigen zwei Punkte, daß alle Werte zwischen diesen Grenzwerten auch gültig



sind. Die Variable kann dann alle Werte zwischen diesen beiden Grenzen annehmen. Bleiben wir bei unserem Beispiel. Der Variablen »Monat« sollen nur Werte zwischen 1 und 31 zugewiesen werden können. Dazu ist folgende Definition nötig:

```
VAR Monat : CARDINAL[1..31];
```

Somit kann die Variable »Monat« nur Werte des Variablentyps »CARDINAL« annehmen, die größer oder gleich eins und kleiner oder gleich 31 sind. Bei allen anderen Werten würde das Programm mit einem Laufzeitfehler abbrechen. Würden Sie im Listing sogar »Monat := 32;« schreiben, so würde der Compiler diese Bereichsüberschreitung schon merken und eine entsprechende Fehlermeldung ausgeben. Ein Laufzeitfehler entsteht immer dann, wenn für den Compiler diese Bereichsüberschreitung nicht sichtbar war, beispielsweise, wenn der Variablen »Monat« der Wert 31 zugewiesen wird (was noch erlaubt ist), und diese Variable dann durch

»INC(Monat)« um eins erhöht würde.

Sollen mehrere Variablen von diesem Unterbereichstyp sein, so ist es sinnvoll, einen neuen Typ Monat zu definieren. Anschließend kann dann verschiedenen Variablen dieser Typ zugewiesen werden. Ein Beispielprogramm für den Unterbereichstyp sehen Sie in Listing 1. Hier wird der Unterbereichstyp einer Cardinalzahl unter dem Namen »Monat« definiert. Anschließend wird im Deklarationsteil eine Variable »Monatstage«, die vom Typ »Monat« ist (Cardinalzahlen zwischen 1 und 31), deklariert. Bei der Definition eines Unterbereichstypes ist es oft überflüssig, den Basistyp anzugeben. So kann auch folgende Definition stehen: »TYPE Monat = [1..31];«. Der Compiler sucht nun selbständig nach einem geeigneten Basistyp. Da »Monat« hier nur Werte zwischen 1 und 31 annehmen soll, würde er automatisch als Basistyp »CARDINAL« nehmen. Die Angabe des Basis-

typs ist eigentlich nur dann sinnvoll, wenn der Compiler einen anderen Basistyp wählen würde. Nehmen wir einmal an, Monat sollte ein Unterbereich vom Typ »INTEGER« sein, so müßten Sie folgende Definition angeben: »TYPE Monat: INTEGER[1..31];«. Hier muß der Basistyp angegeben werden, denn sonst würde der Compiler automatisch »CARDINAL« als Basistyp wählen.

Einen gänzlich neuen Typ stellt der Aufzählungstyp dar. Er hat zwei Aufgaben: Zum einen dient er zur Abstraktion des Programmes, und zum anderen wird er benötigt, um einzelne Bits in einem Wort oder Langwort anzusprechen. Der Aufzählungstyp wird besonders in der Systemprogrammierung gebraucht. Schauen wir uns zuerst einmal die Syntax an: »TYPE Aufz = (Element1, Element2, ..., Element n);«. Auch hier wurde wieder eine Typendefinition gewählt, ein neuer Typ namens »Aufz« wird als Aufzählungstyp definiert. Dieser Typ besteht aus der Anzahl »n« von Elementen, die in Klammern und durch Kommata getrennt angegeben werden. Damit Sie diesen Variablentyp besser verstehen, ist etwas Hintergrundwissen notwendig. Wie Sie vielleicht wissen, ist die kleinste Einheit beim Computer das Bit. Ein Bit kann genau zwei Zustände annehmen, nämlich "gesetzt" (1) oder "nicht gesetzt" (0). Beachten Sie dabei die Analogie zum Variablentyp »BOOLEAN«! Eine Variable dieses Datentyps kann ebenfalls zwei Zustände annehmen, nämlich »TRUE« und »FALSE«.

Ein Byte setzt sich nun aus 8 Bits zusammen. Es gibt nun 256 (2¹⁷) Möglichkeiten, die acht Bits eines Bytes anzuordnen, somit kann ein Byte 256 Zustände annehmen. Hat ein Byte den Wert Null, so ist kein Bit gesetzt, hat es den Wert 255, sind alle Bits gesetzt. Die Elemente eines Aufzählungstyps stellen also nichts anderes als die Bits eines "Wortes" (16 Bit = 2 Byte) dar. Somit ist die Anzahl der Elemente eines Aufzählungstyps auch auf 16 beschränkt. Es ist nicht möglich, einem Aufzählungstyp mehr als 32 Elemente zuzuordnen. In Listing 3 habe ich Ihnen ein Programm aufgelistet, in welchem versucht wird,

einen Aufzählungstyp mit mehr als 32 Elementen zu definieren. Der Compiler merkt dies, und gibt eine entsprechende Fehlermeldung aus. In welcher Weise repräsentieren die einzelnen Elemente eines Aufzählungstyps nun die Bits eines Wortes beziehungsweise eines Langwortes?

Das erste Element stellt das erste Bit, Bit Nummer 0 dar, das zweite Element somit Bit Nummer 1, das dritte Bit Nummer 2 und das 16. Element letztlich Bit Nummer 15. Werden mehr als 16 Elemente angegeben, so repräsentiert das 17. Element Bit Nummer 16 des Langwortes oder das Bit Nummer 0 des zweiten Wortes. (Ein Langwort setzt sich aus zwei Worten oder vier Bytes zusammen.)

Obwohl dieser Datentyp nicht einfach zu handhaben ist, findet dieses Prinzip in der Systemprogrammierung sehr häufig Anwendung und man sollte die Hintergründe dieses Datentyps schon kennen.

Schauen wir uns nun zusammen einmal ein Beispielprogramm zu diesen Datentyp an (Listing 2). Zuerst wird ein neuer Variablentyp namens »TextAttr« definiert. Dieser Aufzählungstyp besteht aus acht Elementen. Hierzu wieder etwas Hintergrundwissen. Das CLI stellt sich dem Benutzer zwar recht eintönig dar, doch das muß nicht so sein. Der Amiga bietet auf dem Workbenchscreen die Möglichkeit, Text in vier Farben darzustellen. Dies sind (bei den Standardeinstellungen der Preferences) Weiß, Blau, Rot und Schwarz. Außerdem können den Texten noch drei verschiedene Textattribute zugewiesen werden, und zwar »unterstrichen«, »kursiv« und »fett«. Im CLI läßt sich ein Text durch die sogenannten Escape-Codes verändern. Eine Liste dieser Codes können Sie der Tabelle 1 entnehmen. Probieren Sie einfach einmal diese Codes aus, sie lassen sich im CLI durch die Zeichenkombination »*e[« erreichen; der Befehl »echo "e[30m Hallo Welt..."« zeigt das berühmte Begrüßungswort in der Preferences-Farbe 0 (bei den Original-Farben blau).

Das Aussehen eines Textes kann jedoch nicht nur im CLIFenster direkt beeinflußt werden, sondern auch von einem Programm aus. Wir haben also

in Listing 2 einen Aufzählungstypen definiert, der diese acht Attribute annehmen kann. Anschließend wird im Deklarationsteil die Variable »Attr« als Aufzählungstyp deklariert.

Schauen wir uns nun einmal die Zuweisung eines Elementes solch eines Aufzählungstyps an. Solch eine Zuweisung findet direkt in der ersten Programmzeile statt. Hinter dem »:=«, welches eine Zuweisung an eine Variable kennzeichnet, wird einfach in Klammern ein Element dieses Aufzählungstyps angegeben. Die Variable »Attr« beinhaltet nun das Element »weiss«.

Modula2 bietet vier Befehle, um Aufzählungstypen zu beeinflussen. Diese Befehle lauten »ORD«, »VAL«, »INC« und »DEC«. Der Befehl »ORD« hat folgende Syntax: »Intzahl:= ORD(Menge);«. In Klammern muß eine Variable vom Typ »Aufzählungstyp« angegeben werden. »ORD« liefert als Ergebnis das Bit, welches dieses Element repräsentiert. Das Ergebnis wird als »INTEGER« zurückgegeben. In Listing 2 wird in der zweiten Programmzeile die Funktion »ORD« aufgerufen, welche der Prozedur »Writeln« eine Zahl zurückgibt, die dann ausgegeben wird. Lassen Sie das Programm einmal laufen, so wird »0« ausgegeben. Das Element »weiss« des Aufzählungstyps »TextAttr« repräsentiert also das Bit Nummer 0.

Genau das Gegenteil von »ORD« liefert »VAL«. Die Syntax dieser Funktion lautet:

```
Element := VAL(Aufzählungstyp, Bitnummer);
```

»VAL« braucht zwei Elemente, nämlich einmal die Angabe des Aufzählungstyps und die Angabe der Bitnummer. Zurückgegeben wird das Element, welches die angegebene Bitnummer des Aufzählungstyps repräsentiert.

Die beiden Prozeduren »INC« und »DEC« sind Ihnen ja schon bekannt. Hier noch einmal ihre Syntax:

```
INC (Variable, Schritt);
```

»INC« erhöht den Wert einer Variable um den angegebenen Schritt, »DEC« verringert ihn. Wird die Schrittweite nicht mit angegeben, so wird eine Schrittweite von eins angenommen. In Listing 2 sehen Sie noch Beispiele für die Anwendung dieser beiden und

der davor genannten Funktionen. Die Bitnummer, die das Element gerade präsentiert, wird immer im Kommentar angegeben. Anschließend wird noch einmal die Bitnummer ausgegeben.

Text kann jedoch auch mehrere Attribute annehmen. So ist es beispielsweise möglich, einen Text gleichzeitig unterstrichen und fett auszugeben. Allerdings kann eine Variable vom Aufzählungstyp immer nur ein Element annehmen. Wir brauchen also einen Variablentyp, welcher mehrere Elemente eines Aufzählungstyps aufnehmen kann.

Modula2 bietet auch solch einen Datentyp. Hierbei handelt es sich um die Menge. Eine Variable, die vom Typ »Menge« ist, kann beliebig viele Elemente eines Aufzählungstyps aufnehmen. Schauen wir uns wieder zuerst die Syntax zum Definieren einer Menge an:

```
TYPE Aufz = (Element1, Element2, ..., Element n);
AufzSet = SET OF Aufz;
```

In der ersten Zeile wird einfach ein Aufzählungstyp definiert. Um eine Menge zu definieren, die als Basis die Elemente des Aufzählungstyps beinhaltet, benutzt man die Befehlswörter »SET OF«. Somit kann die Variable »AufzSet« beliebig viele Elemente des Aufzählungstyps »Aufz« annehmen. Ein Aufzählungstyp muß vorher definiert sein, ansonsten weiß der Compiler nicht, welche Elemente in der Menge enthalten sein sollen. Sie sehen, Aufzählungstypen und Mengen hängen ganz eng miteinander zusammen.

Kommen wir nun zu einem sinnvollen Anwendungsbeispiel für eine Menge. In Listing 4 sehen Sie die Realisierung des eben genannten Beispiels. Eine Menge wird definiert, welche alle möglichen Attribute eines auszugebenen Textes enthält. Diese Menge wird einer Prozedur übergeben, die

Wichtig: Aus satztechnischen Gründen lassen sich unsere Beispiel-Programme nicht immer so einrücken, wie es Modula2 (und der gute Programmierstil) vorschreibt. Beachten Sie bitte deshalb beim Eingeben die Hinweise aus Ihrem Modula2-Handbuch.

die Textattribute so ändert, daß sie den gesetzten Elementen dieser Menge entspricht.

Nach der Angabe des Modulnamens und der Importliste folgen zwei Typendefinitionen. Zuerst wird ein Aufzählungstyp definiert, der 21 Elemente enthält. »v0« steht für Vordergrundfarbe 0, »v1« für Vordergrundfarbe 1, »h0«, »h1«, »h2« und »h3« für die entsprechenden Hintergrundfarben. Bei den Standardeinstellungen von Preferences entsprechen diese Farben den angegebenen Farben Weiss, Blau, Rot und Schwarz. Diese Elemente vereinfachen demjenigen die Arbeit mit diesem Typ, der nicht mit den Standardfarben arbeitet. In der zweiten Anweisung wird eine Menge namens »AttrFlagSet« definiert, welche auf dem Aufzählungstyp »AttrFlags« basiert. »AttrFlagSet« kann beliebig viele Elemente des Aufzählungstyps »AttrFlags« aufnehmen.

Im Deklarationsteil wird nur eine Variable namens »Attribute« vom Typ »AttrFlagSet« deklariert. Die Prozedur »SetzeAttr« wollen wir erst mal übergehen. Schauen wir uns also das Hauptprogramm an. Zuerst werden der Variablen »Attribute« Elemente der Menge zugeordnet. Hinter dem »:=«, welches eine Zuweisung bedeutet, folgt zuerst die Angabe, um welche Menge es sich handelt. In geschweiften Klammern werden dann die einzelnen Elemente angegeben, die diese Menge beinhalten soll. Diese müssen durch Kommata voneinander getrennt werden. Sollen einmal alle Elemente einer Menge gelöscht werden, so übergibt man einfach die

leere Menge, die keine Elemente enthält. Bei dem Beispiel sähe solch eine Zuweisung folgendermaßen aus:

```
Attribute := AttrFlagSet{};
```

»Attribute« wird nun kein Element mehr beinhalten. Sollen einer Menge Teilbereiche eines Aufzählungstyps zugewiesen werden, so muß man die untere und obere Grenze angeben und die Elemente dazwischen durch zwei Punkte andeuten. Die Anweisung

```
Attribute := AttrFlagSet[fett
..unterstrichen, vblau, hwarz
];
```

würde »Attribute« die Elemente »fett«, »kursiv«, »unterstrichen«, »vblau« und »hschwarz« zuweisen. In der zweiten Zeile des Programmes wird dann die Prozedur »SetzeAttr« aufgerufen, und in der dritten Zeile wird ein Probetext ausgegeben. In Zeile vier wird die Prozedur »SetzeAttr« nochmals aufgerufen. Sie sehen, bei der Parameterangabe kann man auch direkt die Elemente der Menge definieren. Anschließend wird ein weiterer Text ausgegeben und das Programm beendet. Mit der Zuweisungsart, die Sie jetzt kennen, wird immer der Inhalt einer Menge ganz neu bestimmt. Dies ist jedoch manchmal umständlich, beispielsweise, wenn zu einer Menge, die schon sehr viele Elemente beinhaltet, ein weiteres hinzugefügt werden soll. Dazu dient der Befehl »INCL«. Er hat folgende Syntax:

```
INCL (Menge, Element);
```

Mit Hilfe dieses Befehles wird der angegebenen Menge das angegebene Element hinzugefügt. Beinhaltet die Menge

Attribute beispielsweise die Elemente unterstrichen und kursiv, so fügt der Befehl »INCL(Attribute, hschwarz);« das Element »hschwarz« zu den bereits bestehenden Elementen hinzu. Der Befehl »Attribute := AttrFlagSet[hschwarz];« würde hingegen die beiden Elemente »unterstrichen« und »kursiv« aus der Menge entfernen und dann erst »hschwarz« hinzufügen. Somit besäße »Attribute« im ersten Beispiel die drei Elemente »unterstrichen«, »kursiv« und »hschwarz«, im zweiten nur »hschwarz«.

Möchten Sie ein Element aus einer Menge entfernen, so dient dazu der Befehl »EXCL«. Seine Syntax lautet:

```
EXCL(Menge, Element);
```

Hier gilt das gleiche wie für den Befehl »INCL«. Natürlich macht eine Menge mit Elementen nicht viel Sinn, wenn man nicht abfragen kann, welche Elemente eine Menge enthält. Hierzu dient der Operand »IN«, welcher überprüft, ob eine Menge das angegebene Element enthält. Als Ergebnis wird »TRUE« geliefert, wenn die Menge das Element enthält, und »FALSE«, wenn es nicht in der Menge enthalten ist.

Schauen wir uns dazu einmal die Prozedur »SetzeAttr« an. Ihr wird eine Menge vom Typ »AttrFlagSet« übergeben. Mit der Anweisung

```
IF (normal IN Flags) THEN
```

wird nun überprüft, ob »Flags« das Element »normal« beinhaltet. Ist dies der Fall, so wird eine Konstante ausgegeben. Diese besteht aus dem Escape-Code, der den Text wieder normal darstellt. So

wird nacheinander überprüft, ob in der Menge die entsprechenden Elemente enthalten sind. Ist dies der Fall, so werden entsprechende Escape-Codes ausgegeben. Auch Vergleiche sind mit Mengen des gleichen Typs möglich. Voraussetzung ist jedoch, daß die Mengen vom gleichen Typ sind. Hier eine Auflistung der Operatoren:

– »Menge1 = Menge2;« liefert »TRUE«, wenn »Menge1« die gleichen Elemente beinhaltet wie »Menge2«.

– »Menge1 <= Menge2;« liefert »TRUE«, wenn »Menge1« in »Menge2« enthalten ist, »Menge1« also Teilmenge von »Menge2« ist.

– »Menge1 >= Menge2;« liefert »TRUE«, wenn »Menge2« in »Menge1« enthalten ist, »Menge2« also Teilmenge von »Menge1« ist.

– »Menge1 # Menge2;« »ge1 < > Menge2;« beide Vergleiche liefern »TRUE«, wenn beide Mengen nicht die gleichen Elemente beinhalten.

Zum Abschluß möchte ich zwei Definitionen aus dem Modul »SYSTEM« erläutern. Möchten Sie Bits eines Wortes oder Langwortes direkt ansprechen, so können Sie aus dem Modul »SYSTEM« die Typen »BITSET« und »LONGSET« importieren. »BITSET« enthält alle Elemente von 0 bis 15, welche die entsprechenden Bits repräsentieren, »LONGSET« enthält alle Elemente von 0 bis 31. Somit sollte »LONGSET« nur dann benutzt werden, wenn die Bits eines Langwortes beeinflußt werden sollen. Kommen wir zu dem nächsten Datentyp, den Feldern. Stellen Sie sich einmal vor, Sie haben

Anzeigenschluß

für Ausgabe 1/91
der AMIGA DOS
ist der

7. November 1990

AMIGA-BUREAU

– preiswerte und leistungsstarke Software –

AUFTRAG

FIBU

TEXT/ADR.

Faktura ● Lager-
kunden ● OP

Sach- ● Personenkonten
UVA ● Bilanz ● G + V
Saldenlisten

Textverarbeitung
Adreßverwaltung

EDOTRONIK

St.-Veit-Straße 70
D-8000 München 90
☎ 089 / 40 40 93
Fax 089 / 40 22 93

eine Menge von gleichartigen Daten abzuspeichern, beispielsweise zehn Cardinal-Zahlen. Sie könnten nun bei der Variablendeklaration zehn Variablen von »Zahl1« bis »Zahl10« definieren, das aber wäre umständlich.

Aus diesem Grunde wurde der Datentyp »Feld« geschaffen. Ein Feld ist eine Ansammlung von mehreren Daten gleichen Variablentyps. Um ein Feld zu definieren, müssen Sie folgende Syntax verwenden:

```
TYPE Feld = ARRAY[untereGrenz
e..obereGrenze] OF Typ;
```

»Feld« ist hierbei der Name des neu definierten Datentyps. Nach dem Gleichheitszeichen folgt das Befehlswort »ARRAY«, welches die Felddefinition einleitet. Natürlich muß der Compiler auch wissen, wieviel Daten das Feld aufnehmen kann. Dies geschieht durch die Angabe von einer unteren und oberen Grenze, dem sogenannten Index. Der Index kann vom Typ »INTEGER«, »CARDINAL«, »CHAR«, »BOOLEAN«, ein Aufzählungstyp oder ein Unterbereich sein. Somit bestimmt also die Angabe der oberen und unteren Grenze, wieviele Daten in dem Feld abgelegt werden können. Anschließend muß nur noch festgelegt werden, von welchem Typ die Daten sind, die in dem Feld abgelegt werden. Dies geschieht durch die Angabe »OF Typ«. Hier sind alle Typen erlaubt, die Modula2 unterstützt. Schauen wir uns einmal eine Definition eines Feldes in Listing 5 an. Der Sinn dieses Programmes ist es, zu einer Anzahl von Zahlen das Quadrat zu errechnen. Ähnlich könnte man eine Funktion wie »f(x)=x²« realisieren. In der Type-Anweisung wird ein Variablentyp namens »Graph« definiert, der ein Feld aus 11 Cardinalzahlen darstellt. Auf das Feld kann durch die Angabe eines Indexes zugegriffen werden, der kleiner oder gleich Zehn und größer oder gleich Null sein muß.

Um dies zu verstehen, schauen Sie sich bitte einmal die Angabe innerhalb der eckigen Klammern an. Als untere Grenze wird eine Konstante angegeben, die vorher definiert wurde und den Wert »0« beinhaltet. Die obere Grenze wird durch die Konstante

»Max« definiert und bekommt den Wert »10«. Somit kann das Feld 11 Daten aufnehmen, die vom Typ »CARDINAL« sein müssen, was hinter der Anweisung »OF« angegeben ist.

Natürlich müssen wir auch die Daten eines Feldes beeinflussen können, dabei gibt man nach Angabe des Variablennamens den Index in Klammern an. Möchten Sie beispielsweise auf das sechste Datum des Feldes zugreifen, so müssen Sie »Graph1[5] := Wert;« angeben. »Graph[5]« stellt hier das sechste Datum dar, da wir beim Index mit Null angefangen haben.

Betrachten wir also noch einmal das Beispielprogramm. In einer Schleife, in der »i« alle Werte zwischen Null und Zehn annimmt, wird jedem Datum des Feldes das Quadrat seines Indexes zugewiesen. So erhalten wir ein Feld, welches als Datum jeweils das Quadrat des Indexes beinhaltet. Erinnern Sie das auch stark an die Funktionsdarstellung in der Mathematik?

Bleiben wir gleich einmal bei Mathe. Wie Sie vielleicht wissen, gibt es auch Graphen, die dreidimensional sind. Hierzu wird zu jedem »x« (dargestellt durch den Index) ein »y« und »z« abgebildet. Auch das können wir in Modula2 darstellen. Die Definition von mehrdimensionalen Feldern ist nämlich ohne weiteres möglich, da nur mehrere Indexbereiche angegeben werden müssen. Die Syntax für solch eine Definition lautet:

```
TYPE Feld = ARRAY[untereGr
enze1..obereGrenze1], [untere
Grenze2..obereGrenze2], ..., [
untereGrenzen..obereGrenzen]
OF Typ;
```

Somit wird ein n-dimensionales Feld definiert, dessen Daten Werte des Typs »Typ« annehmen können. Auch auf solch ein Feld wird zugegriffen durch »Feld[Index1, Index2, ..., Indexn] := Wert;«. Das angesprochene Feld wird also durch die Indizes bestimmt. Dabei müssen immer genau »n« Indizes angegeben werden, ansonsten kann das Datum des Feldes nicht genauer bestimmt werden.

Ist Ihnen eigentlich aufgefallen, daß Sie schon vom ersten Kursteil an mit Feldern gearbeitet haben? Nein, dann überlegen Sie sich einmal, was denn Zeichenketten sind.

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**

und **Donnerstag** in der Zeit von **16.30 bis 18.00 Uhr** zur Verfügung, um Ihre Fragen zu beantworten. Wählen Sie einfach eine der nachfolgenden Nummern, um direkt den betreffenden Redakteur zu erreichen.



Jürgen Borngießer: 05651/809-742

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

Richtig, sie sind nichts anderes als eindimensionale Felder, deren Daten aus Variablen vom Typ »CHAR« bestehen. Somit kann man in einem Feld »VAR Text : ARRAY[0..5] OF CHAR;« das Wort »Hallo« unterbringen. Beachten Sie, daß ein String immer mit dem Zeichen »OC« abgeschlossen wird. Dies ist sehr wichtig. Ein Sondertyp des Feldes stellt das offene Feld dar. Solch ein Feld wird im Zusammenhang mit Prozeduren verwendet, die als Parameter einen Text brauchen. Natürlich kann man ein genügend großes Feld definieren; dieses sähe in etwa so aus:

```
PROCEDURE Demo(Text : ARRAY[0
..255] OF CHAR);
```

Was wäre jedoch, wenn dieser Prozedur ein String übergeben würde, der länger als 256 Zeichen wäre. Das Programm würde mit einer Fehlermeldung abbrechen.

Da dies nicht erwünscht ist, gibt es die Möglichkeit, ein offenes Feld zu definieren. Dessen Größe wird dann erst beim Ablauf des Programmes bestimmt. Ein offenes Feld wird durch das Weglassen der oberen und unteren Grenzen definiert. Die Prozedur »Demo« sähe mit einem offenen Feld als Parameter folgendermaßen aus:

```
PROCEDURE Demo(Text : ARRAY OF
CHAR);
```

Der Prozedur »Demo« kann nun ein beliebig langer String übergeben werden. Beim Aufruf der Prozedur wird der benötigte Speicherplatz reserviert und der Text darin gespeichert. So muß der Programmierer nicht vorher wissen, wie lang die zu übergebene Zeichenkette ist.

Bei der Verwendung von offenen Feldern sollten Sie aber zwei Dinge beachten. Erstens können offene Felder nur bei der Parameterübergabe benutzt werden. So kann man nicht einen Variablentyp »STRING« beliebiger Länge durch

TYPE String = ARRAY OF CHAR; definieren, dies akzeptiert der Compiler nicht. Außerdem wird ein String, der in einem offenen Feld abgelegt wird, **nicht** mit »OC« beendet. Dies ist unbedingt zu beachten. Listing 6 zeigt Ihnen daher den Gebrauch eines offenen Feldes. Der Prozedur »Textlaenge«

wird ein String beliebiger Länge übergeben, die Prozedur gibt diesen String dann in Anführungsstrichen aus, anschließend wird die Länge des Strings ermittelt und ausgegeben. An diesem Beispiel können Sie sehen, wie die Prozedur »WriteString« programmiert werden könnte.

Soviel zu den Feldern, kommen wir nun zu dem letzten noch unbekannten Datentyp, dem »Verbund«, auch »Record« genannt. Der Datentyp »Record« gleicht die Nachteile, die der Typ »Feld« mit sich bringt, aus. Ein Feld kann nämlich immer nur aus einer Ansammlung von Daten gleichen Typs sein. So kann man nicht in einem Feld eine Integerzahl und eine Cardinalzahl gleichzeitig ablegen, geschweige denn einen String und eine Menge. Oft ist aber auch diese Ansammlung von Daten verschiedener Datentyps sehr sinnvoll. Nehmen Sie einmal an, Sie schreiben eine Datenverwaltung, wie »DVW« aus dem ersten Kursteil, und müssen den Namen, die Straße, den Wohnort, die Telefonnummer und eine Bemerkung abspeichern. Natürlich könnten Sie für jedes einzelne Datum eine Variable anlegen. Dies zeigt aber noch nicht die Zusammengehörigkeit der Daten, die aber wichtig ist, denn was hat man von einer Straßenangabe, wenn man nicht weiß, wer in dieser Straße wohnt?

Also muß ein weiterer Datentyp geschaffen werden, der »Record«. Wie der Name schon sagt, ist ein »Record« eine Zusammenstellung von Daten verschiedenen Datentyps. Die Definition eines Verbunds sieht folgendermaßen aus:

```
TYPE Verbund = RECORD
  Feld1 : Typ1;
  Feld2 : Typ2;
  . . .
  . . .
  Feldn : Typn;
END;
```

Die Definition eines Verbundes wird durch das Befehlswort »RECORD« eingeleitet. Danach werden einfach die Datentypen und deren Namen angegeben, als handle es sich um eine Variablendeklaration. Innerhalb eines Verbundes können beliebig viele Daten angegeben werden. Auch

die Typen der Daten sind nicht festgelegt. Der Anwender kann hier sogar auch selbstdefinierte Datentypen angeben. Beendet wird eine Verbunddefinition durch das Befehlswort »END«.

Eben habe ich schon die Datenverwaltung »DVW« aus dem ersten Kursteil angesprochen. Genau dort habe ich nämlich ein »RECORD« zum Abspeichern der Daten benutzt. Schauen wir uns die Definition dieses Verbunds einmal genauer an (Listing 7).

Bei der Typendefinition werden zuerst sechs Strings definiert, wovon die ersten fünf für den Verbund benötigt werden. Hierbei handelt es sich um Felder, die aus einer angegebenen Anzahl von Zeichen besteht. Dann folgt die Definition des Verbundes »Datensatz«. Sie sehen, anhand des Verbundnamens kann man schon sehr gut erkennen, welche Daten in ihm abgelegt werden sollen. Nach Angabe des Befehlswortes »RECORD« folgt die Aufzählung der Datentypen dieses Verbundes. Hierbei handelt es sich um fünf Felder vom Typ »Feld«. Somit können in den fünf Feldern (man spricht auch hier von Feldern, allerdings ist das nicht mit dem Datentyp »Feld« zu verwechseln) Strings verschiedener Länge abgespeichert werden.

Nach der Typendefinition wird nun eine Variable »AktDatensatz« deklariert, die vom Typ »Datensatz« ist. Wie können wir nun auf die Felder eines Verbundes zugreifen? Hierzu gibt es zwei Möglichkeiten. Die gängige Methode besteht darin, zuerst den Namen des Verbundes anzugeben, dann von einem Punkt getrennt den Namen des Feldes, dem ein Wert zugewiesen werden soll. Somit würde eine Zuweisung von »keine« an das Feld Bemerkung der Variablen »AktDatensatz« folgendermaßen aussehen:

```
AktDatensatz.Bemerkung := "keine";
```

Durch die Angabe des Variablennamens wird gewährleistet, daß die Zuweisung richtig erfolgt. Stellen Sie sich einmal vor, Sie hätten zwei Variablen namens »Datensatz1« und »Datensatz2«, beide vom Typ »Datensatz«. Würden Sie nur »Bemerkung := »keine«;« angeben, so wüßte der Compiler

nicht, ob im Feld »Bemerkung« der Variablen »Datensatz1« oder »Datensatz2« dieser Wert zuzuordnen ist. Durch die Angabe des Variablennamens ist das Ziel genau bestimmt.

Die zweite Möglichkeit, einer Variablen vom Typ »Verbund« Werte zuzuweisen, besteht durch die Anweisung »WITH«. Stellen Sie sich vor, Sie müssen dem gesamten Verbund »AktDatensatz« Werte zuweisen. Es wäre umständlich, immer wieder den Namen des Verbundes angeben zu müssen. Aus diesem Grund wurde der Befehl »WITH« implementiert. Seine Syntax lautet:

```
WITH Verbund DO
  Feld1 := Wert1;
  . .
  . .
  . .
  Feldn := Wertn;
END;
```

Hinter dem Befehl »WITH« folgt der Name des Verbundes, dessen Felder innerhalb der Konstruktion angesprochen werden sollen. Nach dem DO folgt die Zuweisung an die einzelnen Felder. Hierbei muß nun nicht mehr der Name des Verbundes angegeben werden, denn das wurde durch die WITH-Konstruktion erledigt. Somit erspart sich der Anwender einiges an Tipparbeit. Innerhalb der Konstruktion können beliebig vielen Feldern dieses Verbundes Werte zugewiesen werden. Der Compiler nimmt nun automatisch an, daß jedes Feld, welchem ein Wert zugewiesen wird, ein Feld des angegebenen Verbundes ist.

In Listing 7 können Sie die beiden Zuweisungsarten noch einmal in der Praxis sehen. Zuerst werden den Feldern des Verbundes »AktDatensatz« Werte zugewiesen. Dies geschieht innerhalb einer WITH-Konstruktion. So erspart man sich die Arbeit, bei jeder Zuweisung den Namen des Verbundes anzugeben. Anschließend werden die Inhalte der Felder wieder ausgegeben. Diesmal wurde dazu keine WITH-Konstruktion benutzt. Somit muß vor jedem Feld auch angegeben werden, zu welchen Verbund dieses Feld gehört. Sie sehen, daß hierzu etwas mehr Tipparbeit nötig ist, da immer wieder der Verbundname angegeben werden muß. Die WITH-Konstruktion kann einem gerade bei umfangrei-

chen Verbunden doch einiges an Tipparbeit abnehmen.

Eine Abwandlung des Verbundes ist der "variable Verbund". Mit ihm ist es möglich, die Verbunddefinition von einem Wert des Verbundes abhängig zu machen. Um einen variablen Verbund zu definieren, ist folgende Syntax nötig:

```
TYPE Name : RECORD
CASE Feld1 : Typ1 OF
Wert1 : Feld1.1 : Typ1.1;
      Feld1.3 : Typ1.3;
      ..... : .....
      Feld1.n : Typ1.n;
Wert2 : Feld2.1 : Typ2.1;
      Feld2.3 : Typ2.3;
      ..... : .....
      Feld2.n : Typ2.n;
.
.
Wertn : Feldn.1 : Typn.1;
      Feldn.3 : Typn.3;
      ..... : .....
      Feldn.n : Typn.n;
ELSE
Wert : Feldx.1 : Typx.1;
      Feldx.2 : Typx.2;
```

```
END;
END;
```

Zugegeben, diese Definition sieht sehr merkwürdig und kompliziert aus, ist es aber nicht. Da Sie die CASE-Anweisung ja schon kennen, dürfte der variable Verbund nicht so schwer zu verstehen sein. Innerhalb der CASE-Konstruktion wird abgefragt, ob das Feld vom angegebenen Typ einen bestimmten Wert annimmt. Ist dies der Fall, so wird der Verbund so definiert, wie es der entsprechende Teil, der zu dem angegebenen Wert gehört, bestimmt. Schauen wir uns dazu am besten ein Beispiel an. Listing 8 zeigt Ihnen, wie ein variabler Verbund definiert wird und wie Zuweisungen stattfinden.

Zuerst wird ein Datentyp namens »STRING«, anschließend ein variabler Verbund vom Namen »Anwender« definiert.

```
..... : .....
Feldx.n : Typx.n;
```

niert. Sein erstes Feld heißt »Zugang« und ist vom Typ »CARDINAL«. Je nachdem, welchen Wert die Variable »Zugang« annimmt, wird ein anderer Verbund definiert. Hat »Zugang« den Wert 1, so kann der Anwender auf das Feld »WichtigeMeldung« zugreifen, die anderen beiden sind gesperrt. Bekommt »Zugang« den Wert 2, so kann nur auf »NormaleMeldung« zugegriffen werden, während bei der Zuweisung des Wertes 3 an »Zugang« nur auf das Feld »OffizielleMeldung« zugegriffen werden kann. Nimmt »Zugang« keinen dieser drei Werte an, so kann auf kein Feld dieses Datentyps zugegriffen werden. Sie sehen, auch die Angabe eines ELSE-Statements ist möglich, ja sogar wünschenswert. Warum dies so ist, müßten Sie eigentlich noch vom letzten Kursteil wissen.

Nachdem eine Variable vom Typ »Anwender« deklariert

wurde, beginnt das Hauptprogramm. Hier wird der Anwender zuerst gebeten, seine Priorität einzugeben. Abhängig davon, welche Priorität er angibt, kann er nun eine wichtige, eine normale oder eine offizielle Meldung angeben. Gibt er keinen der drei Werte an, darf er keine Meldung machen.

Versuchen Sie, auf das Feld »NormaleMeldung« zuzugreifen, wenn der Anwender eine »1« eingegeben hat, so würde dies eine Fehlermeldung zur Folge haben. Somit kann also nur die für den Anwender bestimmte Meldung gesendet werden. Sie sehen, an Hand dieses kleinen Beispiels die Anwendungsmöglichkeit eines variablen Verbundes, hier dient er zum Schutz von Daten.

So, das war's für heute. Im nächsten Kursteil befassen wir uns mit Zeigern und Datenstrukturen.

(jb)

Listings

```
1 MODULE Unterbereich;
2
3 FROM InOut IMPORT WriteString, WriteLn, WriteCard;
4
5 TYPE Monat = [1..31];
6
7 VAR Monatstage : Monat;
8
9 BEGIN
10   FOR Monatstage:=1 TO 31 DO
11     WriteCard(Monatstage, 1);
12     WriteString(" Tag des Monats");
13     WriteLn;
14   END;
15   (*INC(Monatstage);*)
16 END Unterbereich.
```

ModKurs1.MOD

```
1 MODULE Aufzaehlung;
2
3 FROM InOut IMPORT WriteLn, WriteInt;
4
5 TYPE Textattr = (weiss, blau, rot, schwarz, unterstr,
6                 kursiv, fett, normal);
7
8 VAR Attr : Textattr;
9
10 BEGIN
11   Attr:=(weiss);
12   WriteInt(ORD(Attr), 6);
13   WriteLn;
14   Attr:=VAL(Textattr, 4); (* -> Attr:=unterstr *);
15   INC(Attr, 2); (* -> Attr:=fett *);
16   DEC(Attr); (* -> Attr:=kursiv *);
17   WriteInt(ORD(Attr), 6);
18   WriteLn;
19 END Aufzaehlung.
```

ModKurs2.MOD

```
1 MODULE Aufzaehlung2;
2
3 TYPE Typ = (a, b, c, d, e, f, g, h, i, j, k, l, m, n, o,
4            p, q, r, s, t, u, v, w, x, y, z, aa, bb, cc,
```

```
5           dd, ee, ff, gg, hh, ii, jj);
6   TypSet = SET OF Typ;
7
8   VAR Test : TypSet;
9
10  BEGIN
11    Test:=TypSet{a, jj};
12  END Aufzaehlung2.
```

ModKurs3.MOD

```
1: MODULE Mengen;
2:
3: FROM InOut IMPORT WriteString, WriteLn;
4:
5: TYPE AttrFlags = (normal, fett, kursiv, unterstrichen,
6:                  invers,
7:                  vblau, vweiss, vschwarz, vorange,
8:                  hblau, hweiss, hschwarz, horange,
9:                  v0, v1, v2, v3, h0, h1, h2, h3);
10:   AttrFlagSet = SET OF AttrFlags;
11:
12:
13: CONST normals = " [0m";
14:   fetts = " [1m";
15:   kursivs = " [3m";
16:   unterstrichens = " [4m";
17:   inverss = " [7m";
18:   v0s = " [30m";
19:   v1s = " [31m";
20:   v2s = " [32m";
21:   v3s = " [33m";
22:   h0s = " [40m";
23:   h1s = " [41m";
24:   h2s = " [42m";
25:   h3s = " [43m";
26:
27: VAR Attribute : AttrFlagSet;
28:
29: PROCEDURE SetzeAttr(Flags : AttrFlagSet);
30: BEGIN
31:   IF (normal IN Flags) THEN
32:     WriteString(normals);
33:   END;
34:   IF (fett IN Flags) THEN
35:     WriteString(fetts);
36:   END;
37:   IF (kursiv IN Flags) THEN
38:     WriteString(kursivs);
39:   END;
40:   IF (unterstrichen IN Flags) THEN
```



```

41:   WriteString(unterstrichens);
42: END;
43: IF (invers IN Flags) THEN
44:   WriteString(inverss);
45: END;
46: IF ((vblau IN Flags) OR (v0 IN Flags)) THEN
47:   WriteString(v0s);
48: END;
49: IF ((vweiss IN Flags) OR (v1 IN Flags)) THEN
50:   WriteString(v1s);
51: END;
52: IF ((vschwarz IN Flags) OR (v2 IN Flags)) THEN
53:   WriteString(v2s);
54: END;
55: IF ((vorange IN Flags) OR (v3 IN Flags)) THEN
56:   WriteString(v3s);
57: END;
58: IF ((hblau IN Flags) OR (h0 IN Flags)) THEN
59:   WriteString(h0s);
60: END;
61: IF ((hweiss IN Flags) OR (h1 IN Flags)) THEN
62:   WriteString(h1s);
63: END;
64: IF ((hschwarz IN Flags) OR (h2 IN Flags)) THEN
65:   WriteString(h2s);
66: END;
67: IF ((horange IN Flags) OR (h3 IN Flags)) THEN
68:   WriteString(h3s);
69: END;
70: END SetzeAttr;
71: BEGIN
72:   Attribute:=AttrFlagSet{kursiv, unterstrichen, vorang
e,
74:   hschwarz};
75:   SetzeAttr(Attribute);
76:   WriteString("Test");
77:   SetzeAttr(AttrFlagSet(normal));
78:   WriteLn;
79:   WriteString("Sollte normal sein");
80:   WriteLn;
81: END Mengen.

```

ModKurs4.MOD

```

1 MODULE Felder1;
2
3 FROM InOut IMPORT WriteString, WriteLn, WriteCard;
4
5 CONST Min = 0;
6       Max = 10;
7
8 TYPE Graph = ARRAY[Min..Max] OF CARDINAL;
9 VAR Graph1 : Graph;
10      i : CARDINAL;
11 BEGIN
12   FOR i:=Min TO Max DO
13     Graph1[i]:=i*i;
14     WriteCard(i,1);
15     WriteString(" 2 = ");
16     WriteCard(Graph1[i],1);
17     WriteLn;
18   END;
19 END Felder1.

```

ModKurs5.MOD

```

1: MODULE Felder2;
2:
3: FROM InOut IMPORT WriteLn, WriteInt, Write, WriteStrin
g;
4:
5: VAR TextStr : ARRAY[0..10] OF CHAR;
6:
7: PROCEDURE Textlaenge(Text : ARRAY OF CHAR);
8:
9: VAR i : INTEGER;
10:
11: BEGIN
12:   i:=0;
13:   Write(CHR(34));
14:   WHILE (i<=HIGH(Text)) AND (Text[i]#0C) DO
15:     Write(Text[i]);
16:     INC(i);
17:   END;
18:   Write(CHR(34));
19:   WriteString(" Länge : ");
20:   WriteInt(i, 1);
21:   WriteLn;
22: END Textlaenge;

```

```

23: BEGIN
24:   Textlaenge("Hallo");
25:   Textlaenge("Gefällt Ihnen der Kurs?");
26:   Textlaenge("");
27:   TextStr:="Hallo2";
28:   Textlaenge(TextStr);
29: END Felder2.

```

ModKurs6.MOD

```

1: MODULE Verbund1;
2:
3: FROM InOut IMPORT WriteString, WriteLn;
4:
5: TYPE NameString = ARRAY[0..40] OF CHAR;
6:   StrasseString = ARRAY[0..40] OF CHAR;
7:   WohnortString = ARRAY[0..30] OF CHAR;
8:   TelefonString = ARRAY[0..20] OF CHAR;
9:   BemerkungString = ARRAY[0..60] OF CHAR;
10:  Dateiname = ARRAY[0..40] OF CHAR;
11:
12:   Datensatz = RECORD
13:     Name : NameString;
14:     Strasse : StrasseString;
15:     Wohnort : WohnortString;
16:     Telefon : TelefonString;
17:     Bemerkung : BemerkungString;
18:   END;
19:
20: VAR AktDatensatz : Datensatz;
21:
22: BEGIN
23:   WITH AktDatensatz DO
24:     Name:="Dagobert Dick";
25:     Strasse:="Gaenseweg 0";
26:     Wohnort:="9999 Entenhausen";
27:     Telefon:="0815/007";
28:     Bemerkung:="Fluchente";
29:   END;
30:   WriteString(AktDatensatz.Name);
31:   WriteLn;
32:   WriteString(AktDatensatz.Strasse);
33:   WriteLn;
34:   WriteString(AktDatensatz.Wohnort);
35:   WriteLn;
36:   WriteString(AktDatensatz.Telefon);
37:   WriteLn;
38:   WriteString(AktDatensatz.Bemerkung);
39:   WriteLn;
40: END Verbund1.
41:

```



ModKurs7.MOD

```

1: MODULE Verbund2;
2:
3: FROM InOut IMPORT WriteString, ReadString, ReadCard,
WriteLn;
4:
5:
6: TYPE String = ARRAY[0..255] OF CHAR;
7:   Anwender = RECORD
8:     CASE Zugang : CARDINAL OF
9:       1 : WichtigeMeldung : String;
10:      2 : NormaleMeldung : String;
11:      3 : OffizielleMeldung : String;
12:     ELSE END;
13:   END;
14: VAR Anwender1 : Anwender;
15:
16: BEGIN
17:   WriteString("Geben Sie Ihre Priorität ein: ");
18:   ReadCard(Anwender1.Zugang);
19:   CASE Anwender1.Zugang OF
20:     1: WriteString("Sie dürfen eine wichtige Meldung
eingeben: ");
21:     ReadString(Anwender1.WichtigeMeldung);|
22:     2: WriteString("Geben Sie bitte eine Meldung ein:
");
23:     ReadString(Anwender1.NormaleMeldung);|
24:     3: WriteString("Sie dürfen nur eine offizielle Me
ldung
eingeben: ");
25:     ReadString(Anwender1.OffizielleMeldung);
26:   ELSE
27:     WriteString("Sie dürfen keine Meldung machen!");
28:   END;
29:   WriteLn;
30:   WriteLn;
31: END;
32: END Verbund2.

```

ModKurs8.MOD

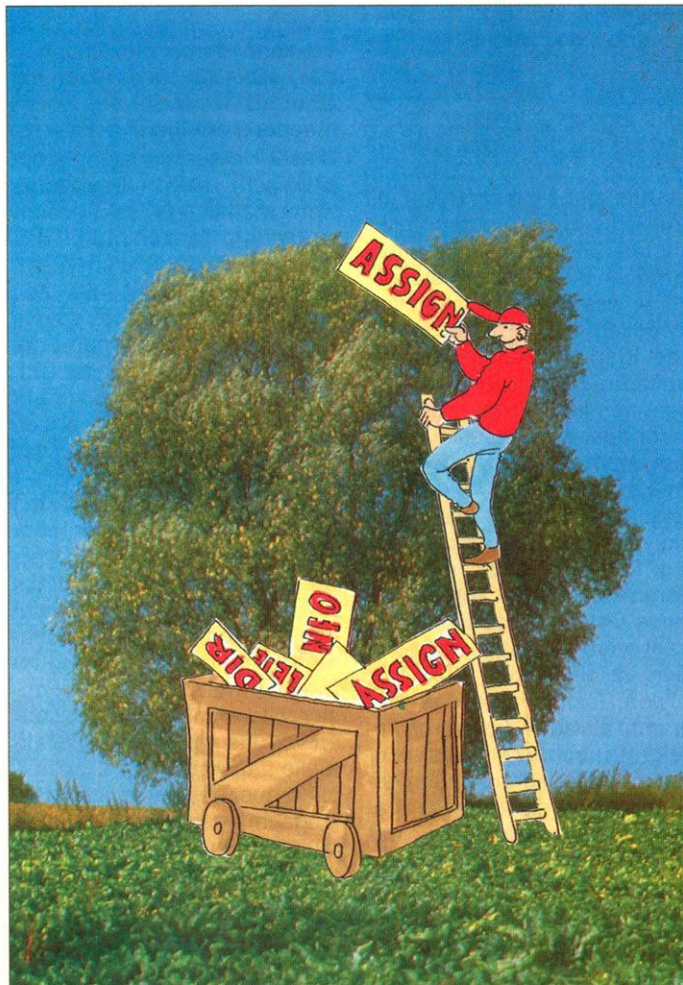
Michael Anton

Das AmigaDOS – leicht und verständlich!

In dieser Folge beschäftigen wir uns mit dem Abenteuer »Multitasking« und den Problemen, die sich bei der Verwendung dieser Fähigkeit in CLI oder Shell ergeben.

Kursfahrplan

- Teil 1: Einführung in das CLI
- Teil 2: Was kann Amiga DOS
- Teil 3: Stapeldateien
- Teil 4: Neu dabei bei 1.3
- Teil 5: Alternativen zum CLI
- Teil 6: Multitasking



etwas Multitasking in der Praxis, und zwar zunächst auf der Workbench. (Bis auf weiteres sollte eine Kopie der Originaldisk verwendet werden, damit sichergestellt ist, daß alle benötigten Programme vorhanden sind). Hangeln wir uns mit der Maus (welch Wohlgefühl?) in die Utility-Schublade und starten dort die Uhr, bei der wir die Anzeige der Sekunden einschalten. Als nächstes aktivieren wir das Notepad und schreiben einige Zeilen – die Uhr läuft weiter. Wenn noch ein Laufwerk frei ist, können wir auch noch ganz nebenbei eine Diskette formatieren. Bis zu 20 solcher Arbeiten können auf dem Amiga quasi gleichzeitig erledigt werden. Arbeiten kann man allerdings nur mit einer Anwendung, da man ja nur einen Satz an Eingabegeräten hat. Zwischen den einzelnen Aufgaben kann umgeschaltet werden, indem das Fenster der Anwendung angeklickt wird. Erwarten andere Tasks eine Eingabe, so warten sie, bis sie entsprechend bedient werden, andere Tasks wie die Uhr laufen kontinuierlich. Um an ihre Menüleisten zu kommen, muß ebenfalls das passende Fenster aktiviert werden. (Dies geschieht normalerweise durch einen einfachen Mausklick oder die Tastatur, es gibt jedoch auch Utilities wie »DMouse« aus der Public Domain, die die Fenster beim Bewegen der Maus automatisch aktivieren).

FensterIn mit System

Soweit die Workbench. Aber was sagt das CLI zum Thema Multitasking? Machen wir doch einfach die Probe aufs Exempel, schließen alle aktiven Anwendungen und begeben uns wieder in die Tiefen der Tippierei... Doch schon die erste Eingabe, nämlich »clock«, läßt die ersten Zweifel aufkommen. Zwar erscheint die gewünschte Uhr, sie ist sogar mit der Maus bedienbar, das CLI, mit dem sie aufgerufen wurde, ist jedoch tot. Zwar können im entsprechenden Fenster noch Eingaben gemacht werden, sie erreichen den Rechner jedoch nie und verlaufen im Sande. Ist das CLI also nicht Multitasking-fähig? Die Antwort lautet im besten »Radio Eriwan-Prinzip«: »Ja, aber...«. Der Grund?

Was ist eigentlich dieses vielgepriesene Multitasking? Es bezeichnet ganz einfach die Möglichkeit, daß ein Rechner mehrere (multi) Aufgaben (task(s)) sozusagen gleichzeitig ausführen kann. Der feine Unterschied liegt hierbei im »sozusagen«. Wenn der Rechner diese Aufgaben tatsächlich parallel abarbeiten sollte, so würde er eine Vielzahl von Rechenelementen benötigen. Solche Architekturen gibt es zwar inzwischen schon, sie sind jedoch recht teuer und daher nur selten zu finden. Stattdessen arbeiten die meisten Rechner nach dem Time-Sharing-Prinzip, mit dem die Rechenelemente optimal ausgenutzt werden. Im allgemeinen besitzt jeder Rechner ein solches Rechenelement, auch als CPU (Central Processing Unit) genannt. Diese CPU steuert so ziemlich alle Vorgänge innerhalb des Rechners – und langweilt sich dabei recht oft. Kein Wunder, denn in der Zeit, die ein Rechner auf das Drücken einer Taste durch den Anwender wartet, könnte er in einem Rechenprogramm schon zwei oder drei Wurzeln gezogen haben. Die Kunst des Multitaskings liegt nun darin, die verfügbare Rechenzeit (time) möglichst optimal auf die verschiedenen Aufgaben zu verteilen (share). Dank der hohen Arbeitsgeschwindigkeit sieht es für den Anwender dann aus, als würde der Rechner mehrere Aufgaben gleichzeitig erledigen, obwohl er sie nacheinander ausführt – nur eben sehr schnell in den Wechseln... Genau dies macht AmigaDOS – wer mit den Zahlenspiellereien in der Computerwelt vertraut ist, der weiß, daß mit der Grundausstattung eines Amiga (7, ungerade MHz und 512 oder 1024 KByte Speicher) beim besten Willen kein großer Blumentopf in punkto Multitasking zu gewinnen ist. Allerdings ist das Betriebssystem für dieses Gebiet beinahe maßgeschneidert, so daß sich doch mancher Startvorteil ergibt.

Multitasking in Praxis...

Bringen wir also unseren Amiga mal so richtig zum Schwitzen und gönnen wir uns nun

Ein Programm benötigt sowohl eine Möglichkeit zur Eingabe als auch zur Ausgabe von Daten. Das Thema Eingabe haben wir schon unter der Workbench kennengelernt, hier hilft der Klick mit der Maus. Aber die Ausgabe? Jedes Programm benötigt auch eine Möglichkeit zur Ausgabe von Informationen. Diese erfolgt über die bereits bekannten Fenster. Die sind normalerweise durch das Fenster, in dem das Programm aufgerufen wurde, gegeben, also das jeweilige CLI. Zwar eröffnen viele Programme ein neues Fenster zur Ausgabe von Informationen, denken jedoch nicht daran, das CLI wieder für andere Eingaben oder Programme freizugeben. So gibt es tatsächlich nur sehr wenig Programme, die über das sogenannte »auto-detach« verfügen und sich ohne Komplikationen vom CLI trennen. Die meisten Programme halten sich bei ihrem Aufruf eine todsichere Ausgabemöglichkeit frei, eben das CLI-Fenster, aus dem sie aufgerufen wurden – und dies ist danach ebenso tot für weitere Eingaben... Also muß man dem Programm beim Aufruf zumindest ein eigenes Fenster zuweisen, um danach problemlos weiterarbeiten zu können.

Freie Fahrt im CLI!!!

Aber wie kommt man dorthin? Der offensichtliche Weg ist der, daß man zunächst einige CLI-Fenster öffnet und dann zwischen den einzelnen Fenstern wechselt. (Wenn ich im folgenden vom CLI spreche, so gilt dies natürlich auch für das Shell!). Es geht, ist aber recht umständlich. Daher wurde im Amiga DOS der Befehl »run« eingeführt: Er macht nichts anderes, als ein neues CLI-Fenster zu eröffnen, in dem das aufgerufene Programm abläuft und das aktuelle CLI für weitere Eingaben frei hält. Mit

```
run clock
```

ist die erste Hürde in der Umsetzung unseres Beispiels der Workbench bereits genommen, der zweite Sprung lautet

```
run notepad
```

Danach können wir immer noch im aktuellen CLI arbeiten, beispielsweise eine Dis-

kette formatieren – mit weiteren Run-Befehlen können wir das CLI freihalten. Dabei ist es übrigens egal, ob es sich bei den zusätzlichen Tasks um echte Programme oder Scripts handelt, letztere können mit

```
run execute Anything.S
```

gestartet werden.

In besonders hartnäckigen Fällen kann es zusätzlich noch notwendig sein, eine Umleitung der Ausgabe mit »>NIL:« oder »>PRT:« vorzunehmen, hierzu muß ausprobiert werden, ob zum einen im entsprechenden CLI noch Eingaben angenommen werden, und ob es sich zum anderen noch mit »endcli« problemlos schließen läßt.

Den Überblick bewahren...

In den auf diese Weise gestarteten Tasks kann man jedoch sehr schnell den Überblick verlieren über das, was gerade abläuft. Eine gewisse Orientierungshilfe bietet noch der Parameter »%N« des Prompt-Befehls – er zeigt die Tasknummer des aktuellen Fensters an. Mit jeder per »run« gestarteten Task steigt diese Zahl jedoch, allerdings reflektiert sie nie die Anzahl der tatsächlich aktiven Tasks im Startfenster. Man kann dort mit »run« bis zu 19 andere Tasks starten, die angezeigte Tasknummer bleibt jedoch ständig »1« – was eigentlich auch verständlich ist... Für letztendliche Klarheit muß man einen weiteren Befehl zu Rate ziehen, nämlich »status«. Er zeigt mehr oder weniger ausführlich die momentanen Aktivitäten im System an. Ein einfaches »status« zeigt die Anzahl der aktiven Tasks und der jeweiligen Programme an, mit dem Parameter »TCB« können diverse Details zu den Tasks erfragt werden. Am ausführlichsten ist die Option »full«, die sowohl Tasknummer und Programm als auch die Interna anzeigt. Durch die Angabe einer Tasknummer können auch gezielte Anfragen zu einer bestimmten Task erfolgen. Ab der DOS-Version 1.3 ist mit der Option »com« oder »command« auch die Suche nach einer aktiven Task möglich. Nach

```
status com dir
```

wird überprüft, ob der Befehl »dir« in einer der Tasks aktiv ist. Wenn dem so ist, wird die Nummer der Task ausgegeben (die Umleitung kann in eine Datei erfolgen!), gleichzeitig wird der Erfolg der Suche in einem Returncode (0=gefunden) weitergeleitet, so daß bei Bedarf Scripts entsprechend reagieren können.

Aber wie wird man nun die Geister, die man rief, wieder los? Im Prinzip ganz einfach, indem man die vorhandenen Unterbrechungsmöglichkeiten ausnützt. Die haben wir schon zu Beginn des Kurses kennengelernt, es handelt sich um die Tasten [Ctrl-C.F], die das Programm oder das Script im aktuellen Fenster unterbrechen. Es gibt jedoch auch noch eine Art Fernsteuerung, die beliebige Tasks abbrechen kann, und zwar den Befehl »break«. Als Parameter wird dabei die Nummer der zu unterbrechenden Task übergeben, ohne weitere Parameter bewirkt dies, daß die Task den Abbruchbefehl [Ctrl-C] übermittelt bekommt. Mit dem zusätzlichen Parameter »all« wird erreicht, daß alle vier Meldungen an die Task übergeben werden. Ab DOS 1.3 kann durch die Optionen »C«, »D«, »E« und »F« auch eine gezielte Wahl der Unterbrechung erfolgen.

Dieser Notausstieg kann recht praktisch sein. Besonders wenn man einige Tasks gestartet hat und danach merkt, daß man eigentlich nicht alle Dienste benötigt oder die Zeit knapp wird. In eine solche Lage wollen wir uns nun bringen. Dazu öffnen wir vier CLI- oder Shell-Fenster, die wir auf praktikable Größen verkleinern und übersichtlich auf dem Bildschirm platzieren – es lebe die Maus! Nun starten wir in drei dieser Fenster möglichst lang dauernde Aufgaben wie beispielsweise »dir all«, die Anzeige einer längeren Datei mit »type opt h« und formatieren noch eine Diskette im Zweitlaufwerk. Das vierte Fenster wird aktiviert, dort schauen wir uns mit »status« die aktuelle Lage an. Zwar läuft nun alles, aber doch nur noch recht langsam – soviel zum Mythos Multitasking. Damit vor Feierabend wenigstens noch die Diskette formatiert ist, schalten wir die anderen Aufgaben einfach wieder ab, indem wir die Nummern der Tasks der Aus-

gabe von »status« entnehmen und ihnen ein freundliches »break all« zurufen. Die Ausgaben in den entsprechenden Fenstern werden angehalten, und das Formatieren geht nun wesentlich schneller vonstatten. Im vorliegenden Fall hätten wir zwar genauso gut die betreffenden Fenster mit der Maus aktivieren und dann eine Taste drücken können, aber nicht in jedem Fall liegen die Fenster derart günstig wie in diesem Beispiel...

Das Listing zeigt die Befehle »run«, »status« und »break« nochmals in der Praxis. Es werden einige CLI-Fenster geöffnet und in einem davon »dir« ausgeführt. Die Nummer dieser Task wird mit »status« abgefragt und die Task mit »break« abgebrochen.

Mit »break« lassen sich nur jene Tasks abbrechen, die vom CLI aus gestartet wurden. Tatsächlich laufen hinter den Kulissen noch viele andere Vorgänge ab, die sich dem Zugriff des Anwenders normalerweise entziehen. Das ist eigentlich auch recht gut so, denn diese Hintergrundtasks sind meist wichtige Systemfunktionen wie die Verwaltung von Tastatur, Bildschirm und Laufwerken. Hier kann man sich ohne entsprechende Erfahrung sehr schnell ins Abseits manövrieren. Es gibt jedoch auch Hilfsprogramme, die die Manipulationen solcher Systemvorgänge erlauben, für die Praxis ist vor allem das PD-Programm »XOPER« zu empfehlen – es ist aber mit Vorsicht zu genießen.

Einstellungssache

Was noch zu klären bleibt, ist die Bedeutung der übrigen Informationen bei der Ausgabe von »status«. Beim Aufruf mit den Optionen »full« oder »TCB« gibt »status« neben den bislang bekannten Informationen über Anzahl der Tasks und dem jeweiligen aktiven Befehl auch noch einige andere Werte zurück. Der Wert nach der Zeichenfolge »stk« gibt die Größe des verwendeten Stacks, »gv« die Größe der Global-Vectors-Tabelle und »pri« die Priorität der jeweiligen Task an. Bei dem gv-Wert handelt es sich um einen internen Wert, der an dieser Stelle nicht

weiter von Bedeutung ist. Die beiden anderen Werte sind es für die Arbeit in Shell oder CLI doch.

Der Stack ist ein bestimmter Speicherbereich, den ein Programm (oder eine Task) für interne Vorgänge belegen kann. Dessen Größe wird mit dem stk-Wert ausgegeben. Der tatsächliche Bedarf kann von Programm zu Programm variieren, bei der Arbeit mit der Workbench wird die jeweils benötigte Größe im zum Programm gehörenden File mit der Extension ».info« festgelegt. Dem ist bei der Arbeit in CLI oder Shell jedoch nicht so: Hier wird ein genereller Wert von 4000 Bytes als Vorgabe angenommen. Dies kann für einige Programme zuwenig sein. Sie reagieren dann mit einem Absturz zu gegebenener Zeit – ärgerlich, aber nicht unumgänglich. Mit dem Befehl »stack« kann die Größe des Stacks für jedes Programm individuell geregelt werden. Allerdings nur in gewissen Grenzen: Eine Änderung des Stacks gilt für das aktuelle CLI-Fenster und alle anschließend von dort aus gestarteten Tasks, nachträgliche Änderungen sind nicht möglich. (Allerdings Änderungen im voraus, und sogar direkt beim Öffnen des entsprechenden Fensters: Bei der Eröffnung eines CLI-Fensters mit »newshell« oder »newcli« kann statt der Standard-Startup-Sequenz auch eine andere Datei für die Vor-

NODE	DATA	CODE	PRI	T	S	TYPE	INTERRUPT NAME
002183d0	00218344	002205bc	0	H	D	Serial Out	serial.device
0020024c	002001fe	00fc4ae8	0	H	D	Disk Block	-----
002000f2	002000c8	00fc4694	0	H	D	Ports	ciab.resource
00275162	-----	00273932	-1	S	E	Blitter	Xop-1/0-counter
002005a0	0020055e	00fc6040	10	S	E	Vert.Blank	-----
002009f4	002008c4	00fc3824	0	H	D	Vert.Blank	gameport.device
00200af0	00200ac6	00fc935a	0	H	D	Vert.Blank	tuner.device
00200544	0020055e	00fc604a	0	H	D	Blitter	-----
00200c40	00200c16	00fc35e4	0	H	D	Audio Ch.0	audio.device
00200c44	00200c82	00fc35e4	0	H	D	Audio Ch.1	audio.device
00200d20	00200cee	00fc35e4	0	H	D	Audio Ch.2	audio.device
00200d8c	00200d5a	00fc35e4	0	H	D	Audio Ch.3	audio.device
002183ba	00218344	00221064	0	H	D	Serial In	serial.device
00200262	002001fe	00fc4000	0	H	D	Disk Sync	-----
0020018a	00200160	00fc4678	0	S	E	External	ciab.resource

ID	STATE	TYPE	MODE	NUM	PROCESSNAME	COMMAND
00251120	Waiting	Batch Bckg		3	ARP Background S MachII	-----
00269268	Waiting	Intct Frgr		2	New_MShell	-----

»Xoper« ist ein nützliches Utility

einstellungen des Fensters herangezogen werden, in dieser Datei könnte auch ein entsprechendes Stack-Kommando stehen. Diese Datei sollte im Verzeichnis »s:« stehen und muß mit »from« als Parameter angegeben werden, also beispielsweise »newshell ... from BigProg-Startup«). Ein Experimentieren mit »stack« ist immer dann angesagt, wenn Programme, die unter der Workbench problemlos laufen, in CLI oder Shell abstürzen.

Ein anderer interessanter Wert ist die Priorität einer Task. Er sagt nichts anderes aus, als wie wichtig eine Task ist und wieviel Rechenzeit ihr daher zur Verfügung gestellt wird. Der Wert für die Priorität einer Task kann zwischen -128 und

+127 liegen, je größer der Wert, desto häufiger wird die entsprechende Task bedient. Die Vorgabe für eine CLI- oder Shell-Task ist 0, dieser Wert kann mit »changetaskpri« jedoch geändert werden. Die Änderung wirkt normalerweise nur auf die aktuelle Task, ab DOS 1.3 kann mit der Option »process« aber auch der Wert für andere Task geändert werden, beispielsweise

```
changetaskpri -1 process 3
```

Allerdings sollten sich diese Änderungen nur in kleinen Bereichen, etwa ± 5 , bewegen, da es sonst zu Konflikten mit den internen Tasks des Betriebssystems kommen kann. (Deren Priorität kann beispielsweise auch mit »XOPER« geändert werden – ein recht ge-

fährliches Pflaster, da man hiermit problemlos Tastatur oder Laufwerke stilllegen kann!)

Die Änderungen sind im Normalbetrieb kaum zu spüren (experimentieren Sie mal etwas mit den letzten Beispielen); in manchen Fällen können sie jedoch auch Vorteile bieten. Wenn beispielsweise eine Datei sortiert oder ein Text ausgedruckt und parallel ein anderer Text editiert werden soll, so kann es praktisch sein, dem Editor zwecks schnellerem Tippens erhöhte Priorität einzuräumen und die Hintergrundaufgaben gemächlich dahindümpeln zu lassen.

Nun gut, da laufen jetzt mehrere Tasks nebeneinander vor sich hin, aber wie sieht es mit der Kommunikation zwischen diesen Tasks aus? Zumindest können sie sich gegenseitig unterbrechen, wie der Break-Befehl zeigt, doch mit der Übergabe von anderen Daten sieht es etwas trübe aus. Sicher, man kann eine temporäre Datei erstellen oder Umgebungsvariablen setzen, allerdings unter Verlust des Multitasking. Normalerweise muß nämlich gewartet werden, bis eine Task die Datei komplett erzeugt hat, bevor eine andere Task darauf zugreifen kann, gleichzeitiges Arbeiten geht also nicht. Zumindest nicht mit den konventionellen Geräten, da gibt es aber (ab 1.3) noch das Gerät namens »pipe:« – und mit dem geht es doch.

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

Der Name des Gerätes »pipe:« bezeichnet die Funktion recht gut: Eine Art Röhre, über die Informationen zwischen den Tasks ausgetauscht werden können. Im Gegensatz zu einer Datei kann auf die »Dateien« in »pipe:« von mehreren Tasks gleichzeitig zugegriffen werden. Während eine Task noch Daten schreibt, kann eine andere Task diese Daten gleich wieder übernehmen, ohne auf das Arbeitsende der schreibenden Task oder das Erstellen einer kompletten Datei warten zu müssen. »pipe:« kann auf zwei Arten betrieben werden: Entweder als eine Art Laufwerk, auf dem »Dateien« angelegt werden, oder als einfacher Kanal wie »nil:« oder »prt:«. Der Vorteil der ersten Methode ist der, daß über diese »Dateien« mehrere Übertragungen gleichzeitig ablaufen können. Als Beispiel sei angenommen, wir wollen den Inhalt eines Verzeichnisses als sortierte Liste darstellen. Der traditionelle Weg ist der, zuerst eine unsortierte Liste in eine Zwischendatei zu schreiben und diese danach zu sortieren. Bei der Verwendung von »pipe:« entfällt der Umweg jedoch und es kann mit Multitasking gearbeitet werden – während das Verzeichnis noch eingelesen wird, kann schon mit dem Sortieren begonnen werden:

```
run list Verzeichnis to
pipe:Inhalt
sort from pipe:Inhalt to
ram:Sortiert
```

(Natürlich könnte man »sort« auch noch im Hintergrund laufen lassen und noch eine dritte Aufgabe »nebenbei« erledigen...) Allerdings hat »pipe:« auch noch einige Einschränkungen: Als Datendurchreiche hat es nur eine begrenzte Kapazität von 4 KByte, wenn die voll sind, wird die schreibende Task solange blockiert, bis eine andere Task Daten ausliest. Weiterhin muß bei der Verwendung von Pipe-Dateien beachtet werden, daß nur einfache Dateinamen erlaubt sind, Unterverzeichnisse können in »pipe:« nicht angelegt werden.

Und zum Abschluß nun noch eine ganz besonders exklusive Art des Multitaskings: Der Multi-User-Betrieb. Bislang gehorchte der Amiga nur einem einzelnen Anwender, er kann aber noch einen mehr verkraf-

ten – sofern ein zweiter Rechner zur Verfügung steht. Was das für ein Rechner ist, tut nichts zur Sache, wichtig ist, daß er über eine serielle Schnittstelle und ein brauchbares Terminal-Programm (beispielsweise »Kermit«) verfügt. Weiterhin wird ein Null-

Modem-Kabel benötigt, mit dem die beiden Rechner verbunden werden, und das Netzwerk für den Hausgebrauch ist perfekt. Je nach den Gegebenheiten sind noch einige Vorarbeiten nötig, beispielsweise die Parameter der Schnittstellen in Einklang zu

bringen, dann kann der Spaß losgehen. Auf dem Zweitrechner wird das Terminal-Programm gestartet und die Verbindung hergestellt (bei »Kermit« zum Beispiel mit »connect«), auf dem Amiga wird ein Shell geöffnet. Allerdings kein normales, sondern eines, welches das Gerät »aux:« als Kommunikationskanal benutzt. (DOS 1.3 ist also nötig!) Der Befehl hierzu ist

```
newshell aux:
```

Dieses Fenster wird nun nicht mehr über die Amiga-Tastatur bedient, sondern über den Zweitrechner, dorthin gehen auch die Ausgaben. Das dortige Shell ist ein vollwertiges Shell, bis auf eine kleine Ausnahme: Die von dort gestarteten Tasks dürfen keine Ausgabe im Grafikmodus, sprich in einem eigenen Fenster, vornehmen. Wenn dem jedoch so sein sollte, wird der Amiga wieder aktiv. Es können jedoch alle DOS-Befehle verwendet werden, bei anderen Programmen muß experimentiert werden. Somit hat man für den Notfall eine recht praktische Art der Fernsteuerung, wenn man beispielsweise Rechenarbeiten wie das Wachsen von Mandelbrot-Bäumchen optisch überwachen und gleichzeitig noch soziale Plattenpflege betreiben möchte.

Abschied...

Hiermit sind wir am Ende unseres CLI-Kurses angelangt. Ich hoffe, Sie hatten etwas Vergnügen an diesem Kurs und auch einige neue Einsichten gewonnen. Im Idealfall sollte sich der Amiga nun mit einem völlig neuen Gesicht präsentieren und Ihr Interesse an einer – zugegebenerweise etwas rustikalen – 'neuen' Art der Computerei geweckt worden sein. Auch wenn nicht alles auf Anhieb geklappt hat, so verspricht diese Art der Bedienung doch eine maximale Ausnutzung der Möglichkeiten des Amiga. In diesem Sinne auch die Abschlufsaufgabe: Bleiben Sie am Ball und nur sehr selten an der Maus – die Tastatur mag zwar als Eingabemedium archaisch sein, sie erfüllt dennoch ihren Zweck und die nötige Fingerfertigkeit sollte ständig trainiert werden...

(vb)

Lösungen...

Hier die Lösungen zu den Aufgaben der letzten Folge – im Gegensatz zu anderen Fragen sind sie nicht "Zweihundvierzig...(Grüße an Arthur Dent).

Da wäre zunächst einmal die Abfrage von Umgebungsvariablen. Das Tierchen sollte etwas Komfort bieten und auch recht gut funktionieren. Unser Vorschlag:

```
1 .KEY Variable
2 IF <Variable> EQ ""
3 ECHO "n*e[33mVerwendung: AS-
KENV Variablenname*e[0m*n"
4 QUIT
5 ENDIF
6 ECHO "nBitte den Wert für die
Variable <Variable> eingeben"
7 ECHO "und die Eingabe mit
*e[33mCTRL+\ *e[0mbeenden.*n"
8 COPY * ENV:<Variable>
9 ECHO "n"
```

Und es geht tatsächlich... Es wurde schon angedeutet, daß der Weg zum Erfolg über den ominösen Befehl »copy« führt. Der Rest ist reine Kosmetik: Wird beim Aufruf kein Name für die Variable übergeben, so erfolgt in bester DOS-Tradition eine kleine Bedienungsanleitung. Der Rest ist offensichtlich...

Die zweite Aufgabe ist ebenfalls recht unproblematisch, wenn wir uns die Feinheiten innerhalb von Scripts noch einmal vor Augen halten. Der erste Block der Musterlösung definiert die erwarteten Parameter und stellt einige Feinheiten für die Übersichtlichkeit klar. Danach folgt ein Block, der die dringend benötigten Befehle resident macht, und danach die eigentliche Schleife. Dort werden über die Anweisung »\$« die beim Start nicht übergebenen Werte mit Vorgaben besetzt:

```
1 .KEY Anzahl,Meldung
2 .BRA {
3 .KET }
4 ; BEISPIEL1.S - Schleifenprogrammierung
```

```
5 ; mit Umgebungsvariablen und
EVAL
6 ; Erweiterte Fassung mit freier
Wahl
7 ; oder fester Vorgabe
8 RESIDENT C:EVAL PURE
9 RESIDENT C:COPY PURE
10 RESIDENT C:ECHO PURE
11 RESIDENT C:IF PURE
12 RESIDENT C:SKIP PURE
13 ECHO "ec"
14 SETENV Count 0
15 LAB Loop
16 EVAL <ENV:Count >NIL: TO
ENV:Temp VALUE2=1 OP=+ ?
17 COPY ENV:Temp ENV:Count
18 ECHO "[Meldung$Mehr!]"
19 IF NOT $Count EQ {Anzahl$2}
VAL
20 SKIP Loop BACK
21 ENDIF
22 ECHO "Das war alles..."
23 RESIDENT SKIP REMOVE
24 RESIDENT IF REMOVE
25 RESIDENT ECHO REMOVE
26 RESIDENT COPY REMOVE
27 RESIDENT EVAL REMOVE
```

Danach werden die residenten Befehle wieder entfernt, und der Amiga ist sauber wie zuvor... Bei diesen Scripts sollte das Flag »S« gesetzt sein, um eine korrekte Abarbeitung zu ermöglichen. Beim erweiterten Beispiel ist noch etwas Fingerspitzengefühl bei der Übergabe der Parameter angesagt. Die Reihenfolge der Parameter ist »Anzahl« und »Meldung«, also »Wie oft« »Was«. Die Vorgaben ohne Angabe von Parametern sind zweimal »Mehr!«, der erste übergebene Wert bezeichnet die Anzahl der Durchläufe. Übergibt man dem Script als ersten Parameter die Meldung anstatt der Zahl der Durchläufe, so ist das Ergebnis nicht unbedingt das erwünschte. Dem ist ebenso, wenn die Meldung Leerzeichen enthält: Hier ist die Verwendung von Anführungszeichen nötig, ansonsten kommt es zu Fehlern. Die Übergabe der Parameter könnte mit der Option »/K« im Befehl »KEY« geklärt werden.

Assembler ist keine Hexerei

In diesem Heft nun beenden wir Teil 1 mit den letzten wichtigen Erklärungen zum Aufbau des Startup-Codes.

Mit Schwung schließen wir in diesem Teil den Entwurf des Startup-Codes ab und erarbeiten die Entwicklungsziele für den AmigaDOS-File-Requester (ADR). Das Projekt gewinnt an Gestalt; die vorgeschlagene Spezifikation mag "Respekt" einflößen, weil der Realisierungsweg lang und steinig erscheint. Schrittweise, wie beim Startup-Code, wird es uns gelingen, die sich auftürmenden Hindernisse zu überwinden.

CLI-Argumente auswerten

Es gilt, aus der Kommandozeile die Programmargumente herauszufiltern, um sie unseren Anwendungen zur Verfügung zu stellen. Auch für diese Aufgabenstellung läßt sich ein Ablaufplan als Basis für die Codierung erstellen. Vorher ist jedoch abzuklären,

- (1) wie wir an die Argumente herankommen,
- (2) welche Informationen dort anfallen können,
- (3) welche Konventionen zu beachten sind und
- (4) wie die aufbereiteten Argumente der Anwendung, das heißt, dem eigentlichen Programm, anzubieten sind.

Die Kommandozeile haben wir bereits am Anfang unseres Startup-Codes in den Griff genommen, als wir die Inhalte der Register »a0« und »d0« sicherten. Das Adreßregister weist auf den Anfang der Zeichenkette (»cmdBuf«), die als Kommandozeile eingegeben wurde. Das Datenregister hingegen enthält die Länge der Zeichenkette (»cmdLen«), das heißt die Anzahl der gültigen

Zeichen. Zweifel an diesen Angaben ließen sich leicht mit Unterstützung des Debuggers ausräumen. Führen Sie dazu den Startup-Code bis zum Label »OpenDos« aus und inspizieren Sie den Arbeitsspeicher (Option »mb«) ab der Adresse, die »a2« enthält. Versäumen Sie es nicht, vor dem Start des Programmes eine Kommandozeile (Option »c«) zu übergeben und die Register zu initialisieren (Option »i«). Gelegentlich muckt der Debugger dabei auf und lockt Gurus an. Schade!

Als nächstes gilt es herauszubekommen, welche Informationen die Kommandozeile enthalten kann. "Die übergebenen Argumente", werden Sie sagen, "so wie sie eingetippt wurden!" Schon recht, sehen wir uns aber ruhig Abbildung 3.1 dazu an, welche die prinzipiell möglichen Fälle darstellt. Es ist zu erkennen, daß

- einleitende und abschließende Leerzeichen vorhanden sein können,
- die Zeile durch "0A" (Line Feed) abgeschlossen wird,
- Argumente durch ein oder mehrere Leerzeichen getrennt sind und
- Argumente, die ein Leerzeichen enthalten, in Anführungszeichen stehen.

Wie im täglichen Leben kann man auch bei der Auswertung der Kommandozeile nichts wirklich falsch machen, wenn man sich an die etablierten Regeln hält. Auf unseren konkreten Fall bezogen erfordern sie, die Argumente aus der Kommandozeile als einzelne, mit »00« abgeschlossene Zeichenketten herauszufiltern. Anführungszeichen sind in die neuen Zeichenketten nicht zu

übernehmen; sie dienen lediglich dazu, die Zusammengehörigkeit von Zeichen (auf der Kommandozeile) zu markieren. Die Konventionen verlangen jedoch auch, daß der Name des Programms (»cli_CommandName«) als erstes Argument aufzufassen ist.

Konventionen geben Sicherheit

Die Aufgabenstellung ist klar; wie aber stellt man die Argumente der jeweiligen Anwendung zur Verfügung? Auch dafür besteht eine Konvention. Die Anzahl der Argumente und die Adresse eines Arrays, das die Adressen der Argumente im Arbeitsspeicher enthält, sind auf dem Stack (Langworte) als »argc« (argument count) und als »argv« ("argument vectors") zu übergeben. Diesen Vorgang haben wir bereits in unserem Startup-Code simuliert ([1], Listing 1-4, »ArgumenteÜbergeben«). Das Array haben wir ebenfalls als dynamische Struktur angelegt ([1], Listing 1-4, »SUArgsAnlegen«). Gleichzeitig haben wir in der Struktur »suArgs« ("StartupArgumente") auch ein sicheres Plätzchen für die Argumente geschaffen (»su_argvBuf«). Wir dürfen den Speicherplatz für die Kommandozeile dazu nicht verwenden, da jedes CLI nur über eine einzige verfügt, von der mehrere Programme (zum Beispiel mit »run«) gestartet werden können. Multitasking bereitet somit nicht nur Annehmlichkeiten. Wir wissen, daß der Name des Programms als erstes Argument zu verarbeiten ist. Der erste Gedanke rät, einfach die Adresse des »cli_CommandName« in der Struktur »suArgs« zu vermerken. In den meisten Fällen wäre das sicherlich auch ausreichend. Was aber, wenn unsere An-

Wichtig für alle Kursteilnehmer!

Unser Kurs "Assembler ist keine Hexerei" gliedert sich in insgesamt fünf (5!) Teile, so weit, so gut; daß Sie nach drei AMIGA-DOS-Heften immer noch Teil eins vor Augen haben, liegt nicht daran, daß Sie in der falschen Ausgabe gelandet sind, sondern daran, daß die Kursteile extrem lang sind und sich in mehrere Unterkapitel aufteilen. Wir wollen Ihnen mit dieser Aufspaltung nur helfen:

- a) jeden Schritt des Kurses nachvollziehen zu können und
- b) eine Überforderung angesichts des Themas zu vermeiden.

wendung ihren Namen sucht, um zum Beispiel einen so benannten Message-Port zu schließen, zwischenzeitlich aber weitere Programme vom CLI aus gestartet wurden? Ein Desaster könnte eintreten. Damit ist klar, daß der »CommandName« in den Puffer von »suArgs« zu kopieren ist. Einen Weg, wie dabei vorgegangen werden kann, beschreibt Abbildung 3.2.

Nun sind Sie an der Reihe. Den Code liefern wir in der nächsten Ausgabe nach. Beachten Sie bitte, daß die CLI-Struktur und der »CommandName« in der jeweils übergeordneten Struktur mit durch vier geteilten Adressen (BPTR) vermerkt sind (Adresse in Langworten, nicht in Bytes, wie gewohnt). Die eigentliche Adresse erhalten Sie durch »2 * SHIFT left« oder durch zwei Additionen.

In den Strukturen »Process« und »CommandLineInterface« ([1], Listing 1-4) finden Sie weitere Adressangaben in Lang-

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. Beispiel: Eine Kommandozeile mit drei Argumenten

worten, die mit »bptr« gekennzeichnet sind. Die Unterroutine »GetCmdNameArg« rufen Sie bitte aus »CLISart« auf. Der Debugger kann Ihnen helfen zu prüfen, ob sich das gewünschte Ergebnis eingestellt hat.

Nun gilt es, die Argumente aus der Kommandozeile (Beispiel Tabelle 3.1) herauszufiltern. Für das Vorgehen gibt es mindestens zwei Wege:

- die uns bekannte Anzahl an Zeichen auf der Kommandozeile (»cmdLen« von »d0« nach »d2« gerettet) durcharbeiten oder

- das Ende der Kommandozeile mit »00« terminieren (»0A« ersetzen) und solange zeichenweise vorgehen, bis das Ende erkannt wird.

Wir entschließen uns für den zweiten Weg, um das Führen eines Zählers zu vermeiden; außerdem läßt es sich sehr effizient auf Null testen. Tabelle 3.3 skizziert einen möglichen Weg, die Argumente aus der Kommandozeile zu extrahieren und zu kopieren. Der Code folgt in der nächsten Ausgabe. Bei unserer Lösung setzen wir voraus, daß keine Steuerzeichen als Argumente (in Anführungszeichen) auftreten. Bei Bedarf ist die entsprechende Unterroutine zu erweitern.

Zum Test Ihres Codes könnten Sie eine Routine entwickeln, die die Argumente ausliest und auf dem Bildschirm ausgibt. Ihre Vorstellungen von der Art, wie Strukturen im Arbeitsspeicher abgebildet werden, könnten Sie jedoch viel besser durch Verwendung des Debuggers fördern. Inspizieren Sie einfach den Speicher mit

der Option »mb« von der Anfangsadresse der Struktur »suArgs« ab. Sie werden am Anfang die Adressen der einzelnen Argumente und danach diese selbst erkennen. So wird es Ihnen möglich zu prüfen, ob die Argumente richtig übertragen wurden, und ob die Anfangsadressen in »argvArg« stimmen. Falls nicht, so können Sie hoffentlich Hinweise auf Fehlerursachen gewinnen.

Workbench-Argument umsetzen

Der Start eines Programms von der Workbench kann gleichfalls mit Argumenten erfolgen. Dazu sind die Icons anzuklicken.

Die Argumente werden in Form einer Liste angeboten, auf die die Startup-Message verweist (siehe Tabelle 3.4). Die verwendeten Strukturen finden Sie in Listing 3.1, das gleichzeitig den Grundstock für eine neue Include-Datei bildet.

Um den Startup-Code einfach und universell zu halten, verzichten wir darauf, ein Window anzulegen. Folglich können wir der Prozeßstruktur (Jedes von der Workbench gestartete Programm ist ein neuer, exklusiver Prozeß) auch keinen brauchbaren »FileHandle« entnehmen, um zum Beispiel mit DOS-Routinen Text auszugeben. Zu diesem Vorgehen entschlossen wir uns, da Anwendungen zumeist maßgeschneiderte Windows erfordern. Ein routinemäßig geöffnetes Fenster würde nur stö-

Tabelle 3-1. Systemverhalten

- A 01 System möglichst gering belasten; ADR nur einmal aktiv
- A 02 ADR auf Workbench öffnen; dessen Farben nutzen

Benutzeroberfläche

- A 03 Bedienung per Maus, Texteingaben mittels Tastatur
- A 04 Benutzer bei der Bedienung führen
- A 05 Nutzerführung über Programmschnittstelle bestimmen
- A 06 Umfang darzustellender Verzeichniseinträge wahlweise reduzieren

Internes Verhalten

- A 07 logische und physikalische Devices automatisch erkennen und darstellen (bis zu zehn physikalische)
- A 08 keine Begrenzung für Anzahl der Devices
- A 09 bis zu 15 Verzeichniseinträge/Devices gleichzeitig darstellen
- A 10 Keine Begrenzung für Anzahl der Devices beziehungsweise Einträge darin
- A 11 Wartezeiten bei Analyse von Verzeichnissen minimieren

Programmschnittstelle

- A 12 sprachunabhängig und einfach (mit Ausbaumöglichkeit für »AREXX«)

ren. Die Auswertung der Workbench-Argumente ist recht einfach, wie Sie der Tabelle 3.5 entnehmen können. Wir setzen lediglich das aktuelle Verzeichnis auf das als »Lock« übergebene Verzeichnis (erstes Argument). Für die Codierung finden Sie die be-

nötigten Informationen in Listing 3.1. Wissen Sie schon, wann wir unseren Code bringen?

Beachten Sie, daß die Locks in der Liste der Argumente von der Workbench (genauer dem Programm, das die Workbench betreibt) angelegt wur-

Tabelle 3-3. Entwicklungsziele ADR-Benutzeroberfläche

- N 01 Aussehen, Maße und Funktionen des ADR
- N 02 ADR-Window (beim Öffnen) in der Horizontalen mittig über dem Mauszeiger positionieren
- N 03 ADR-Window verkleinern, wenn 10 Sekunden lang unbe-nutzt oder Eingabe beendet
- N 04 Window durch Anklicken oder per Aufruf über die Schnittstelle auf volle Größe entfaltbar
- N 05 Farben im ADR-Window bei Standard Farbeinstellung:
 - Hintergrund : blau
 - statische Texte : weiß
 - Anzeigen:
 - Verzeichnisse : rot
 - Devices : rot
 - Dateien : weiß
 - "info-Dateien" : schwarz
 - selektiert : invertiert
- N 06 Im ADR 15 Verzeichnisse/Devices in einem mit Rollbal-len zu bedienenden Fenster darstellen; Namen bei Be-darf für die Anzeige kürzen
- N 07 Eingaben (Datei- Verzeichniswahl) abschließen mit:
 - Doppelklick
 - "Weiter"-Gadget
 - RETURN
- N 08 Name des Programmes im ADR darstellen, für das gera-de gearbeitet wird oder "direkt", falls nicht über Schnit-tstelle aktiviert
- N 09 Gadgets mit den zu wählenden Funktionen beschriften und nicht mit den aktiven
- N 10 Benutzerführung durch Sperren (Verschleiern) nicht benutzbarer Gadgets

Tabelle 3-2. Entwicklungsziele "Systemverhalten"

- S 01 Programm startfähig von CLI und Workbench
- S 02 Code residentfähig im Sinne des AmigaDOS
- S 03 Ein Programmargument als Verzeichnis interpretieren und dessen Inhalt darstellen
- S 04 Nicht mehr benötigten Arbeitsspeicher umgehend frei-geben
- S 05 ADR alternativ nutzbar durch Anklicken des Windows be-ziehungsweise nach Aufforderung durch ein Programm
- S 06 Aktives Programm (geladen und gestartet) identifiziert sich durch einen Message-Port
- S 07 Der gestartete ADR terminiert sich selbst, wenn sich ein Message-Port (S 06) bereits im System befindet
- S 08 Workbench-Screen mit 640 * 256 annehmen
- S 09 Screen nach vorne bringen und 10 Zeilen nach unten zie-hen, falls bei Aufruf des ADR durch andere Screens ver-deckt. Screen gegebenenfalls wieder in den Hintergrund bringen, wenn ADR beendet
- S 10 Multitasking-gerechte Implementierung

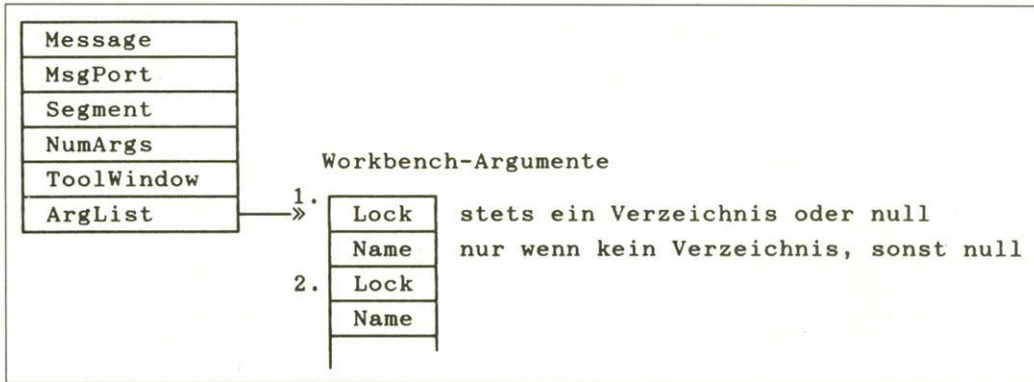


Abb.2. Die Workbench-Startup-Message

den. Es ist somit Aufgabe der Workbench, die Locks auch wieder freizugeben. Das bedeutet, daß die Liste der Argumente durch eigene Anwendungen nicht verändert werden darf.

Ran an den ADR

Das Thema "Startup-Code" ist nun so gut wie abgeschlossen. Den Code werden wir bei der Realisierung unseres Projektes "AmigaDOS-File-Requester" (ADR) einsetzen. Bis es dazu kommt, haben wir noch die Analyse- und die Entwurfsphase zu durchlaufen. Heute wollen wir so weit kommen, daß erste Teile des Entwurfes entstehen, die Sie bis zur nächsten Ausgabe in Code umsetzen können. Uns kommt es darauf an, zu vermitteln, daß einzelne Teile eines Projektes sich durchaus in unterschiedlichen Entwicklungssta-

dien befinden können, ohne daß daraus zusätzlicher Aufwand entsteht. Der Systembeziehungswise Programm-entwurf hält die "losen Enden" zusammen und sorgt dafür, daß sie zusammenpassen. Die Technik der "schrittweisen Verfeinerung", wie bereits bei der Entwicklung des Startup-Codes eingesetzt, erhält uns die Flexibilität. Das Ziel des heutigen Beitrages wäre erreicht, wenn Sie ansatzweise akzeptierten, daß methodisches Vorgehen nicht nur einen "overhead" verursacht, sondern die Realisierung wirksam unterstützt.

Spitze des Eisberges

Treten wir nun mutig in die Analysephase ein, um die Anforderungen aufzustellen, die unsere Arbeit bestimmen wird. Was

halten Sie davon, wenn wir einen File-Requester entwickeln, der

- als eigenständiges Programm einsetzbar ist, aber auch
- resident im Arbeitsspeicher als eigener Prozeß gehalten werden kann, um über eine einfache Schnittstelle von beliebigen Programmen genutzt zu werden?

Mit dieser "globalen" Forderung ist nur die äußerste Spitze des Eisberges beschrieben. Man könnte auch etwas freundlicher gestimmt von der Spitze einer Pyramide sprechen, die aus Schichten besteht, die jeweils einer Detaillierungsebene (Abstraktionsebene) entsprechen. Von der Spitze haben wir uns zur Basis vorzuarbeiten, zum eigentlichen Programm, der Realisierung der Anforderung. Der Weg bis dahin ist noch weit, die Details sind uns noch verborgen (also doch Eisberg).

Über der Wasserlinie

Wären die beiden allgemeinen Anforderungen an den ADR nicht von uns selbst erhoben, so ließen sie noch sehr viel

Spielraum, um Entwicklungsziele abzuleiten. Steigen wir nun eine Stufe von der Spitze herab, um die Forderungen zu präzisieren. Könnten Sie sich dem Forderungskatalog gemäß der Tabelle 3.1 anschließen? Eigene Ideen könnten Sie ja hinzufügen. Nehmen Sie sich ruhig viel vor. Man muß nicht alles auf einmal verwirklichen. Spätere Erweiterungen fallen um so leichter, je besser sie im Bauplan bereits berücksichtigt sind. Hoffentlich haben Sie sich bei der Sammlung Ihrer Ideen genauso wenig von den Details der Realisierung beeindrucken lassen wie wir. In dieser Phase geht es nur um Anforderungen (Sie verstehen schon den Vergleich mit dem Eisberg!?).

Aus den Anforderungen sind die Entwicklungsziele abzuleiten, deren Aufstellung auch als Pflichten- oder als Lastenheft bezeichnet wird. Anforderungen und Entwicklungsziele lassen sich mitunter schwer abgrenzen. Die Übergänge sind fließend. Auf unser Projekt bezogen könnte man durchaus der Meinung sein, daß das Aussehen des ADR (Darstellung auf dem Bildschirm) eine Anforderung ist. Auf die Abgrenzung kommt es aber gar nicht an. Viel wichtiger ist es, die Analysenakte mit einer eindeutigen Spezifikation abzuschließen, die die Entwicklungsziele bestimmt. In der Spezifikation konkretisieren sich die Anforderungen. Tja, wenn das so ist, dann müssen wir ja nur die Tabelle der Anforderungen durchgehen und sie in Entwicklungsziele umsetzen. Anders als bei der Aufstellung der Anforderungen gewinnen nun Details an Bedeutung. Vom Leistungsvermögen der Hardware und des Betriebssystems dürfen

Tabelle 3-4. Entwicklungsziele ADR-Internes Verhalten

I 01	Devices nur einmal nach Programmstart ermitteln
I 02	Device-, Verzeichnis- und Datei-Namen in dynamisch verwalteten Listen führen; diese alphabetisch sortieren und anzuzeigende Objekte kennzeichnen
I 03	Listen merken, bis Verzeichnis gewechselt wird oder schreibend darauf zugegriffen wurde; alternativ erneutes Auswerten von Verzeichnissen unterbinden (Mode: statisch), um vor allem den Diskettenbetrieb zu beschleunigen.
I 04	Betriebsarten: - FRE : Verzeichnisse und Dateien wählbar - VER : nur Verzeichnisse auswählbar - DAT : nur Dateien auswählbar
I 05:	Darstellungsarten: - mit/ohne Patternspezifikation - mit/ohne File Extension - statisch: Verzeichnisinhalte beibehalten bis zum Wechsel - dynamisch: Verzeichnis nur neu einlesen, wenn darauf schreibend zugegriffen wurde. - mit/ohne ".info"-Dateien

Tabelle 3-5. Entwicklungsziele ADR-Programmschnittstelle

P 01	Exec-Message Passing System verwenden
P 02	Message Port mit "MailBox-ADR" bezeichnen
P 03	Nachrichten bestehen aus Zeichenketten
P 04	Nachrichten steuern Betriebsverhalten und Anzeige des ADR
P 05	Nachrichten enthalten den gewünschten Zustand (Prog -> ADR) und den eingenommenen (ADR -> Prog).
P 06	Schnittstelle wird nur bedient, wenn ADR nicht im direkten Modus arbeitet (ADR-Window dann klein)


```

1 * Workbench.i
2
3 * WbMessage
4
5 struktur
6   strukt sm Message,20
7   aptr   sm_Process
8   long   sm_NumArgs
9   aptr   sm_ToolWindow ; auf Zeichenkette
10  aptr   sm_ArgList    ; auf Liste mit WBargs
11  struktLaenge sm_size
12
13
14 * WBArg
15
16 struktur
17  bptr   wa_Lock
18  aptr   wa_Name      ; auf Zeichenkette
19
20
21 * Ergänzung zu incl/Dos.i
22
23 CurrentDir = -126 (lock)(d1)

```

Die Include-Datei "Workbench.i" enthaelt die Strukturen, um Workbench-Argumente auszuwerten. In "Dos.i" ist noch das "CurrentDir" nachzutragen.

Abb.3. Die zum Kurs gehörende Include-Datei

wir nun nicht länger abstrahieren. Die Tabellen 3.2 bis 3.5 enthalten das Ergebnis unserer Bemühungen, die Anforderungen in Entwicklungsziele umzusetzen. Die Darstellung des ADR auf dem Bildschirm ist ein Entwicklungsziel. Beachten Sie bitte, daß mehrere Entwicklungsziele auf das äußere Erscheinungsbild des ADR durchschlagen. Versehen Sie Ihren Entwurf mit Maßen (Abstände, Koordinaten in Pixel) und überlegen Sie auch, wo und für welchen Zweck Gadgets zu platzieren sind. Selbst wenn Sie über ein Tool wie den »R.C.T.« [2] verfügen, sind eine Reihe von Vorüberlegungen erforderlich, die sich auch auf die zu verwendenden Datenstrukturen auswirken.

(jb)

Literatur und Software:

- [1] AMIGA DOS 9/90
- [2] Requester/Menu Construction Tool, MAXON

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. Wenn 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

AMIGA 90
 Wir stellen aus
 09.-11. November 1990
 Halle 12, Stand 307

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

88,- DM 145 ÖS 18 sfr
 DMV Verlag

EDITION

TOOLBOX

· Amiga · Atari · 68000 · Macintosh ·

AMIGA:

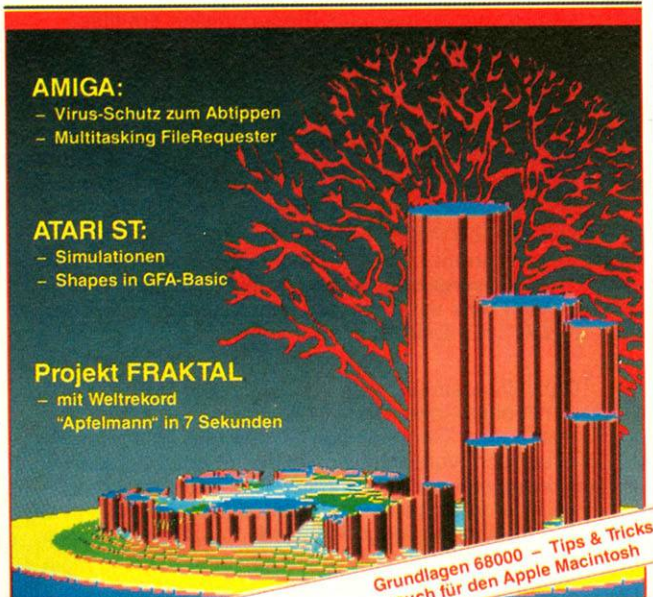
- Virus-Schutz zum Abtippen
- Multitasking FileRequester

ATARI ST:

- Simulationen
- Shapes in GFA-Basic

Projekt FRAKTAL

- mit Weltrekord
- "Apfelmännchen" in 7 Sekunden



Grundlagen 68000 – Tips & Tricks
 auch für den Apple Macintosh

Die glorreichen Drei

Animation · Video · Grafik

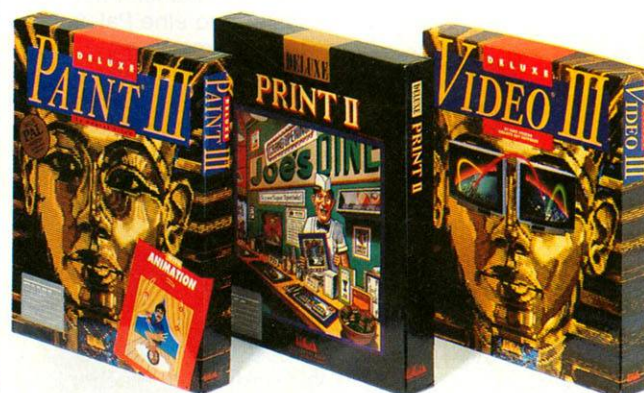


Create Interactive Multimedia presentations. . .

- ☐ Business presentations
- ☐ Entertainment
- ☐ Education
- ☐ Training



Click Buttons to Change Screen



Jetzt kommt Leben ins Bild. Mit der Multimedia-Software, die's ganz schön bunt mit Ihrem Amiga treibt – und ganz neue Perspektiven eröffnet, weil die Programme sich untereinander so gut verstehen.

Deluxe Paint III

Computergrafik mit Animationsfunktion in trickfilmartiger Darstellung. Mit wenigen Anweisungen geraten Ihre Bilder in Bewegung und verwandeln Ihren Amiga in ein Trickfilm-Studio.

Systemanforderungen:
Amiga 500, 1000, 2000 mit mindestens 1 Mbyte Arbeitsspeicher.
Bestell-Nr. 54138

DM 249,-*

Update von Deluxe Paint II auf Deluxe Paint III.

Bestell-Nr. 54138U

DM 99,-*

Deluxe Video III

Ihre Video-Produktion am Bildschirm. Es verarbeitet die Bilder aus Deluxe Paint III. Mit Umwandlung über Timing-Kontrolle

und Anim-Dateien. Die vier Einzelprogramme verknüpfen Musik, Ton, Bilder und Animation zu tollen Multimedia-Shows, die wieder auf Disketten oder Videokassetten überspielt werden können.

Systemanforderungen:
Amiga 500, 1000, 2000, 2500 mit mindestens 1 Mbyte RAM, 2 Diskettenlaufwerke (Festplatte empfohlen).
Bestell-Nr. 52586

DM 299,-*

Update von Deluxe Video 1.2 auf Deluxe Video III.

Bestell-Nr. 52586U

DM 149,-*

Deluxe Print II

Gestalten Sie Visitenkarten, Briefköpfe, Einladungen oder Poster nach professioneller Art: ein Programm für kreatives Design von Texten, Signets, Logos. Mit einer Palette leuchtender Farben.

Bestell-Nr. 52582

DM 199,-*

Updates erhalten Sie gegen Einsendung Ihrer Originaldiskette und eines Verrechnungsschecks an Markt & Technik Verlag AG, Kundenbetreuung, Update-Service, Hans-Pinsel-Straße 2, 8013 Haar

Markt & Technik-Bücher und -Software erhalten Sie bei Ihrem Buchhändler, in Computerfachgeschäften und in den Fachabteilungen der Warenhäuser.



Markt & Technik
Zeitschriften · Bücher
Software · Schulung

* Unverbindliche Preisempfehlung

Mittlerweile gibt es eine große Anzahl an Antivirenprogrammen. Jedoch schleichen sich immer wieder neue tückische Viren in den Rechner und infizieren Ihre Disketten. Auf der Time Special Nr.2 befindet sich ein weiteres Programm zur Bekämpfung der Viren.

VTest 1.67 – den Viren auf der Spur

»VTest 1.67« kann sowohl aus der Workbench als auch aus dem CLI gestartet werden. Falls Sie sich im Besitz einer Harddisk befinden, empfiehlt es sich, das Programm in die Startup-Sequence einzubinden, zumal das Programm Ihnen nicht viel Speicherplatz raubt. »VTest 1.67« läuft dann im Hintergrund ab und meldet

AMIGA DOS Info

Falls bei Ihnen ein neuer Virus auftaucht, der von Ihrem bevorzugten oder anderen Antivirenprogrammen nicht erkannt wird, so schicken Sie eine Diskette mit diesem Virus an die Firma A.P.S. Electronic. Diese wird an den Autoren von VTest 1.67 weitergeleitet. Er bindet dann neuerkannte Viren in das Programm ein. Für jeden neuen Virus erhalten Sie die neueste Version des Antivirenprogramms gratis.

PD-Werkzeugkiste

Viel Information und News aus der Szene finden Sie ab sofort in unserer Werkzeugkiste.

sich nur bei Vektorenänderungen oder einer Abweichung vom Standard-Bootblock.

Falls eine der Veränderungen eintritt, beginnt die Suche nach Viren. Der Rechner vergleicht die besagte Stelle mit ihm bekannten Virenprogrammen und gibt den Namen des Virus aus. Nun setzt eine Besonderheit des Programms ein: Der Rechner überschreibt den Virus mit Nullen, so daß kein Reset mehr durchgeführt werden muß. Doch welche Viren erkennt eigentlich der Virenspion?

Vorerst ist hier der neue Xeno-Virus zu nennen. Des weiteren testet er auf den BGS9, die verschiedenen Lamer-Versionen, Desaster und eine Palette von vielen anderen Viren. Zu dem Programm gehört auch ein Arbeitsmenü. Auf dem Bildschirm befinden sich vier Arbeitsfenster und zwei Editorbereiche zur Eingabe per ASCII-Code und Hex-Type, so daß Sie selbst als Virenterminator agieren können. Hier können

Sie nun den Bootblock in den Speicher laden und auf Viren überprüfen lassen. Dabei vergleicht das Programm nicht wie üblich ein Langwort, sondern deren drei. Mit einer weiteren Option wird der Speicher in die Bootblöcke null und eins geschrieben. Beim Kopieren von Bootblöcken ist jedoch Vorsicht angesagt, da der Rechner nicht auf Viren überprüft. Damit Sie keine voreiligen Installationen auf der Diskette vornehmen, die nicht mehr reversibel sind, bietet Ihnen das Programm die Möglichkeit, diese im Speicher vorzunehmen.

Der Befehl »Noboot« installiert den DOS-Block ohne Bootprogramm und »Install« erstellt den Original-Bootblock. Leider ist das Programm nicht speicherresident, doch dieses Manko wird durch die Tatsache, daß drei Laufwerke und zwei Harddisks angesprochen werden können, aufgehoben. Beim Auffinden neuer Viren ist mit einer weiteren Version zu rechnen. Abschließend ist noch zu sagen, daß die Kapazitäten des Programms längst nicht erschöpft sind.

(Jürgen Seibel/vb)

Für den Amiga gibt es schon viele gute Zeichenprogramme. Das beste ist ohne Zweifel »Deluxe Paint«. Den einzigen Nachteil, den dieses Programm hat, ist sein Preis. Für viele Anwender stellt dieser eine unüberwindliche Hürde dar. Abhilfe findet der Grafikbegeisterte im PD-Pool. Wo sonst? »DaVinci« ist ein Shareware-Programm, daß durch seine vielen Optionen und durch eine leichte Bedienung angenehm auffällt.

Alter Meister

Wird das Programm gestartet, öffnet sich ein Fenster, in dem die Auflösung und die Farbanzahl angegeben werden muß. Bei den Auflösungen hat man nur die Wahl zwischen »Lores« und »Medres«. Hat man sich entschieden, wird das Hauptprogramm nachgeladen. Auf der linken Seite der Malfläche klappt ein Menü herunter, ähnlich wie bei »DPaint«, in dem die verschiedenen Zeichenwerkzeuge wie Pinsel, Farben und andere ausgewählt werden können. Hat sich der Anwender für eine bestimmte Pinselform im Menü entschieden und diese mit der Maus angeklickt, dann verschwindet das Menü und gibt dadurch die gesamte Zeichenfläche frei. Will man eine neue Farbe zum Zeichnen verwenden, genügt ein Druck auf die rechte Maustaste und das Auswahlménü klappt wieder herunter. Welche Funktionen beinhaltet »DaVinci« nun eigentlich? Es stehen einem alle oft benötigten Funktionen, wie Recht-

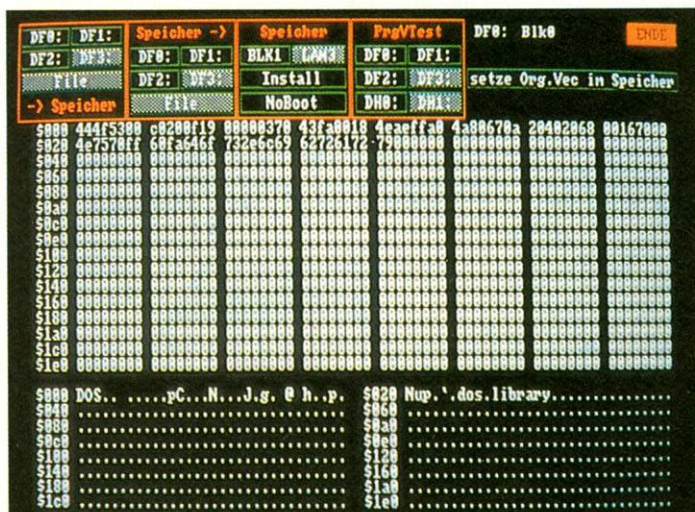


Bild 1. Anhand des Editorfeldes erkennt man, ob Viren im Speicher sind

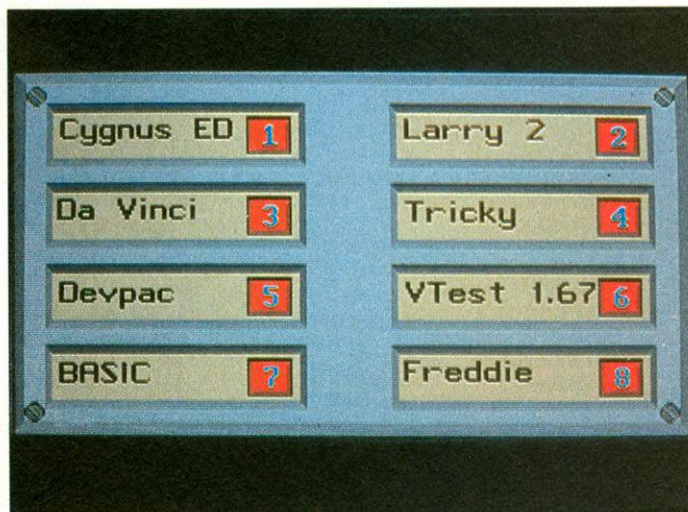


Bild 2. »DaVinci«, ein leicht zu handhabendes Malprogramm

PUBLIC DOMAIN

ecke, Linien und das Zeichnen von Ellipsen zur Verfügung. Darüber hinaus besitzt das Programm noch eine Text-, eine Füll- und – was besonders wichtig ist – eine Undo-Funktion. Was dem Autor nicht sonderlich gut gelungen ist, ist die integrierte Lupe. Sie ist, ohne zu übertreiben, als "schneckenlahm" zu bezeichnen. Man muß schon gute Nerven mitbringen, wenn man dieses Tool häufiger einsetzen will oder muß.

Wer seine Werke zu Papier bringen möchte, findet im Pull-down-Menü dafür einen besonderen Eintrag. Die Druck-Routine unterstützt die Preferences-Drucker. Es dürfte also keine Schwierigkeiten beim

Ausdruck geben. Schwierigkeiten hat man nur, wenn man mit anderen Zeichenprogrammen Bilder austauschen möchte. »DaVinci« kann nämlich keine Bilder im IFF-Format speichern. Das ist der größte Nachteil, den dieses Programm aufzuweisen hat. Am Ende bleibt ein weitgehend positiver Eindruck zurück. Wer nicht viel Geld für ein Zeichenprogramm ausgeben möchte, kommt an »DaVinci« nicht vorbei.

(Ingmar Reyer/vb)

Name: DaVinci
Quelle: Antares 62
Anbieter: R. Dombrowski

Eine kleine Übersicht der Vertrieber 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

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 Pfäeffikon ZH
Tel.: (0) 1 950 5611



KLIMAVERÄNDERUNG, WIESO ? HAUPTSACHE DIE SONNE SCHEINT !

Irrtum. Es könnte Ihnen bald zu heiß werden.

Die meisten Menschen wollen nur schönes Wetter. Und merken nichts von der allmählichen Veränderung des Weltklimas. Obwohl jeder davon betroffen sein wird. Schon jetzt gibt es genügend Gründe, sich mit den Folgen auseinanderzusetzen.

Wenn Sie wissen wollen, warum das Wetter immer merkwürdiger wird und welche Konsequenzen das für Sie hat, dann füllen Sie den Coupon aus. Wir beantworten Ihre Fragen, sagen Ihnen, wo wir uns noch engagieren.

Ja, ich will von ROBIN WOOD wissen, warum das Wetter immer merkwürdiger wird und was das für Folgen hat.

Name: _____
Straße: _____
Wohnort: _____
dmv

Weil ich helfen will, lege ich 3 Mark in Briefmarken bei. Coupon einsenden an ROBIN WOOD
Postfach 10 21 22, 2800 Bremen 1

ROBIN WOOD

Gewaltfreie Aktionsgemeinschaft für Natur und Umwelt e.V.



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

Ordnung und Übersicht schaffen die beliebten DMV Sammelmappen



Bitte Bestellkarte benutzen
DMV Verlag · Postfach 250 · 3440 Eschwege



PD aktuell

★ Nachwuchs im "Fish-Teich"

Entgegen anderslautenden Gerüchten, nach denen die neue Fish-Disketten erst für Ende des Jahres zu erwarten seien, sind die AmigaLibDisks 361 bis 370 doch schon früher fertiggestellt worden – und wie immer bringen sie frei kopierbare Software vom Feinsten. Ob das daran liegt, daß immer mehr Software von deutschen Programmierern den Sprung über den großen Teich schafft?

Schon die Fish 361 ist fast eine deutsche Alleinproduktion, sind doch abgesehen von Bruce Thomsons Grafik-Konverter »Brush_4D«, der IFF-Bilder in das Objektformat von »Sculpt 4D« wandelt, nur Werke deutscher Autoren darauf zu finden. So liegen mit dem File-Editor »FileMaster 1.20« von Roger Fischlin und dem ANSI-Editor »TextPaint« in der Version 0.97 von Oliver Wagner Updates zu "alten Bekannten" vor. Schließlich haben sich doch beide Utilities in der deutschen PD-Szene in Windeseile verbreitet. Auch von Roger Fischlins »XColor-Library«, die nicht zuletzt dank ihres komfortablen Farbrequesters Programmierern mittlerweile bestens bekannt ist, findet sich auf dieser Diskette ein Update mit Demoprogrammen in C, AmigaBasic, DevPac-Assembler und KickPascal. Und vom Bielefelder Spieleprogrammierer Peter Händel ist »The Turn« dabei, ein Knobelspiel ohne große Effekte, aber mit schönen und detailliert gezeichneten Grafiken und einem eigenen Leveleditor.

Die Fish 362 ist randvoll mit den verschiedensten Programmen. So bietet Robert Lang für die DFÜ-Besessenen mit »ArchEdge« ein Intuition-Interface für die bekannteren Archivierungsprogramme wie ARC, ZOO, LHARC oder PAK an. Rainer Salamon hat mit seinem Programm »KeyMenu«

eine interessante Lösung für alle gefunden, die Pulldown-Menüs komfortabel über die Tastatur nutzen wollen, ohne sich ständig die Kombinationen mit der Amiga-Taste merken zu müssen. Roger Fischlins bekanntes Utility »Fenster«, mit dem sich versehentlich von einem fehlerhaften Programm offen gelassene Windows problemlos schließen lassen, ist in der neuen Version 2.2 vertreten. Das deutsche Shareware-Strategiespiel »Imperium Romanum« von Roland Richter liegt in der Version 1.50e vor. Robert Broughton hat sich Gedanken über optimalen C-Code für 68020- und 68030-Turboboard-Besitzer gemacht. Seine Varianten der Lattice-C-Funktionen »memcpy()«, »memcmp()« und »memset()« sind so optimiert, daß sie so weit wie möglich auf Langwort-Instruktionen zurückgreifen, was allerdings nicht nur auf Amigas mit 32-bit-Ram Zeitvorteile bringt. Und von Martin Round stammen schließlich drei eher spielerische Programme, die diese Fish-Disk abrunden: »PUZZ« ist ein nettes kleines Programm, das IFF-Bilder in ein Schiebepuzzle verwandelt, mit »Rubik« läßt sich der verflixte Würfel nun auch mit Computerhilfe lösen, und hinter »sMOVIE« verbirgt sich schließlich ein Textanzeiger mit Soft-Scrolling, vom Autor für Video-Vorspanne und Diashow-Intros entworfen.

Gebt uns endlich Spiele für die Workbench... Das muß sich auch Ekke Verheul bei der Programmierung von »MigaMind« gedacht haben, das unter völligem Verzicht auf bunte Custom-Screens das bekannte Superhirn-Konzept auch auf die Workbench bringt. Ein ebenfalls als Workbench-Utility prädestiniertes Programm ist »WO«. Der kryptisch kurze Name steht für eine einfache Adressverwaltung, die jedoch als besonderes Feature die Möglichkeit bietet, die Daten Paßwort-geschützt zu speichern. Um den Schutz nicht

gleich wieder zunichte zu machen, fehlen folgerichtig auch die Paßwort-Routinen im ansonsten vollständig verfügbaren Source-Code. Virengelagten Besitzern von Spielen mit Bootblockladern bietet Steven Lagerweij mit »BootBase« ein weiteres Utility zum Sichern und Zurückschreiben von Bootblöcken an. Vom gleichen Autor stammen auch »RandSam« und »SampleScanner«, ersteres ein Programm zum zufallsgesteuerten Abspielen von Samples, letzters eines zum Suchen derselben auf Disketten, die nicht im DOS-Format vorliegen. Andreas Krebs hat »LabelPrint 3.5« beigesteuert, die neueste Version des bereits von früheren Fish-Disketten her bekannten Druckprogramms für Diskettenlabels. Und allen Besitzern von Hayes-kompatiblen Modems bietet Christian Fries mit »PLW« ein Utility, das alle eingehenden Anrufe protokolliert, eine kurze Ansage ausgibt und dem Anrufer die Möglichkeit gibt, eine kurze Nachricht zu hinterlassen.

Regelrecht eingeschlagen zu haben, scheint der »Pointer Animator« von Tim Kemp in den Staaten. Die Fish Disk 364 widmet sich gleich zweifach animierten Mauszeigern. Zum einen hat Bob McKain den zweiten Teil seiner Sammlung von animierten Mauszeigern beigesteuert, zum anderen hat auch die Southern Nevada Amiga Group (SNAG) einen Pointer Contest veranstaltet, dessen gesammelte Ergebnisse hier ebenfalls auftauchen. In einem weiteren Verzeichnis findet sich eine Auswahl achtfarbiger Icons, die von R.G.Tambash sogar mit Script-Files ausgestattet wurden, um sie in den richtigen Farben betrachten zu können. Offensichtlich ist das bereits ein Vorgriff auf die Möglichkeiten von AmigaDOS 2.0, das ohne Tricks und vom Betriebssystem her in der Lage ist, Workbench-Screens mit mehr als 4 Farben anzuzeigen. Aber nicht nur animierte Mauszeiger und

bunte Icons gibt es auf dieser Fish-Disk zu bewundern, im Gegenteil: Mit »DPFFT 2.2« liegt die neueste Version des umfangreichen Programmpakets zur Fourier-Analyse vor, und schließlich gibt es noch »MemLook«, das einen grafischen Einblick in die aktuelle Speicherorganisation und -aufteilung des Rechners erlaubt.

Auf Fish 365 findet sich mit »Badger« von George Kerber gleich ein weiteres interessantes Utility für Harddisk-Benutzer. Komfortabel lassen sich Termine einstellen, an die man unbedingt erinnert werden möchte, und wenn ein solcher anliegt, meldet sich das Programm bereits beim Abarbeiten der Startup-Sequence.

Buntgemischt

Ebenfalls für Harddisk-Besitzer ist »EasyBackup« gedacht, ein Backup-Programm, das zwar auf CLI-Benutzung ausgerichtet ist, ansonsten aber komfortable Features bietet. »EasyMouse« hingegen ist wieder für alle Amiga-Besitzer brauchbar, handelt es sich dabei doch um einen schlichten MachII-Verschnitt, der um einige Funktionen gekürzt wurde. In Anbetracht der Tatsache, daß »MachII« Shareware ist, mag »EasyMouse« jedoch eine Existenzberechtigung haben. Nic Wilson hat mit »DmeAsm« ein Utility für alle geschrieben, die den Devpac-Assembler in Verbindung mit Matt Dillons fantastischem Editor »DME« benutzen, das die Zusammenarbeit der beiden Programme erheblich erleichtert. Vom selben Autor stammt »TrackDos«, ein Hilfsprogramm, das beim Schreiben und Lesen von Disketten auf Trackdisk-Ebene nützlich ist. Die italienische Muchmore-Variante »View80« stellt uns Federico Giannici vor. Immerhin drei verschiedene Scroll-Geschwindigkeiten lassen befürchten, daß demnächst auf den Fish-Disketten ein noch verspielterer Textanzeiger Einzug hält. Und mit »Password« und »Udate« finden sich noch zwei Updates von George Kerber zu bereits früher auf Fish veröffentlichten Programmen. Ersteres versucht das Amiga-System durch ein Paßwort vor unbefugter Benutzung zu schützen, ein Schutz allerdings, der von

geübten Benutzern leicht umgangen werden kann. Und letzteres ist eine Unix-ähnliche Variante des Datum-Befehls, die nun auch mit dem 68030 verträglich ist.

Gepackt voll ist auch die Fish 366. Für die Spielernaturen gibt es eine 3D-Version vom altbekannten »TicTacToe«, jedoch nicht mehr in einem 3*3-Feld, sondern in einem 4*4*4-Würfel. Der Computer ist ein starker Gegner, aber durchaus zu schlagen, da es hier keine garantiert narrensicheren Spielzüge mehr gibt, mit denen immer ein Unentschieden herauszuholen ist. »IntuiFace« ist eine andere Variante eines Intuition-Interfaces zu den gängigen Archivierungsprogrammen.

Allerdings werden hier im Gegensatz zu »ArchEdge« von der Fish 362 nur ARC-, ZOO- und LHARC-Archive unterstützt. Drei stark auf den amerikanischen Markt ausgerichtete Programme sind »NDebt«, das regelmäßig in einem Fenster die aktuelle amerikani-

sche Staatsverschuldung gemäß der historischen Wachstumsrate ausgibt, und »PhoneWord« sowie »Unjumble«, welche zum einen Telefonnummern entsprechend der Alphabet-Zuordnung auf amerikanischen Telefonen, zum anderen Buchstabensalate aus Sonntagszeitungsrätseln zu vernünftigen Wörtern zusammensetzen versuchen. Nutzlos, aber witzig. »DosError« hilft bei der genaueren Beschreibung eines DOS-Fehlers, ist also eine Art erweiterter Why-Befehl. »MeMeter« ist eine andere Variante eines Speicherplatzanzeigers.

Im Gegensatz zu »MemLook« von Disk 364 wird hier der belegte Speicher nur in Form eines schmalen Füllstandszeigers dargestellt. Einmalig jedoch das Feature, die Menge derzeit freien Speichers zu merken, ein Programm zu starten und zu beenden und danach mit den gemerkten Werten zu vergleichen, um herauszufinden, ob das Programm seinen Speicher auch kom-

plett zurückgegeben hat. Hilfe beim Berechnen von Krediten und Rückzahlungsraten bietet »LoanCalc«, das ansonsten leider recht stark auf den amerikanischen Markt ausgerichtet ist. Und »PrintStudio« von Andreas Krebs kann sich fast schon als der »Shareware-Printmaster« bezeichnen, so vielfältig sind die Möglichkeiten, die das Programm zum Speichern von Grafiken sowie zum Ausdruck derselben auch in Kombination mit Text bietet.

Ein weiteres Peter-Händel-Spiel begrüßt uns mit dem bereits in der letzten AmigaDOS-Ausgabe vorgestellten »Tricky« auf der Fish 367. Mit »GwPrint« findet sich hier auch ein komfortables Utility zum Ausdrucken von Texten, bei dem nahezu alle möglichen Parameter komfortabel über ein Intuition-Interface einzustellen sind. Ebenfalls mit Intuition-Interface ist der »HyperDialer« von David Plummer ausgestattet, eine komfortable Adressen- und Telefonnummernverwaltung, die in Verbindung mit

einem Hayes-kompatiblen Modem auch automatisch wählen kann.

Eine nette grafische Simulation der im zweiten Weltkrieg gebräuchlichen Enigma-Codiermaschine ist von Gaylan Wallis programmiert worden. Jean-Marc Nogier liefert mit »SCM« (Screen Color Modifier) eine weitere Variante eines Palette-Tools, das die Modifikation der Bildschirm-Farben erlaubt.

Vollgepackt

Ebenfalls mit Grafik befaßt sich »SuperView«, der komfortable Viewer für IFF-Files aller Arten. Angezeigt werden in der neuesten Version 3.0 sämtliche Display-Modi mit Color-Cycling und automatischer Overscan-Erkennung. Desweiteren unterstützt »SuperView« noch das AmigaBasic ACBM-Format und die diversen Anim-Formate, von denen es allerdings nur das Startbild anzeigt. Chemiker werden sich für die neue Version 2.0 von

POWER PC-BOARD: BEI ROSSMÖLLER INCL. KNOW HOW-GARANTIE

Lange Zeit Top Secret, nun der Topseller: Das Power PC-Board, das den AMIGA® 500 IBM®-kompatibel macht. Kenner kaufen es für nur 798 DM bei ROSSMÖLLER. Denn bei ROSSMÖLLER ist eine Know How-Garantie gleich im Preis inclusive. Als erfahrener AMIGA®-Zubehörentwickler und renommierter Hardware-Preisträger bietet ROSSMÖLLER fundiertes Know How. Zum Beispiel an der Telefon-Hotline: Dort sind die Entwickler unserer AMIGA®-Produkte von Montag bis Freitag (16-17 Uhr) ganz Ohr für Ihre Fragen. ROSSMÖLLER – Know How- und Service-Qualität aus einer Hand. Kennen Sie eine bessere Adresse für den Kauf Ihres Power PC-Boards?

ROSSMÖLLER
COMPUTER-TECHNIK

AMIGA
Köln 90

Halle 10, Stand 1023



**Der neue Katalog ist da!
– Jetzt anfordern!**

Roßmüller Handshake GmbH · Neuer Markt 21 · 5309 Meckenheim · Tel. 02225/2061-62-63

»Elements« auf der Fish Disk 368 interessieren. In der neuen Version sind nun gegenüber der Version 1.3a von Fish 297 auch allgemeine Informationen zur jeweiligen Gruppe und Periode verfügbar, sowie ein Testmodus, in dem das Programm verschiedene Informationen über das jeweilige Element vom Benutzer erfragt. Vom gleichen Autor (Paul Thomas Miller) stammt auch »GraphicsPak«, ein Funktionspaket zur Unterstützung verschiedener allgemeiner grafischer Operationen, das auch in den Demoprogrammen »PopMenu« und »ListWindow«, die ebenfalls auf dieser Diskette zu finden sind, verwendet wird.

»PopMenu« ist seinerseits nur ein auf »GraphicsPak« aufbauendes Funktionspaket zur Unterstützung von Popup-Menüs, die im Gegensatz zu Pull-down-Menüs nicht aus der Menüzeile herunterklappen, sondern da erscheinen, wo der Mauszeiger sich gerade befindet. Und »ListWindow« bezeichnet einen Implementationsversuch der vom Macintosh her bekannten Auflistungsfenster, der ebenfalls auf »GraphicsPak« zurückgreift. Das fünfte Miller-Programm im Bunde ist »SuperMenu«, ein komfortables Auswahlmenü für mehrere Textfiles, das hilfreich ist, wenn man ein bestimmtes Textfile oder eine bestimmte Stelle einfach auf Mausclick statt mit vielen Tippereien ansprechen will. Besitzer von Postscript-Druckern finden mit »Lila« ein komfortables Druckprogramm, das sogar wahlweise Ausdrücke im Hoch- oder Querformat sowie mehrspaltige Listingausdrücke erlaubt und selbstverständlich auch die üblichen Parameter wie Kopf- und Fusszeilen, Seitennummerierung und andere bietet. Kjell Cederfeldt hat eine interessante Lösung gefunden, normalerweise nur aus dem CLI startbare Programme auch von der Workbench aus startbar zu machen. Sein »NewEx« funktioniert zwar im Prinzip ähnlich wie »IconX« oder »Xlcon«, hat jedoch den Vorteil, daß die nötigen Informationen im Tool-Icon selbst übergeben werden können, und kein Project-Icon samt Batchfile erzeugt werden muß. Abgerundet wird diese Fish-

Disk dann durch »SysInfo«, ein kurzes Programm, das Informationen über das eigene Amiga System ausgibt und einige kleine Benchmarks durchführt, sowie »Today«, eine Portierung eines IBM PL/1-Programms, das wichtige und weniger wichtige Termine speichert und rechtzeitig daran erinnert.

Begeisterte Aquarium-Nutzer (Aquarium erschien auf Fish 301 und ist die derzeit komfortabelste Datenbank zur Suche nach einem bestimmten Programm auf den Fish-Disketten) werden an der Fish Disk 369 ihre helle Freude haben, enthält sie doch die komplette Datenbank für alle Disketten bis zur Diskette 360, während die beim Programm mitgelieferte aus verständlichen Gründen nur die Daten bis zur Diskette 300 bieten konnte. Der deutsche Programmierer Andreas Schilbach, bisher in den Netzwerken durch sein Programm »XprTransmit« eher als »GOONIE@MUSIKBOX.ZER« bekannt, steuerte neben der Version 1.0 desselben auch einen Screenhack namens »Flip« bei, den man sich aber selbst mal ansehen sollte. Die von UNIX-Rechnern bekannten »Fortune«-Sprüche, die man jedesmal beim Einloggen in die Maschine präsentiert bekommt (zumindest bei den humorvolleren unter den Systemverwaltern), lassen sich mit »Fortune« von George Kerber auch auf dem Amiga installieren. Die neue Version 2.04g erlaubt jetzt wahlweise auch das Starten von der Workbench, sinnvollerweise sollten zumindest Festplattenbesitzer »Fortune« aber an geeigneter Stelle in ihrer Startup-Sequenz einbauen. Beim Analysieren fremder und beim Debuggen eigener Programme versucht »Spy« zu helfen, indem es jeden Funktionsaufruf in der DOS- oder in der ExecLibrary mit den Parametern und dem Ergebnis auf dem Bildschirm ausgibt. Und wer auf einen Großrechner der VAX/VMS-Serie Zugriff hat, findet in »VAXterm« von Tuomo Mickelsson eine fast vollständige Emulation eines VT220-Terminals, das sowohl in den unterstützten Features als auch bei der Benutzerschnittstelle einem echten sehr nahe kommt. Nur ein Programm befindet sich schließlich auf der

Fish 370, der letzten der zehn neuen Fish-Disketten, im Gegensatz zu einigen der übergut gefüllten Disketten davor. Sie wird völlig von dem neuesten Update zur »SKsh« von Steve Koren gefüllt, einer komfortablen Shell-Umgebung mit einer Funktionsvielfalt, die längere Einarbeitung erfordert, wenn man nicht schon mit dem Unix-Vorbild »ksh« vertraut sein sollte. Neu in Version 1.5 sind gegenüber der letzten Version 1.4 von Fish 342 unter anderem selbst definierbare Tastaturbelegungen, ein AREXX-Interface (das sich zunehmend auch in frei kopierbaren Programmen durchsetzt), etliche neue interne Kommandos, freischaltbare Joker-Ersetzung of Shell-Seite und ein Mini-Preparser für Execute-Dateien.

(HolgerLubitz/vb)

★ Imploder

Bei unserer Vorstellung der Diskette »Special 41« in der AMIGA DOS 9'90 ist uns leider ein kleiner Irrtum unterlaufen: Das Programm »Imploder« ist nicht PD, es handelt sich dabei um ein kommerzielles Programm, das von der Firma Discovery Software vertrieben werden sollte. Sollte deshalb, da die Firma nicht mehr existiert und die Rechtslage bezüglich dieses Programms momentan im Unklaren ist. Der Programmierer ließ aber verlauten, daß eventuell mit einer PD-Version zu rechnen sei.

★ TextPlus

Von der momentan besten Shareware-Textverarbeitung »TextPlus« ist inzwischen die Version 2.2 erschienen, die mit einem wesentlich verbesserten Filerequester, einer gänzlich neuen Tastaturbelegung, einer Iconify-Funktion und diversen bereinigten Fehlern alle Ansprüche erfüllt, die an eine einfache Textverarbeitung gestellt werden. Zu finden ist das Programm auf der Ausgabe 24 des Freeware-Diskmagazins AmigaJUICE.

★ AmigaBASIC

Wer auf dem Amiga 3000 Probleme mit AmigaBASIC-Programmen hat (viele PD-Disketten starten direkt ins BASIC durch), der sollte das altbekannte »RUN NOFASTMEM« in die Startup-Sequenz einfügen. AmigaBASIC benutzt bisher nämlich nur 24-Bit-Adres-

sen, was auf dem Amiga 3000 mit seinem größeren Adressraum zum Guru führt.

★ Grafik

Grafik-Interessenten sollten sich einmal die von Roger Hassler ins Leben gerufene PD-Serie »Amiga Vice« anschauen. Auf den Disketten finden sich praktisch Grafiken jeder Art, unter anderem auch einige sehenswerte Raytracing-Grafiken von Thomas Schäfer.

★ Neue Serie

Eine neue Serie stellen die Soft-News-Disketten dar. Die Zusammensteller fassen auf diesen Disketten Software zusammen, die durch die Datenetze geistert und noch nicht »regulär« auf einer PD-Disk erschienen ist. Freunde brandaktueller Software jeder Art sollten sich diese Serie einmal zu Gemüte führen.

★ Diskettenmagazin

Neben AmigaJUICE, GetIT und anderen Diskettenmagazinen existiert seit kurzem Powerbrei als Organ der Usergruppe TUC. Wie auch bei den meisten der Konkurrenten wird »ZeigE« als Anzeigeprogramm verwendet. Inhaltlich setzt sich Powerbrei allerdings erheblich von der Masse ab, so daß es durchaus zu den qualitativ höherwertigen Magazinen zu rechnen ist.

★ Demo-Versionen

In der AMIGA DOS 9'90 stellen wir PD-Demo-Versionen vor. Inzwischen existiert eine entsprechende Diskettenreihe. Zum Zeitpunkt der Drucklegung umfaßte diese bereits 31 Disketten. Die Praxis, PD-Demoversionen zu erstellen, scheint sich bei den Softwarehäusern inzwischen durchzusetzen, mit Vorteilen für den Anwender.

★ Neue Share in Kürze

Folgende Programme werden bald aus der Share-Reihe zur Verfügung stehen:

- Super Schachprogramm
- Plattenliste (Top-Verwaltung von Schallplatten, CD und MC)
- Pythagoras - ein Spitzen-Matheprogramm

Ausführliche Tests folgen.

(Oliver Wagner/vb)

Das oberste Gebot im Diskettenhandling ist wohl das Kopieren. Bei der Überflutung von den allseits gehäßen Viren und den unvermeidlichen Read-Errors ist der einzige Schutz gegen diese Misere das Duplizieren von Disketten. Aus diesen Gründen haben wir einige Kopierprogramme aus der Public Domain getestet.

Jürgen Seibel

Kopieren fast zum Nulltarif

Die Public Domain ist ein schier unerschöpflicher Quell guter Programme. Besonders nützliche Utilities sind bei den Amiganern sehr gefragt.

■ Dfc

Nach dem Einladen des Programms »Dfc« erscheint auf dem Bildschirm ein Fenster mit einer Vielzahl an Gadgets. Darüber können die gesamten Funktionen des Programms gesteuert werden. Auf vier Destination-Laufwerken können Sie gleichzeitig kopieren. Dieses Programm realisiert also Multitasking auf dem Amiga in reinster Form. Natürlich können auch vier Disketten gleichzeitig formatiert werden, womit sicherlich eine immense Zeitersparnis verbunden ist. Beim Kopieren können Sie »Verify« einbeziehungsweise ausschalten. Eine besondere Option ist der »Buffer«. Der »Buffer« erlaubt Ihnen, falls genügend Speicher vorhanden ist, die Daten der gesamten Diskette in das RAM abzulegen.

■ FCopy

Nach der Aktivierung von »FCopy« werden am unteren Bildschirm alle Befehle anklickbereit per Maus oder über die Tastatur aufgelistet. Das Programm erkennt automatisch bis zu vier Laufwerke. Je-

doch müssen zur Betriebsbereitschaft von »FCopy« mindestens zwei Laufwerke oder ein Laufwerk mit RAM-Disk vorhanden sein. Die in die Workbench implementierte resetfeste RAM-Disk wird beim Kopieren vollständig unterstützt. Eine kontinuierliche Überwachung des Kopierens ist durch eine »Verify-Funktion« und die grafische Auflistung der ganzen Diskette gewährleistet. Beim Kopiermodus bieten sich dem Benutzer gleich zwei Alternativen, das »BAMcopy« beziehungsweise »Trackcopy«.

Beim BAM-Kopiermodus wird zu Beginn des Kopierens die Bitmap der Diskette analysiert. Aufgrund der Auswertung werden dann nur die belegten Blöcke kopiert, was eine immense Zeitersparnis beim Kopieren bedeuten kann. Damit gibt sich der Track-Kopiermodus aber nicht zufrieden. Entgegen dem spartanischen BAM-Modus kopiert er die gesamte Diskette. Auf dem Bildschirm werden die belegten Tracks mit einem Pluszeichen

und die unbelegten mit einem Minuszeichen markiert. Die wohl gelungenste Option, »Optimizer«, birgt folgenden Kopiervorgang: Zuerst wird die Verteilung der Blöcke (Bitmap) analysiert. Auf den ermittelten Daten basierend, errechnet der Computer eine neue Blockstruktur, nach der die Source-Diskette auf die Destination-Diskette kopiert wird. Abschließend ist noch zu sagen, daß man einzelne Disketten formatieren kann und das Programm Multitasking-fähig ist.

■ SDBackup

Dieses Programm präsentiert sich nicht wie gewohnt mit einem Gadget-Menü. Es wurde eine andere Steuerung gewählt. Die Optionen, die das Programm steuern, werden durch Befehlseingabe im CLI erreicht. »SDBackup« findet besonders beim Kopieren im Umgang mit der Festplatte Anwendung. So ermöglicht ein Befehl das Kopieren von der Festplatte. Hier liegt das ei-

gentliche Problem eher beim Diskettenwechseln, da ja die Festplatte über immense Speicherkapazitäten verfügt. Beim Kopieren können die Daten gleich gecruncht werden, was natürlich eine große Diskettenersparnis zur Folge hat. Unter dem Begriff »Zurückspeichern« verbirgt sich nichts anderes, als daß eine gewünschte Anzahl der soeben kopierten Disketten wieder auf die Festplatte zurückgeschrieben wird.

■ TurboBackup V1.0

Nach dem Aufruf von »TurboBackup« baut sich ein Screen mit vier Gittermustern auf, die nebeneinander angeordnet sind. Diese Gittermuster stellen die jeweiligen Laufwerke dar. Es folgt danach die Deklaration der einzelnen Diskettenstationen. Mit Hilfe von Gadgets erreicht man die Einstellung der Source-, Destination- und der ausgeschalteten Laufwerke. Auch bei diesem Programm haben die Multitasking-Fähigkeiten ihren Einzug gehalten, da beim Kopieren bis zu drei Laufwerke gleichzeitig beschrieben werden können. »TurboBackup« kopiert jedoch nur mit der Option »Verify«. Ein Manko ist die Tatsache, daß es versäumt wurde, so wichtige Optionen wie das Formatieren in das Programm einzubinden.

Die vorgestellten Programme sind auf den verschiedensten PD-Reihen vertreten.

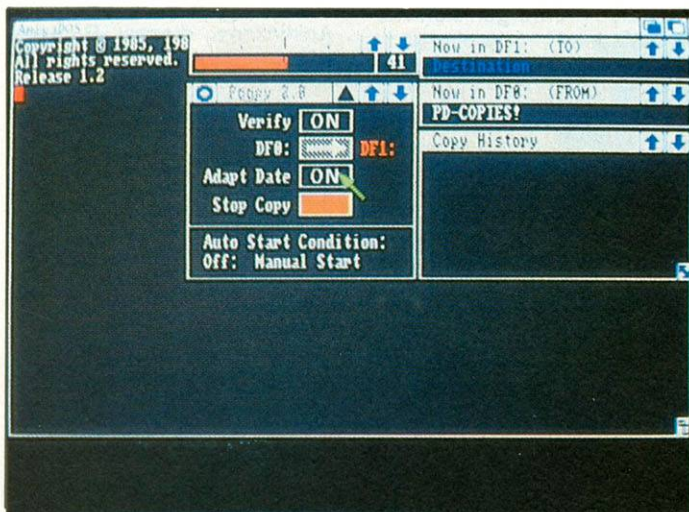


Bild 1. »PCopy« – Zwei Fenster zeigen ständig den Inhalt der Diskette an

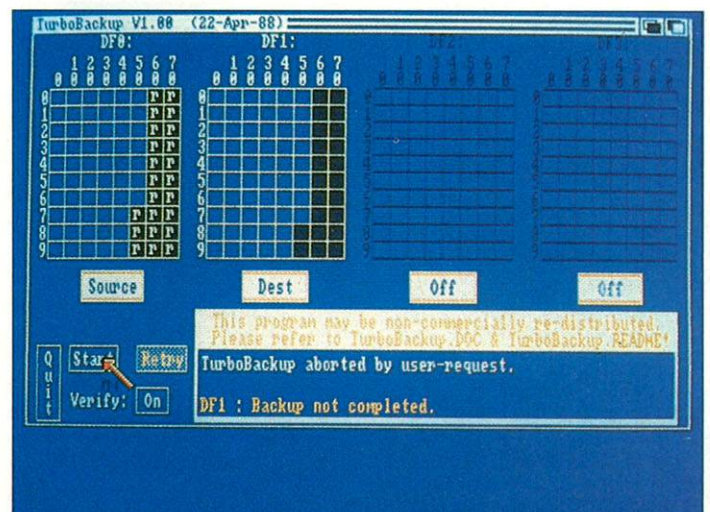


Bild 2. Mit »Turbo Backup V1.0« ist ein Kopieren mit drei Laufwerken gleichzeitig möglich

■ Quickcopy

Ein Programm für besonders harte Computerfreaks. »Quickcopy« ist etwas für die eiskalten Kopierer, die nur die Disketten ins Laufwerk schieben und sich an den Geräuschen des ratternden Schreib-Lesekopfes ergötzen. Das Programm ist auch nur mit zwei Laufwerken lauffähig. Auf dem Bildschirm erscheinen zwei Felder und zwei Gadgets. Die Felder zeigen die Blöcke der Source- und Destination-Disk an. Die Gadgets sind zur Aktivierung des Kopiervorgangs nötig. Nun gilt es, die Source-Disk in Laufwerk DF0: und die Destination-Disk in Laufwerk DF1: zu legen. Nach dem Anklicken des Gadgets »Start« beginnt der Kopiervorgang. Mit »Stop« kann der Kopiervorgang dann unterbrochen werden.

■ MRBackup

Das Programm wurde zu dem Zweck geschrieben, alle Daten der Festplatte auf Disketten zu sichern. Denn wenn sich einmal ein Virus auf der Festplatte verirrt, sind die Daten zerstört. Vorab öffnet sich ein Fenster, indem ein Stop-Zeichen, mehrere Gadgets und weitere Fenster zu erkennen sind. Zuerst muß der Ursprungspfad »Homepath« und der Sicherungspfad »Backup Path« definiert werden, damit eine Rekonstruktion der Daten gewährleistet ist. Mit dem Kopieren von ganzen Directories bietet das Programm ein hohes Maß an Komfort. Zuletzt kann der Listpfad zum Drucker

geschickt werden. Der Clou von »MRBackup« erscheint nach dem Anwählen des Menüpunktes »Speak«. Nach Aktivierung ist der Sprachprozessor angesprochen, so daß von nun an der Computer jede Operation in digitalisierter Sprache dokumentiert.

■ Thunder-Copy V0.1

Das Programm besticht zunächst in seiner grafischen Darstellung durch ein ansprechendes Erscheinungsbild. Die Voraussetzung für das Programm ist ein zweites Laufwerk. Nun bieten sich dem Benutzer vier Optionen an: »Copy-Disk«, »Check-Disk«, »Format-Disk« und »Exit«. Vor dem Anwählen der Optionen wird dem Benutzer die Möglichkeit eröffnet, den Start- und End-Track frei zu wählen. Dies ist vor allem beim gezielten Kopieren oder Überprüfen von Disketten dienlich. Nachdem Sie das Source- und Destination-Laufwerk, die jeweils von DF0: bis DF3: reichen, bestimmt haben, steht dem Kopieren nichts mehr im Wege. Während des Kopiervorgangs gibt ein Feld, das die 80 Tracks der Diskette darstellen soll, über den Status Auskunft. Darüber hinaus werden mögliche Fehler angezeigt, die sich unter Umständen auf der Diskette befinden können. Ein negativer Aspekt bei »Thunder-Copy V0.1«: Die so begehrte Verify-Option entfällt. Dies soll das Programm jedoch in seiner Präsentation nicht schwächen, zumal es sowohl gra-

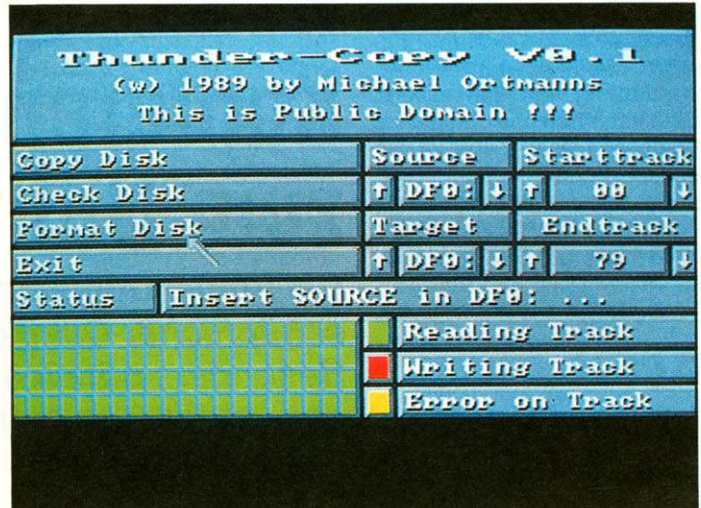


Bild 3. »Thunder-Copy V0.1« – ein optisch gut aufbereitetes Programm

fisch als auch programmier-technisch voll gelungen ist.

■ AHA-Copy

Gegenüber »Thunder-Copy« zeigt sich »AHA-Copy« vorerst in einem eher schlichten Kleid. So füllen lediglich drei Optionen, von denen eine die Beendigung des Programms zur Folge hat und eine andere den Kopiervorgang startet, den Bildschirm. Jedoch wird man nach dem Anwählen der Option »Change-Parameters« wohl sehr überrascht sein, da hier dem Benutzer so manche Schmankele offeriert werden. Von DF0: bis DF3: können die Laufwerke der Destination- und Source-Disk frei festgelegt werden. Nun gelangen wir zu den besonderen Optionen des Kopierprogramms.

Hinter dem Begriff »Type of Operation« verbergen sich vier verschiedene Kopiermodi: »Normal-Copy«, »Nibble-Copy«, »Parameter-Copy« und das ständig notwendige »IBM-Format-Copy«. Dem eingefleischten Amiganer wohl bekannt, bietet das AmigaDos nur die Möglichkeit, von Track 0 bis 79 die Disketten zu beschreiben. Entgegen diesen Gesetzmäßigkeiten ermöglicht nun »AHA-Copy« das Schreiben und Lesen von 84 Tracks (0-83), wobei auch hier der Start- und End-Track frei wählbar ist. Mit der Option »Single oder Double-Sided Copy«, die sicherlich nicht näher erläutert werden muß, schließt sich das fulminante Angebot des Programms.

■ PCopy

Das Programm »PCopy« kann sowohl aus dem CLI als auch aus der Workbench gestartet werden. Es ist jedoch nur mit mindestens zwei Laufwerken lauffähig. Zu Beginn werden die angeschlossenen Laufwerke in einem Fenster durch Anklicken per Maus definiert. Zwei kleine Fenster zeigen ständig den Inhalt der Diskette an. Das Programm wird vollständig über diverse Gadgets gesteuert. Auch hier ist die Funktion »Verify« integriert. Eine besondere Option bietet das »AutoStart«. Sobald »PCopy« ermittelt, daß eine Diskette gewechselt wurde, startet der Computer den Kopiervorgang. Ein Manko folgt jedoch bei der Unterbrechung des Kopiervorgangs. Es erscheinen die Gadgets »Abort« und »Retry«. Das Gadget »Retry« ist jedoch irrelevant, da das Programm, trotz Anklickens dieses Punktes, den Kopiervorgang nicht fortsetzt.

■ TreasuresCopy

Man könnte es als eins von Hunderten bezeichnen. Dennoch wartet das Kopierprogramm mit einigen beachtlichen Optionen auf. Sie können mit ein bis zwei Laufwerken kopieren. Hier ist es gleich, welche Laufwerke Sie ansprechen. Nach der Eingabe des Start- und End-Tracks haben Sie schon alle Voreinstellungen getätigt und sogar auskosten. In einem Feld werden die eingelesenen Tracks grafisch angezeigt. (vb)

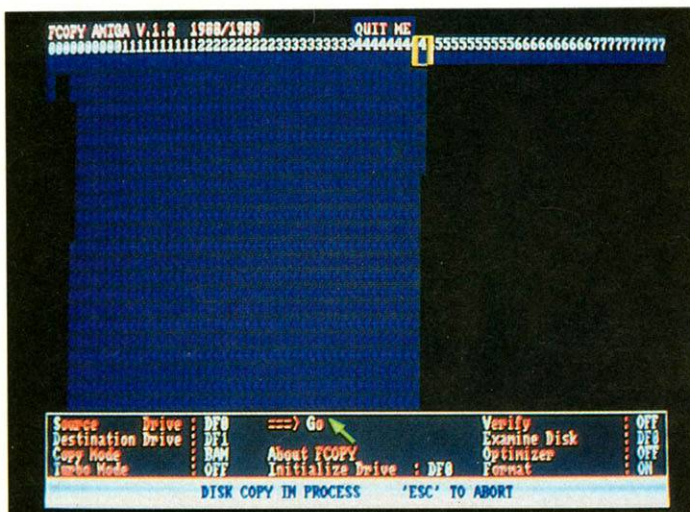


Bild 4. »FCopy« – alle Befehle sind über die Menüleiste am unteren Bildschirmrand zu erreichen

PD-Spieleshow

Vielfalt ist Trumpf bei der Vorstellung neuer PD-Spiele. Aus der Vielzahl guter Programme haben wir für Sie einige Bonbons herausgefischt.

Das Spiel kenn' ich doch, wird sich mancher sagen, dem die Fish-Disk 351 in die Hände fällt. Auf der Diskette ist eine PD-Version des Kommerz-Spiels »Pipemania« vorhanden, das diesem in keiner Weise nachsteht.

Pipeline

Der Spieler hat die Aufgabe, auf einem zehn mal sechzehn Felder großen Spielfeld Rohrleitungen an einen Hydranten zu legen, damit das Wasser (oder was auch immer) hindurchfließen kann. Die Rohrteile hierzu kommen, zufällig gewählt, von einer Art Fließband und müssen alle verbaut werden. Nach einer bestimmten Zeit fängt das Wasser an zu fließen. Man bekommt solange Punkte gutgeschrieben, wie das Wasser fließt. Hat es ein offenes Rohrende erreicht, geht es in die Auswertung und die Punkte werden in eine speicherbare Highscore-Liste eingetragen. War das gebaute Rohr lang genug, kommt man in den nächsten Level und das Spiel beginnt von vorn, an-

dernfalls hat man das Spiel verloren. Natürlich steigt mit den Levels auch der Schwierigkeitsgrad. So erhöhen sich beispielsweise die Anzahl der Mauern, um die herumgebaut werden muß, oder das Wasser beginnt früher zu fließen. Aber der Spieler hat durch stehengebliebene Zeit, stockendes Wasser und andere kleine Netigkeiten ein wenig Gelegenheit, sich auszuruhen oder sich wieder einen Vorsprung zu "legen".

Ohne Zweifel dürfte hier wohl der beste Pipemania-Clone vorliegen, der derzeit auf dem PD-Markt vorhanden ist. Das Spiel unterscheidet sich, mal abgesehen von Grafik und Sound, nur wenig von seinem Vorgänger. Es wurde sogar noch durch einige Features ergänzt. Beispielsweise wurde das Spielfeld vergrößert und einige Extras hinzugefügt. Das geniale Spielprinzip verfügt über den Suchtkoeffizienten eines »Rainbow Island«. Man sollte sich schon überlegen, ob es nicht lohnender wäre, sich anstatt des Originals die PD-Version zuzulegen. Der

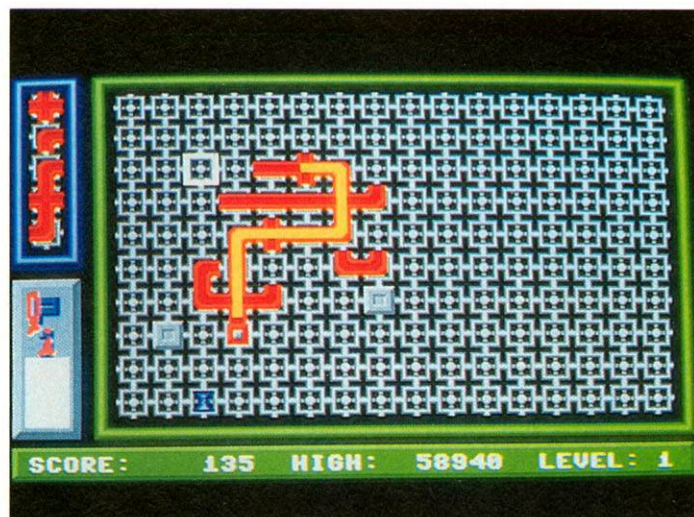


Bild 1. Spielen Sie Klempner, damit das Wasser wieder läuft

Spielepaß ist sicherlich der gleiche und bei einem Preis weit unter der 10-DM-Grenze kommt man mit dieser Lösung wohl sehr viel billiger davon.

(Timo Siebert/vb)

Name: Pipeline
Quelle: Fish 358
Anbieter: Hermanns & Kommelter und andere PD-Anbieter

Mit »TRON« bekommt man wieder einmal ein Spiel in die Hand, was unter der Kategorie "XXX, die tausendunderste!" zu finden ist.

TRON

»TRON« erinnert stark an den guten alten »Fallensteller« (Mancher C-64-Freak dürfte bei diesem Namen wohl ein Wehmütstränchen verlieren). Das Spielprinzip ist ebenso einfach wie fesselnd. Bis zu sechs Spieler (von welchen dann mindestens vier vom Computer gesteuert werden) versuchen, sich gegenseitig "in die Pfanne zu hauen". Dies tun sie mittels eines "Lichttrades", das eine farbige Spur hinterläßt. Stößt ein Mitspieler gegen diese Spur, ist für ihn das Rennen gelaufen.

Sieger des Spieles ist, wer bis zuletzt übrig bleibt. Doch nicht nur die Spuren anderer Mitspieler machen dem eigenen Spieler das Leben schwer, auch Hindernisse wie beispielsweise Mauern setzen einiges an Geschick voraus. Mit dem Feuereknopf läßt sich die

Geschwindigkeit der Fahrzeuge verdoppeln. Zu den Computergegnern läßt sich nur sagen: Mit viel Liebe wurden die nun wirklich nicht programmiert. Sie bewegen sich äußerst primitiv über das Spielfeld und laufen stur wie ein Hard Error gradlinig in eine Richtung. Dazu kommt noch, daß sie dem Spieler weitaus überlegen sind, denn sie kommen auch durch noch so kleine Lücken, was dem Spieler aufgrund der groben Steuerung nur selten gelingt. Über das Menü sind Spieleranzahl, Geschwindigkeit, Anzahl der Hindernisse, Tracktracing (ob sich die gezogene Spur nach einiger Zeit wieder auflösen soll) und PAL- oder NTSC-Mode einstellbar. Das Spiel wurde in GFA-BASIC geschrieben, worunter die Geschwindigkeit, insbesondere bei 6 Mitspielern, ein wenig zu leiden hat, ist aber trotzdem spielbar. Im großen und ganzen kann man sagen, daß »TRON« auch trotz einiger Unschönheiten eine Menge Spaß machen kann. Insbesondere die Duelle mit einem menschlichen Mitspieler "gehen tierisch ab". Wäre die Steuerung präziser ausgefallen und die Computergegner ein bißchen komplexer programmiert worden, würde ich dieses Spiel noch mehrmals aus der Schublade kramen.

(Timo Siebert/vb)

Name: TRON
Quelle: Fish 355
Anbieter: Hermanns & Kommelter und andere PD-Anbieter

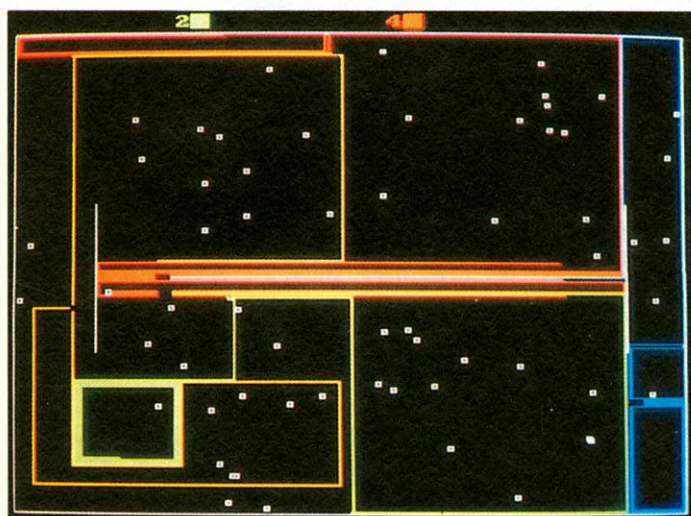


Bild 2. Kampf der Lichtstrahlen



Bild 3. Mit »Frantic Freddie« einen Trip in vergangene Zeiten

Umsetzungen vom Amiga auf den kleinen Bruder C-64 werden immer seltener. Ebenso geht es den Umsetzungen auf den Amiga. Doch hier schlägt ein Spiel in diese Kerbe, das man ruhigen Gewissens als das absolute Kultgame der frühen Achtziger für den C-64 bezeichnen kann.

Frantic Freddie – Ein totgeglaubter Mythos lebt wieder auf

Bei »Frantic Freddie« handelt es sich um ein Jump'n'Run-Spiel, wie es seinerzeit auf dem C-64 üblich war. Freddie hat die Aufgabe, alle Schätze aufzusammeln, um ins nächste Level zu gelangen. Hierbei wird er aber von Monstern und anderen Börsartigkeiten behindert. So trifft er auf Vampire, glubschäugige Paranoiden und Spinnen, deren Berührung auf alle Fälle tödlich ist. Zu diesen Gegnern ist zu sagen, daß sie äußerst geschickt programmiert wurden, und es ist kein Leichtes, diese auszu-tricksen. Auch stellen die Treppen, mit denen er Etagen erklimmen kann, ein Manko dar. Er kann nicht einfach an ihnen vorbeilaufen, sondern nur durch Überklettern in den nächsten Raum gelangen. Dies ist sehr hinderlich, da in der oberen Etage meist schon ein Alien wartet. Außerdem setzt die Steuerung ab und zu für einen Augenblick aus (entweder eine Böswilligkeit des Programmierers oder meines

Joysticks), was manchmal zu Panikanfällen eskaliert. Aber damit hat der Programmierer das C-64-Feeling exakt getroffen. Überhaupt wurde das komplette Spiel Eins zu Eins umgesetzt. Ein großes Lob an die Programmierer, die sich von der Hardware des Amiga nicht dazu verleiten ließen, das gute alte Frantic-Freddie-Flair durch programmierte 68000er-Technik zu stören. Der Sound, bei dem nur reine Wellenformen und keine Samples benutzt wurden, hat zwar etwas nachgelassen, aber dafür ist die Grafik ein bißchen klobiger geworden. Sogar der Zeichensatz wurde konvertiert, um den User voll und ganz in frühere Zeiten zu versetzen. Selbst das Schönste am Spiel, die »Interludes«, in denen Freddie von bösen Fröschen (oder was immer dieser Haufen Pixel auch sein soll) gejagt wird und mit Hilfe von reinsten Rechteckformen »Don't bring me down« aus den Boxen tönte, wurde nicht vergessen. »Frantic Freddie« lebt durch Nostalgie, und davon gibt es in dieser Konvertierung eine ganze Menge! Bleibt nur zu hoffen, daß dieser Trend anhält, und ich meinen C-64 endlich verkaufen kann!

(Timo Siebert/vb)

Name: Frantic Freddie
Quelle: AOW
Anbieter: Digital Marketing

Ich mag Knobelspiele. Egal, welcher Art sie sind, nur sinn-

voll und fesselnd müssen sie sein. Es gibt nichts Schöneres, als sich abends nach getaner Arbeit das Hirn mit irgendwelchen Logikspielen in Einzel-segmente aufzuspalten. Aber, wie gesagt, die Spiele sollten Pfiff haben.

Tricky – wohin des Wags, AMIGA-BALL?

Nicht alle »PD-Knobis« können von sich behaupten, Renner auf dem Gebiet des »Überlegenskampfes« zu sein, doch ab und zu findet man ein Goldstück unter dem Bleihaufen, in diesem Fall heißt das Goldstück »Tricky«. Eine kleiner Amiga-Ball muß über bestimmte Spielsteine rollen und diese somit entfernen. Was das mit Knobelei zu tun hat, wollen Sie wissen? Nun, die Spielsteine haben unterschiedliche aufgezeichnete Bilder, wie Amiga-Mäuse, Disketten, Joysticks und Endlospapier, und der gute Ball darf nur Bilder einer Art überrollen. Um an die nächste Bilderreihe zu kommen, muß der letzte angerollte Stein einer der neuen Reihe sein. Das klappt jedoch nicht immer, denn viele Reihen bestehen nur aus einer Komplettsérie. Damit der Ball die Richtung ändern kann, gibt es die Pfeilsteine, die in Uhrzeigersinn gedreht werden können. Der Programmierer ist allerdings von der listigen Art, denn jede Drehung setzt einen

Zähler um eins herab. Irgendwann steht der Zähler dann auf Null, mit Drehen »is nix mehr« und man braucht dringend einen Beruhigungstee. Nun gibt es noch die Joker, und zwar deren drei, mit denen irgendeine Steinreihe angesprochen werden kann. Aber wie gesagt, drei Joker sind es bloß, der vierte wird vergeblich gesucht. Keine Drehung mehr, kein Joker mehr, keine Möglichkeit mehr – so einfach ist das. Über den verfluchten zwölften Level stolpert die halbe Redaktion immer noch.

»Tricky« enthält übrigens 100 Levels. Bis man die alle durchgespielt hat, fließt viel Wasser im Rhein an Köln vorbei. Wer trotzdem mehr braucht, dem steht ein Level-Editor zur Verfügung, mit dem man sich neue Gemeinheiten ausdenken kann. Ein kleiner Tip: Auch die Level, die zu schwer erscheinen, lassen sich mit dem Editor »nachbearbeiten«.

Mit »Tricky« erhält man ein interessantes, kniffliges und ansprechendes Spiel, das man auch mit mehreren Spielern angehen kann. Spielstände können auch zwischengespeichert werden, so daß man nicht immer von vorn anfangen muß. Spiele wie »Tricky« wünsche ich mir öfters auf PD-Disketten.

(jb/vb)

Name: Tricky
Quelle: Time 10
Anbieter: A.P.S. Electronic

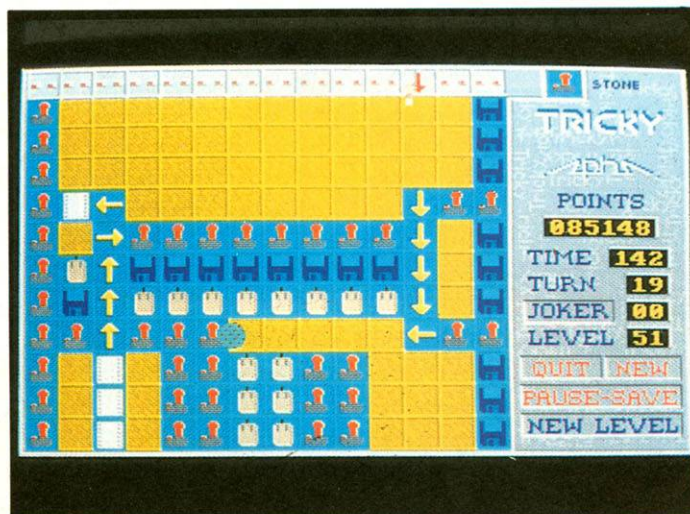


Bild 4. Wie räumt die Kugel am meisten ab, ist die Frage bei »Tricky«

Dabei ist die populärste Disziplin die Bodenverteidigung, die aber auch gleichzeitig eine der gefährlichsten ist.

Sie haben sich als Flieger gemeldet, um Ruhm, Ehre und Geld zu erlangen. Zunächst können Sie sich einen von fünf Jagdfliegern aussuchen, mit dem Sie dann alle Runden bestreiten müssen. Die Arena, in der die Disziplin ausgetragen wird, wechselt alle zehn Runden. Wer fünfzig Runden überlebt, hat es geschafft und erhält die Auszeichnung "Raumverteidiger".

In den Arenen gibt es jeweils eine wechselnde Anzahl von Installationen und Drohnen-Generatoren. Ihre Aufgabe ist es, die Installationen und sich selbst gegen Drohnen zu verteidigen, die von den Schiedsrichtern aus den Generatoren gelassen werden. Sind alle Generatoren vernichtet, brauchen Sie nur noch die restlichen verbleibenden Drohnen abzuschießen, und die Runde ist geschafft. Jetzt wird Ihre Vorstellung durch das Schiedsgericht bewertet. Die Geschwindigkeit, der Kurvenradius und die Beschleunigung Ihres Fighters werden verbessert, wenn Sie eine hohe Trefferquote hatten, und verschlechtert, falls Installationen durch die Drohnen vernichtet wurden. Weiterhin ist Ihre Art zu kämpfen wichtig, da sie für Einschaltquoten sorgt. Sollten sie unter ein Minimum an Zuschauern sinken, wird Ihr Vertrag mit der Fernsehgesellschaft gekündigt.

Im Spiel selbst wird Ihr Fighter von einer hinter Ihnen herfliegenden Kamera in 3-D-Vektorgrafik gezeigt. Ihre Installationen sind rote Pyramiden, die Generatoren blaurote Pyramidenstümpfe. Außerdem durchziehen noch Tracks die Arena, die ein einfaches und schnelles Vorankommen ermöglichen, indem man einfach von oben in sie hineintaucht. Zur Orientierung stehen zum einem eine Karte zur Verfügung und zum anderen ein Kompaß, der in Form eines spitzwinkligen Dreiecks als Headup-Display eingeblendet wird. Manchmal ist die Richtung Norden allerdings nur sehr schwer zu erkennen. Von den vielen verschiedenen Arten von Drohnen können nur die Saboteure den Installatio-



Verteidigen Sie Bodenstationen gegen Drohnen

Thunderstrike

In rund zweihundert Jahren wird die Erde von Fernseh-Netzwerken und der Rüstungsindustrie beherrscht. Der jährliche Höhepunkt sind die Raumverteidigungsspiele.

nen gefährlich werden. Sie werden auf Ihrem Radar in Blau dargestellt. In der Arena sehen Sie sie als blaugrüne Drehkreisel, die sich nur langsam bewegen und nicht schießen können. Hat ein Saboteur eine Einrichtung erreicht, erklimmt er sie, setzt sich auf deren Spitze, verfärbt sich rot und beginnt, sie zu zerstören. Zehn Sekunden kann er jetzt noch abgeschossen werden, bevor er sein Werk vollendet hat und

sich entweder in einen Saboteurmutanten verwandelt, der versucht mit Ihrem Fighter zusammenzustoßen, oder die nächste Installation anpeilt. Die anderen Drohnen haben es meist auf Ihren Fighter abgesehen. Sie nennen sich Kämpfer, Gleiter, Stürmer und Bomber. Kämpfer und Gleiter schießen aktiv auf Sie, wenn auch der Gleiter nur schwach ist. Sind alle Generatoren zerstört, begehen Gleiter und

Stürmer Kamikaze. Stürmer können selbst nicht schießen, zerplatzen aber wie ein Schrapnell in alle Richtungen, wenn sie zerstört werden. Die Bomber fliegen langsam, hinterlassen aber einen Luftminenteppich, der es äußerst schwer macht, ihnen zu folgen.

Werden Drohnen abgeschossen, hinterlassen sie manchmal einen Waffenroboter, der aus der Flughöhe der Drohne zu Boden fällt und auf- und abspringt. Solange er hüpft, kann er eingesammelt werden und verhilft dem Fighter zu einem kleinen Extra. Hier gibt es Verteidigungsroboter, die Ihr Gefährt umkreisen, Roboter, die dem Fighter einen Doppelschuß verschaffen, Hochleistungsroboter, die die Schußstärke erhöhen, sowie Schutzschild- und Turboroboter. Die aktivierten Roboter werden auf dem HUD angezeigt, ebenso, falls nötig, die Zahl der noch verbleibenden Einsatzmöglichkeiten.

»Thunderstrike« ist ein gelungenes 3-D-Vektor Ballerspiel, das den Spieler überzeugen und für lange Zeit an den Monitor fesseln kann. Das HUD-System ist durchdacht genau wie das gesamte Game-Play. Einzig beim Sound merkt man, daß das Spiel auf dem PC entwickelt wurde. Freunden von Spielen dieser Art kann »Thunderstrike« nur wärmstens empfohlen werden.

(Robert Marz/mm)



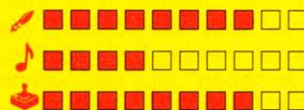
Die Karte kann dabei sehr hilfreich sein

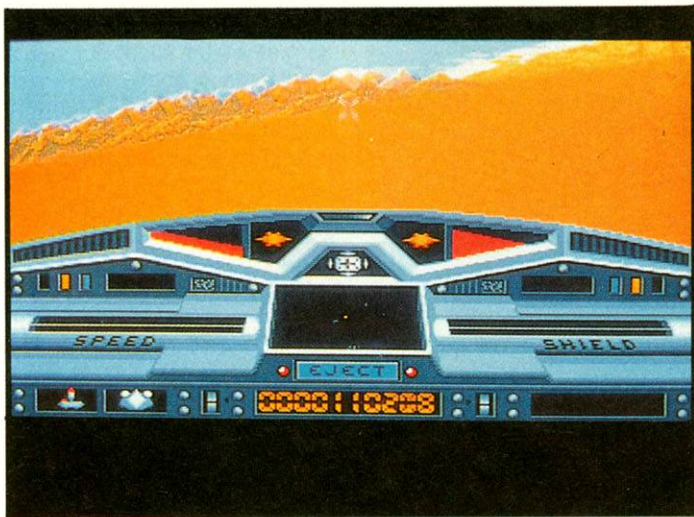
AMIGA DOS

Blitz ★ licht

Name: Thunderstrike
Hersteller: Millenium
Quelle: Fachhandel
Preis: 84,95 DM

GUT





Bei den Kraterspielen kämpfen Sie gegen die Langeweile

Moonblaster

Die Kraterspiele finden alle drei Jahre statt. Der Gewinner darf wichtige Monde bis zu den nächsten Spielen ausbeuten. Es ist an Ihnen, den Sieg für die Royal Airforce zu erlangen.

Die Spiele werden auf drei Monden ausgetragen: dem Eismond, dem Waldmond und dem Wüstenmond. Ihre Aufgabe ist es, gegen Wellen von feindlichen Angreifern innerhalb des Kraters zu bestehen. Solange es noch Gegner für Sie gibt, ist der Krater von einem Energiefeld umgeben, so daß der einzige Weg, ihn zu verlassen, über die Rettungskapsel führt.

Die Gegner fahren ständig auf Sie zu und versuchen, Ihnen in den Rücken zu gelangen, um Sie abschießen zu können. Das müssen Sie natürlich verhindern und sie Ihrerseits abschießen.

Ab und zu gibt es einen Bonusgegenstand, den Sie finden, wenn Sie den Pfeilen auf ihrem Armaturenbrett folgen. Als Extras gibt es bessere Waffen, Unsichtbarkeit, einen engeren Kurvenradius, eine Aufbesserung des Schildes und vieles mehr.

Die Steuerung erfolgt entweder über Joystick oder über Maus, wobei der Joystick, wie bei den meisten Ballerspielen, vorzuziehen ist. Die Grafik selbst ist ein wenig mager, obwohl die Kraterwände wach-

sen, wenn man auf sie zufährt. Die Monde unterscheiden sich eigentlich nur in ihrer Farbe und den Gegnern, mit denen man es zu tun hat. Auch der Sound läßt zu wünschen übrig. Alles in allem kann man auf dieses Spiel ruhig verzichten, da sehr schnell Langeweile aufkommt. Eigentlich sehr schade, denn aus der Idee und der technischen Umsetzung hätte sich sichlich mehr machen lassen.

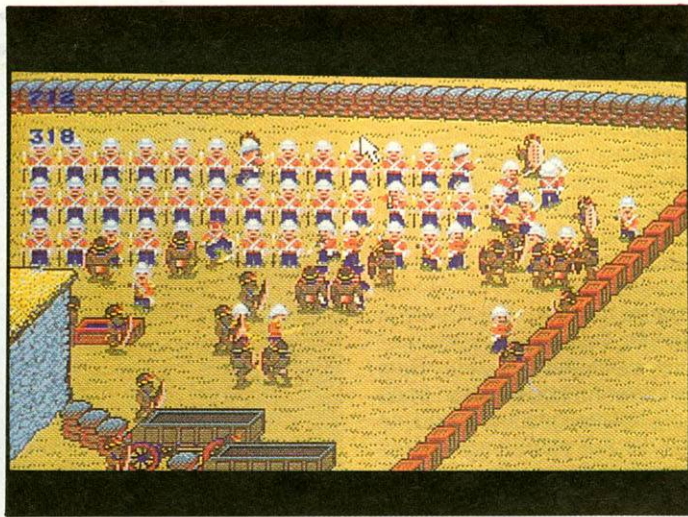
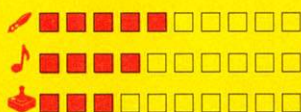
(Robert Marz/mm)

AMIGA DOS

Blitz ★ licht

Name: Moonblaster
Hersteller: Loriciel
Quelle: Fachhandel
Preis: (lag bei Redaktions-schluß noch nicht vor)

GEHT SO



Britten gegen Zulus – die Schlacht am Rorke's Drift

Rorke's Drift

Als die Briten nach Südafrika kamen, hielten sie die Zulus für eine Horde von Wilden. Eine fatale Fehleinschätzung...

Die Zulus waren eine wohlorganisierte Nation, deren militärische Fähigkeiten, auch wenn sie das Schießpulver nicht kannten, die Briten in echte Schwierigkeiten brachten. Im Jahre 1879 fand bei Rorke's Drift eine legendäre Schlacht statt, in deren Verlauf 137 Briten gegen 4000 Zulukrieger antraten, die kurz vorher einen Nachschubkonvoi nebst 1500 britischen Soldaten vernichtet hatten. Die Männer in Rorke's Drift standen also auf verlorenem Posten. Die Schlacht wurde verloren, aber sie regte einige Software-Schaffende zu einer Simulation des Geschehens an. Das Ergebnis liegt nun in Form eines Computerspiels vor, in dem der Spieler jeden einzelnen der 130 britischen Soldaten kommandieren muß. Auch im Computerspiel können die Briten die Zulus nicht bezwingen, es geht also darum, die eigene Haut so teuer wie möglich zu verkaufen.

An sich stellt das Spiel eine nette Idee dar, die im Bereich der Strategiespiele einen neuen Akzent setzt. Leider ist die Umsetzung nicht ganz geglückt. Auch wenn sich in der Spielpraxis keine Fehler bemerkbar machen, fällt die mä-

ßige Ablaufgeschwindigkeit des Spiels unangenehm auf, insbesondere da es 137 Soldaten Mann für Mann zu steuern gilt. Daraus resultiert leider nur ein zögerlicher Spielfluß, der arg an den Spielernerven zehren kann. Als ärgerliche Randerscheinung fiel uns auf, daß die historischen Bemerkungen zu den Ereignissen am Rorke's Drift zwar in Englisch vorhanden sind, aber nicht übersetzt wurden.

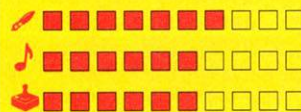
(hs)

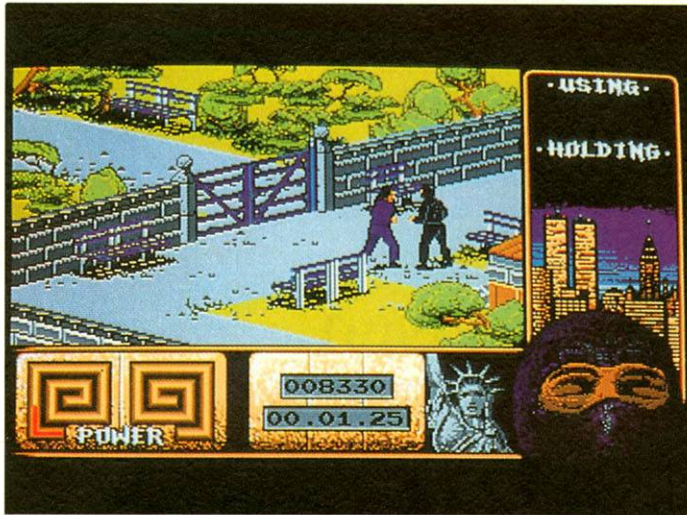
AMIGA DOS

Blitz ★ licht

Name: Rorke's Drift
Hersteller: Impressions
Quelle: Fachhandel
Preis: 79,95 DM

GEHT SO





Der letzte Ninja hat's in unserer Zeit nicht leicht

Last Ninja 2

Der böse Shogun Kunitoki wurde zwar geschlagen, aber nicht vernichtet. Mittels Zauberkraften versetzte er sich in das moderne Manhattan.

Armakuni, der letzte Ninja, wird von den Göttern ebenfalls in diese Zeit versetzt, um Kunitoki endgültig zu schlagen. Etwas verwirrt über diesen unvorbereiteten Ort- und Zeitwechsel findet sich Armakuni im Central-Park von New York wieder.

Noch unbewaffnet macht er sich auf den Weg. Sie steuern Armakuni durch die sechs Szenarien, die in einer dreidimensionalen Darstellung von schräg oben gezeigt werden. Daraus ergibt sich eine gewisse Eingewöhnungszeit an die Steuerung, die auch sonst zu wünschen übrig lässt. Natürlich sind Sie in den Szenarien nicht allein, sondern treffen immer wieder auf Personen, die Sie meistens angreifen. Den Gegnern rücken Sie mit Tritten und Schlägen oder, wenn Sie eine Waffe gefunden haben, mit dieser zu Leibe.

Haben Sie einen Gegner erst einmal endgültig ausgeschaltet, bleibt er für den Rest des Spiels an dieser Stelle auf dem Boden liegen. »Last Ninja« ist eine Mischung aus Action, Adventure und Kampfspiel. Viele Gegenstände müssen gefunden und richtig eingesetzt werden, aber es gibt auch herbe

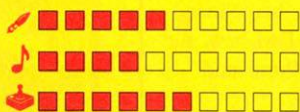
Kämpfe gegen kernige Gegner. Für Fans ist das Spiel eine Enttäuschung. Nach mehr als zwei Jahren sollte man auf dem Amiga eigentlich mehr erwarten, als eine kaum veränderte C64-Grafik und einen müden Sound. Wer das Spiel bereits kennt, sollte sich das Geld besser sparen, zumal nur derjenige in den »Genuß« des Sounds kommt, der über mindestens 1 MByte RAM verfügt.

(Robert Marz/mm)

AMIGA DOS Blitz ★ licht

Name: Last Ninja 2
(Back with a Vengeance)
Hersteller: Systems 3
Quelle: Fachhandel
Preis: 84,95 DM

OK



Zu zweit kann man sich hier heiße Gefechte liefern

Pipe Rider

Ein Ballerspiel, das auf eine Hintergrund-story gänzlich verzichtet. Auch sonst hat »Pipe Rider« eigentlich nicht viel zu bieten.

Das Spiel besticht durch sein einfaches Konzept. Für einen Einzelspieler eher öde, kommt zu zweit richtig Freude auf.

Zwei verschiedene Level-Typen stehen zur Auswahl: Im Pipe-Level bewegen sich die Raumschiffe auf einer um den Bildschirm verlegten Rohrleitung. Aliens, die sich im Inneren dieses Rechtecks befinden, müssen von dort unter Beschuß genommen werden. Im zweiten Leveltyp hält sich jeweils nur ein Raumschiff am unteren Bildschirmrand auf, während die Aliens ständig vom Boden und einer sich herabsenkenden Decke abprallen.

In jedem Level wechselt das Aussehen der Aliens, die immer stärker zurückschießen. Ab und zu hinterlassen sie einen Buchstaben, der eines von sechs Extras symbolisiert. Gelegentlich erscheint ein Pacman, der eine Runde um die Leitung macht, und jedes Raumschiff, das er einholt, frißt. Mit einem Extra, einer Mine, kann man den Gegner dazu bringen, daß er von der Pipe katapultiert wird und sich durch die Aliens zur gegenüberliegenden Seite durchkämpfen muß.

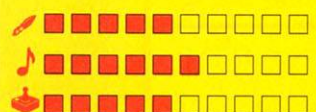
»Pipe Rider« ist sehr einfach gehalten. Spielt man alleine, erhält man ein zweites Raumschiff, das sich spiegelbildlich zum eigenen bewegt. Spielt man hingegen zu zweit, muß man sich ständig entscheiden, ob man sich ein heißes Gefecht mit dem Gegner liefert oder aber zunächst die Aliens abräumt, da hierzu nur knappe Zeit zur Verfügung steht und die mögliche Schußfrequenz gering ist.

(Robert Marz/mm)

AMIGA DOS Blitz ★ licht

Name: Pipe Rider
Hersteller: Magic Soft
Quelle: Fachhandel
Preis: 59,95 DM

OK



AMIGA DOS

Spielboxe

mit traumhaften Preisen!

ZAK McKRACKEN

Der neue urkomische Thriller von Lucasfilm: Können ein Sensationsreporter, Zwei Studentinnen und ein altes Brot die Erde vor Außerirdischen retten? Nicht ohne Ihre Hilfe!

nur DM 69,95*



STARFLIGHT

Erkunden Sie 270 Sternsysteme mit 800 Planeten und sieben außerirdischen Rassen. Einschließlich Ausbildung von Offizieren und Trickfilmgrafik mit Planetenrotation.

nur DM 49,95*

MANIAC MANSION

Lucasfilm präsentiert eine total verrückte Story wie im Kino. Machen Sie sich für die gefährlichste, spannendste und albernste Rettungsaktion Ihres Lebens bereit.

nur DM 69,95*

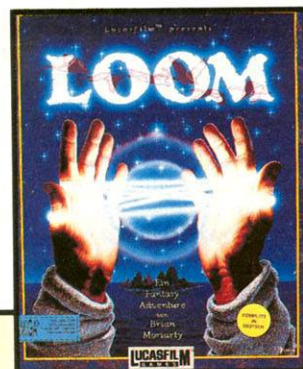


OIL IMPERIUM

Business-Spiel um Geld, Macht und fiese Tricks. Beweisen Sie Ihr Gespür bei den Aktion-Sequenzen, in denen Sie Öl suchen, Pipelines bauen und Feuer löschen.

nur DM 39,95*

**AMIGA
Köln 90**
Wir stellen aus
09.-11. November 1990
Halle 12, Stand 307



LOOM

Das bahnbrechende Fantasy-Spektakel von Lucasfilm mit märchenhaften Landschaften in grafischer Perfektion. Komplett in Deutsch mit deutschem Hörspiel.

nur DM 69,95*

BARD'S TALE II

Das bekannte Fantasy-Abenteuer jetzt 50% größer mit sechs Städten und 25 scrollenden Labyrinthen. Genau das richtige für ausgefuchste Abenteurer.

DM 39,95*



* Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland 4.- DM bzw. für das Ausland 6.- DM Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.

F/A 18 Interceptor

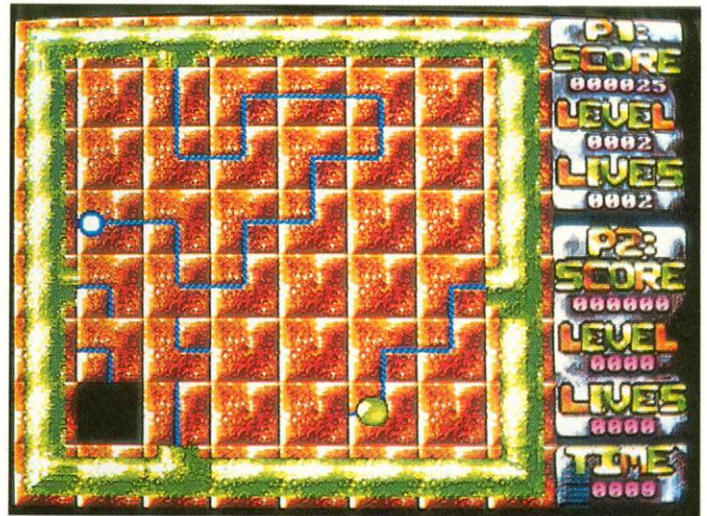
Fliegen Sie das faszinierendste Flugzeug unserer Zeit in sieben Missionen mit unterschiedlichen Schwierigkeitsgraden in realitätsnaher 3D-Grafik.

nur DM 39,95*





Mit Don Johnsons Superboot die Küste unsicher machen...



Da wird der Computerfan zum Klempner

Powerboat USA

Hochgepowerte Motoren, schnittige Boote und trickreiche Kurse sind einige der Zutaten des neuesten Produkts aus Accolades Simulationsküche.

Nach einer grafik- und soundmäßig nicht gerade berauschenden Intro wartet »Powerboat« mit einem recht umfangreichen Menü auf. Sechs Kurse, vier Boote, die auch noch entsprechend ausgestattet werden können, und zwei Schwierigkeitsstufen stehen zur Wahl. Je eine Rennstrecke ist auf dem Mississippi und vor San Francisco angesiedelt, die übrigen sind quasi vor Don Johnsons Haustür in Miami zu finden. Auch eigene Strecken können mit einem sehr komfortablen Editor erstellt und gespeichert werden.

Die zur Wahl stehenden Boote weisen durch ihre sehr unterschiedlichen Bauweisen auch sehr unterschiedliche Eigenschaften auf. Daher wird dem Spieler immer wieder ein etwas anderes Fahrgefühl vermittelt und bei der Steuerung noch mehr Fingerspitzengefühl abverlangt.

Die ausgefüllten Vektorgrafiken sind einfach, aber effizient, nur die Darstellung der Boote ist wirklich etwas dürftig ausgefallen. Das Rennen kann sowohl vom Boot, wie auch aus der Vogelperspektive verfolgt werden. Letzteres ermög-

licht zwar einen besseren Überblick über die Ausflagung, dafür muß aber auf die sehr übersichtlichen Anzeigen im Cockpit verzichtet werden. Hat man sich aber erstmal durch das gute und umfangreiche deutsche Handbuch gearbeitet, die sehr realistisch wirkende Steuerung in den Griff gekriegt und auch die Qualifikationsrunde durchgestanden, dann steht einem heißen Rennen eigentlich nichts mehr im Wege.

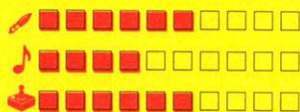
(Antje Hink/mm)

AMIGA DOS

Blitz ★ licht

Name: Powerboat USA
Hersteller: Accolade
Quelle: Fachhandel
Preis: 84,95 DM

OK



Tip Trick

Altes Prinzip auf neuem Medium: Dieses "Denkspiel der neuen Generation" ist nichts anderes als ein Schiebefix für Computer.

Das Programm beginnt mit einem opulenten Eingangsmenü, in dem die Anzahl der Spieler (1 oder 2), Sound, Farben und Schwierigkeitsgrad eingestellt sowie ein Blick auf die Highscore-Liste geworfen werden kann. Man kann frei wählen, in welchem der mehr als 60 Levels man sein Glück zuerst versuchen will.

Dann wird's ernst. Auf dem Bildschirm erscheint das Spielfeld, das in viele kleine Quadrate unterteilt ist. Diese Quadrate enthalten Teile einer Rohrleitung, durch die sich eine Kugel bewegt. Leider ist die Leitung an manchen Stellen defekt und muß durch Verschieben der herumliegenden Einzelteile wieder durchgängig gemacht werden. Die Kugel rollt unterdessen unaufhaltsam auf die Bruchstelle zu. Ist das Leck dann noch nicht mit dem Quadrat repariert, das den richtigen Rohranschluß enthält, fällt die Kugel ins Leere, und eines der Leben ist dahin.

Die Idee hinter diesem Programm ist zwar einfach, aber nicht uninteressant. Leider wirken sich jedoch der auf Dauer nervtötende Sound, die merk-

würdigen Farbkombinationen, die etwas matschige Steuerung und vor allem die langen Warte- und Ladezeiten äußerst negativ auf die Freude am Spiel aus. Auch das Gelächter nach Verlust eines Lebens ist spätestens nach dem dritten Mal überhaupt nicht mehr komisch, sondern nur noch nervend. Da sich das Programm selbst im Zwei-Spieler-Modus nur sehr zäh spielt, geht auch hier die Motivation sehr schnell gen Null. Schade.

(A. Hink/mm)

AMIGA DOS

Blitz ★ licht

Name: Tip Trick
Hersteller: Magic Soft
Quelle: Fachhandel
Preis: 59,95 DM

GEHT SO





Das Image eines Engels steht unserem Charly wirklich gut

All Dogs Go To Heaven

Schon als Film war "All Dogs Go To Heaven" ein Kassenschlager. Hält das Spiel, was der Film verspricht?

Entgegen unseren Vorstellungen, daß es sich hierbei vielleicht um ein Textadventure handle, präsentiert sich das Game als reines Action-Spiel, das sich aus zehn kleinen Arcade-Games zusammensetzt. Die Story beginnt mit einem Gefängnisaustritt. Itchy, ein gerissener Dachshund, befreit seinen zu Unrecht eingesperrten Freund Charly. Von Haß getrieben und auf Rache erpicht, sucht Charly seinen Widersacher, auf dessen Aussage basierend Charly fälschlich beschuldigt und verurteilt wurde. Dabei sollen ihm seine alten Freunde behilflich sein. Nach einer etwas ausschweifenden Wiedersehensparty findet sich Charly in einem Traum als Engel im Himmel wieder. Hier beginnt nun eines der zehn Spiele. Denn nun versucht Charly, von seiner Wolke aus über ihm schwebende Uhren zu erreichen. Mit Ihrem Joystick oder über die Tastatur sollte diese Aufgabe wohl leicht zu bewältigen sein, da lediglich ein wenig Timing verlangt wird. Im Finale entscheidet es sich, ob wieder einmal

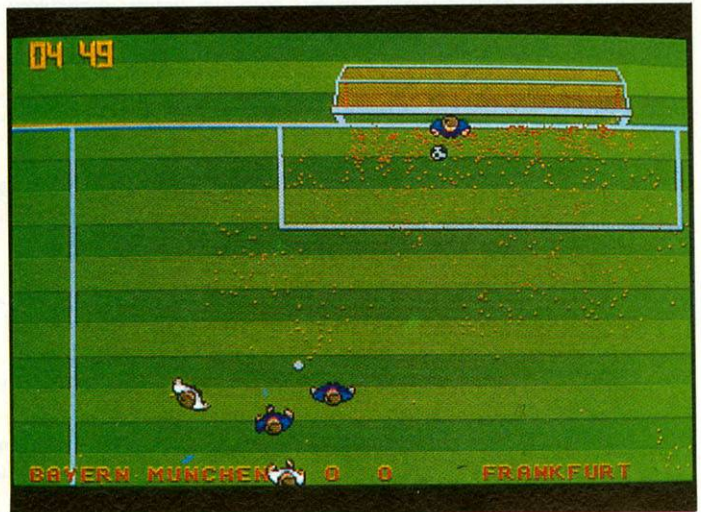
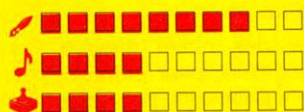
das Gute den Sieg davontragen wird. Charly und der als Entführer entpuppte Carface tragen einen Kampf um die von beiden geliebte Ann Marie aus. Insgesamt ist das Spiel vom Schwierigkeitsgrad ziemlich einfach gehalten. Vielmehr entzückt das Game mit seiner ausgesprochen gelungenen Grafik, wie man sie auch von »Dragons Lair« her kennt.

(Jürgen Seibel/mm)

AMIGA DOS Blitz ★ licht

Name: All Dogs Go To Heaven
Hersteller: Polarware
Quelle: Fachhandel
Preis: 84,95 DM

OK



Ein Hot Shot in den Ofen!

Litti's Hot Shot

Daß es bei der Fülle von Fußballspielen auf dem Markt nicht nur gute geben kann, ist ganz natürlich.

Immmerhin ist »Litti's Hot Shot« ein Low-Cost-Spiel, so daß man nicht so genau schauen sollte. Trotzdem gehört »Hot Shot« zu den schlechtesten Fußballspielen, die ich bislang gesehen habe. Nachdem man seine Trikotfarben, die Mannschaft und die Liga, in der man spielen will, ausgesucht hat, geht es ab ins Spiel. Mit dem Satz "Mannschaft xx Seitenwahl gewinnen" geht's dann los. Das Spielfeld, das etwas zu klein geraten ist, wird aus der Vogelperspektive gezeigt.

Es wird kräftig und ruckelig gescrollt. Immer, wenn irgend etwas passiert, sei es ein Foul, ein Abstoß oder Einwurf, ertönt das Trillern einer Pfeife, und es erscheint das Bild eines gelangweilten Schiedsrichters, der seine Entscheidung in einer Sprechblase verkündet. Selbst zu zweit gegeneinander will keine rechte Freude aufkommen, da das Spiel einfach zu schwerfällig ist. Harte Kämpfe um den Ball sind beinahe unmöglich. Nach der Halbzeit werden die Seiten gewechselt und das Spielfeld um einhundertachtzig Grad gedreht, so daß man nach der Halbzeit in die gleiche Richtung wie vorher spielt, was eigentlich nicht Sinn des Seitenwechsels ist. Immerhin, nach

einem Foul werden gelbe und rote Karten vergeben, und es gibt sogar Elfmeter.

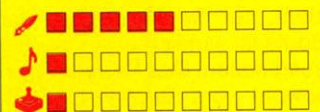
»Litti's Hot Shot« ist wirklich Fußball zum Abgewöhnen. Da die Spieler anonym bleiben und auch sonst keine taktischen Optionen, wie das Zusammenstellen einer Mannschaft oder gezieltes Einwechseln vorhanden sind, kann man sich weder mit dem Team, noch mit den Spielern richtig identifizieren. Selbst den geringen Preis ist das Spiel nicht wert; es sollte seinen Platz wohl eher in der PD finden.

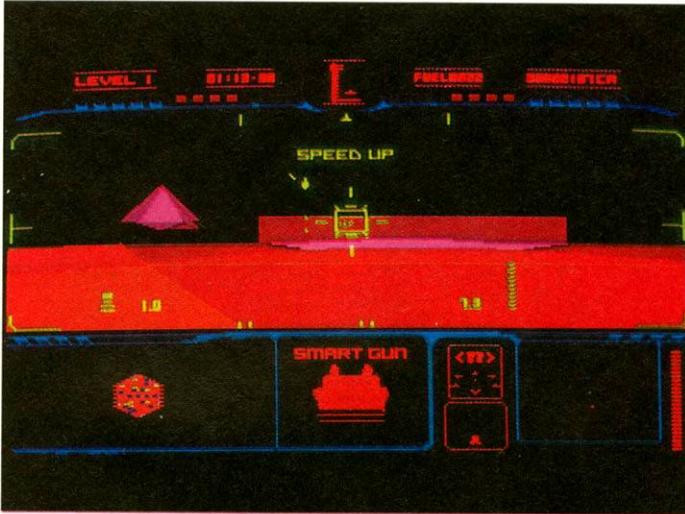
(Robert Marz/mm)

AMIGA DOS Blitz ★ licht

Name: Litti's Hot Shot
Hersteller: Gremlin
Quelle: Fachhandel
Preis: 34,95 DM

SCHLECHT





Extras sammeln, fleißig ballern und entkommen



Hier kann der Zocker seine Sucht so richtig ausleben

Matrix Marauders

Eine Art Rennspiel gegen 17 Gegner. Der Gewinner ist der Annihilator One. Das Ziel: Überleben.

Nachdem Sie sämtliche Vorentscheidungen auf der Erde gewonnen haben, treten Sie gegen die Gewinner der anderen Planeten an. Jetzt wird es ernst. Worum es geht? Sie fliegen mit Ihrem Fighter auf der Suche nach den nächsten Reparaturboxen durch ein Labyrinth von Wegen und Tunneln, sammeln Extras auf und bekämpfen die Gegner. In der Box erhalten Sie für die Abschlußprämien Waffen und, falls nötig, Reparaturen. So gut das Spiel auch losgeht, mit fetziger Hardrock-Musik und guter Grafik in der Box, es läßt entschieden nach, wenn es dann in die Arena geht. Hier wird auf 3D-Vektorgrafik umgeschaltet. Die Wege sind Linien, an den Kreuzungen kann man sich für eine neue Richtung entscheiden. Das Spiel ist sehr schnell, aber auch verwirrend. Die Kanone trifft nicht immer, und es fehlt eine Anzeige, die über die noch einsteckbaren Treffer informiert. Ist man eine Zeit auf der Oberfläche gefahren, sollte man sich zur Mitte begeben und dort über eine Barriere springen, worauf man in die Tunnel kommt. Ein Computer weist den Weg zur Box, indem er sagt, in welche Richtungen man abbiegen soll.

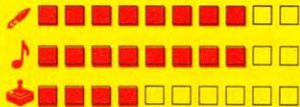
»Matrix Marauders« ist durchaus spielbar, nachdem man sich einige Zeit damit beschäftigt hat. Durch das etwas verwirrende Spielkonzept wird einem das aber nicht leicht gemacht. Es besteht eine Option zum Zusammenlinken von zwei Rechnern, wodurch der Spielspaß natürlich extrem gesteigert wird, wenn man gegen seinen Freund antreten kann. »Matrix Marauders« sollten sich vor allem die Freunde von actiongeladenen 3D-Rennspielen einmal anschauen.

(Robert Marz/mm)

AMIGA DOS Blitz ★ licht

Name: Matrix Marauders
Hersteller: Psyclapse
Quelle: Fachhandel
Preis: 64,95 DM

OK



Fruitmania

Man müßte mal so richtig sein Geld zum Fenster herausschmeißen können, ohne gleich Konkurs anmelden zu müssen.

»Fruitmania«, ein Spielautomat auf dem Amiga, ist mit vier verschiedenen Walzen ausgestattet, auf denen einzelne Früchtesymbole und Joker zu erkennen sind. Das Ziel ist es, vier gleiche Symbole auf den einzelnen Walzen zu erhalten. Natürlich gibt es auch kleinere Gewinne. Zu Beginn des Spiels erhalten Sie einen bestimmten Grundbetrag, mit dem Sie dann Ihr Glück versuchen dürfen. Falls bei einem Spiel zwei von vier Walzen mit demselben Symbol versehen sind, ist es Ihnen erlaubt, die Walzen Ihrer Wahl zu stoppen. Die übrigen Walzen laufen weiter und offerieren Ihnen eine neue Chance zum Gewinn. Zu der einfachen Gewinnübernahme bietet sich dem Spieler eine Alternative. Der eingefleischte Spieler wird nicht den Gewinn übernehmen, sondern sich einer Risikotaste bedienen. Wie beim Roulette dreht sich hier ein Glücksrad mit einem Kugelchen. Der Gewinn kann dabei sogar um das Zwölfwache erhöht werden. Dieses Glücksrad dreht sich so lange, bis Sie Ihren Gewinn verspielt haben oder bis Sie die gesammelten Punkte übernehmen wollen. Wie bei vielen Glücksspielen darf auch hier

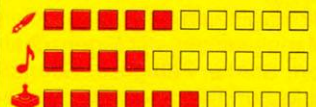
der Joker nicht fehlen. Und wie so oft ersetzt er alle Symbole, so daß zum Beispiel drei Zitronen und ein Joker zum Gewinn führen. Zusätzlich wird das Erscheinen dieser Extrabilder separat gezählt, so daß der Spieler bei einem Gewinn auch die Punkte der Joker übernehmen kann. Grafisch ist gegen die Simulation nichts einzuwenden. Fruitmania mangelt es lediglich am Sound.

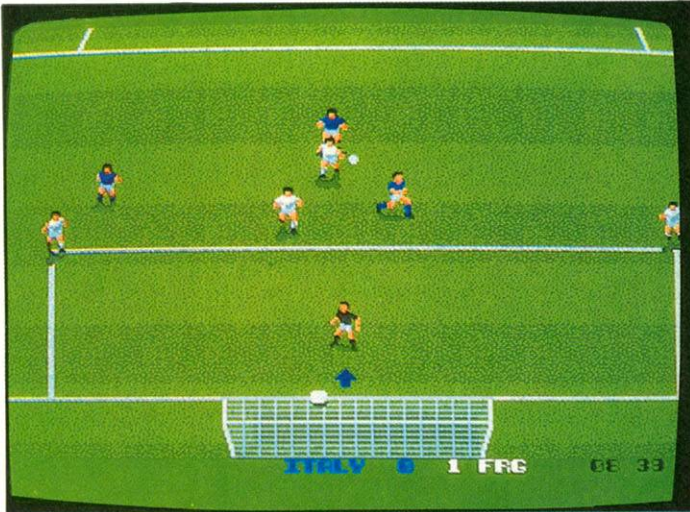
(Jürgen Seibel/jb)

AMIGA DOS Blitz ★ licht

Name: Fruitmania
Hersteller: BSS
Quelle: Alsdorfer PD-Center
Preis: 34,95 DM

OK





Machen Sie Ihr eigenes Team zum Weltmeister

Italy 90

Als Spieler bei der WM dabei sein! Ein Traum. »Italy 90« macht's möglich.

In Italien wurden zur Zeit der Weltmeisterschaft heiße Gefechte ausgetragen, die wohl jeder Fußballfan auf dem Bildschirm mitverfolgt hat. Jetzt ist dem computerisierten Fan die Möglichkeit gegeben worden, das Finale in der originalen Besetzung oder aber sein persönliches Wunschfinale zu spielen. Den Unermüdlichen ist sogar die Möglichkeit gegeben, das gesamte Turnier nachzuspielen. Das Spiel selbst macht grafisch und vom Gameplay einen guten Eindruck. Das Spielfeld wird von schräg oben gezeigt, die Größe des Feldes im Verhältnis zur Spielergröße stimmt auch, was bedeutet, das nicht nur vertikal sondern auch horizontal gescrollt wird. Hier fehlt eindeutig ein "Radar", wie zum Beispiel bei »Kick Off 2«, um zu wissen, wohin man passen kann. Kommt es zu Ecken oder Abstoßen, wird eine kleine Zeichentrickszene gezeigt, bevor es mit dem Spiel normal weitergeht.

»Italy 90« ist ein Fußballspiel, das eindeutig in die Oberklasse gehört. Viele Details, wie die originalen Spieler aus allen Mannschaften, mit Eigenschaften, Stärke, Fähigkeit, Schnelligkeit und Aggressivität, die wenigstens ungefähr im Verhältnis der Wirklichkeit entspre-

chen, belegen dies deutlich. Aber es gibt auch kleine Mängel und es fehlt ein gewisses Maß an Vielfältigkeit. Trotzdem ist »Italy 90« eine Fußballsimulation, die man sich getrost einmal anschauen kann, wenn der Schwierigkeitsgrad der Computergegner auch nicht sehr hoch ist.

Ein Lob auch der Steuerung, die recht gut gelungen ist. Denn so manche Fußball-Simulation ist von der Idee sicher gut, von der Umsetzung jedoch problematisch.

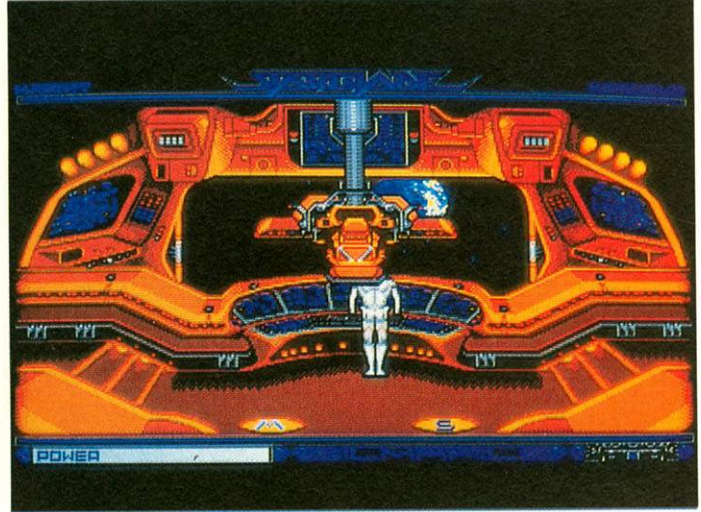
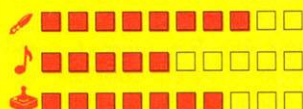
(Robert Marz/mm)

AMIGA DOS

Blitz ★ licht

Name: Italy 1990
Hersteller: U.S. Gold
Quelle: Fachhandel
Preis: 64,95 DM

GUT



Storm Walker hat eine gefährliche Mission

Starblade

Im Jahr 3001 ist die Menschheit, bis auf ein paar Kolonien, von Insektoiden ausgerottet worden. Storm Walker soll die Königin der Insekten vernichten.

Dazu müssen Sie, als Storm Walker, aber zunächst deren Position ausfindig machen. Diese ist auf fünf Disketten über die gesamte Galaxie verstreut. Ihnen wird die Starblade, das modernste Raumschiff zur Verfügung gestellt, um diesen Auftrag zu erfüllen. Sie besteigen also das Schiff und fliegen damit zu einem der Planeten, auf dem Sie mit der Attla, Ihrem Landegleiter, landen und nach der ersten Diskette suchen.

Wer jetzt denkt, »Starblade« sei ein simples Ballerspiel, hat weit gefehlt. Dem Spieler steht eine Vielzahl an Möglichkeiten offen, er muß sich zwar manchmal mit Granaten und Laserschwert den Weg freikämpfen, aber die Suche nach den Disketten, das Beschaffen von Ersatzteilen für die Starblade und das Handeltreiben stehen doch im Vordergrund.

Im eigenen Schiff kann sich Storm Walker ebenfalls frei bewegen, Schubladen und Laderäume öffnen, das Schiff reparieren und vieles mehr. Will er zu einem anderen Planeten, kann er zwischen zwei Arten zu reisen wählen: Hyper- und Normalflug. Im Normalflug kann die Starblade angegriffen

werden, und Sie müssen sie mit den Bordgeschützen verteidigen.

»Starblade« ist ein Actionadventure, bei dem sowohl Grafik als auch Sound nicht zu kurz kommen. Die gesamte Steuerung erfolgt über Joystick und ist leicht erlernbar. Da das gesamte Spiel und die Anleitung in Deutsch gehalten wurden, gibt es eigentlich kaum Negatives zu sagen.

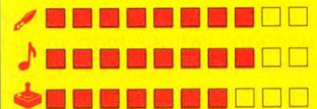
(Robert Marz/mm)

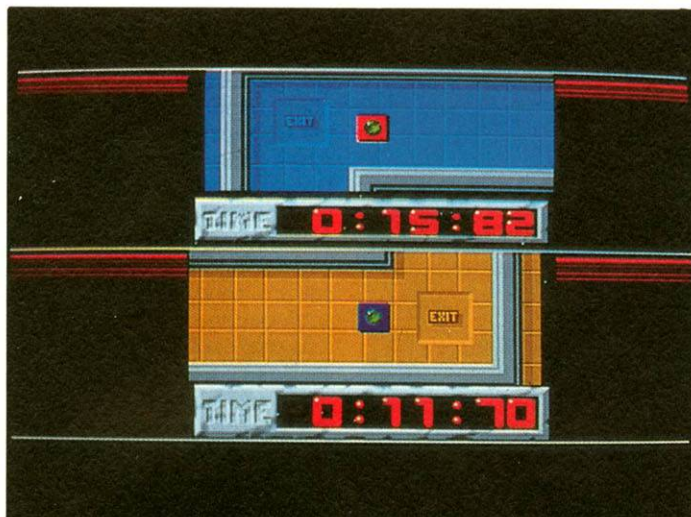
AMIGA DOS

Blitz ★ licht

Name: Starblade
Hersteller: Palace Software
Quelle: Fachhandel
Preis: 84,95 DM

GUT





Nicht überall, wo "Exit" draufsteht, ist auch einer dahinter

Labyrinth

Um in dem Gewirr von Korridoren und falschen Ausgängen den richtigen Weg ins nächste Level zu finden, sind Glück und ein gutes Gedächtnis erforderlich.

Die Story des Spiels ist kurz, etwas seltsam und schnell wieder vergessen: Zwei miteinander befreundete Kugeln verkrachen sich und wollen sich duellieren, was man als Kugel schlecht mit einer Pistole machen kann. Stattdessen rennen die beiden durch die Gänge der Xanden (was auch immer das ist), um möglichst viele Punkte zu sammeln. Besagte Punkte sind nur leider nirgendwo abzulesen.

In den Gängen selbst wimmelt es von Fragezeichen, unter denen sich Beschleuniger, Extraleben oder Extrazeit verbergen können. Genauso gut jedoch kann die Kugel wieder an den Anfang des Levels zurückversetzt werden, ein Leben verlieren oder aber einen versteckten Ausgang aufdecken. Der Inhalt der Fragezeichen ist in jedem Spiel gleich, so daß es mit einem bißchen Gedächtnistraining gar nicht so schwer ist, die 50 Levels trotz oft sehr knapper Zeitvorgaben ohne größere Probleme zu bewältigen. Spannend wird das Spiel erst, wenn man gegen einen zweiten Spieler antritt, denn nun ist auch strategisches Denken gefragt.

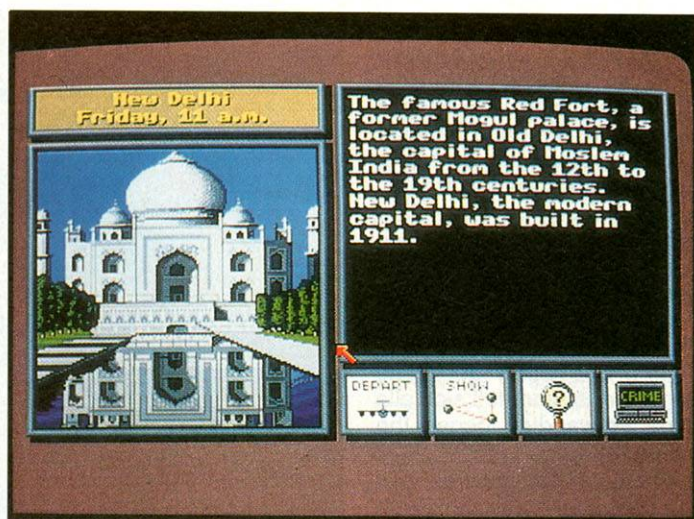
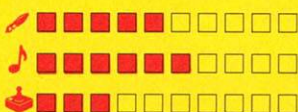
Der gesplittete Bildschirm, dessen untere Hälfte beim Einzelspiel leider dunkel bleibt, scrollt sanft in alle Richtungen und auch die Steuerung ist erfreulich exakt. Auch an Sound und Grafik ist nichts auszusetzen. Nur die Motivation läßt nach kurzer Zeit merklich nach, da man die Levels und ihren Inhalt auswendig kennt.

(A. Hink/mm)

AMIGA DOS
Blitz ★ licht

Name: Labyrinth
Hersteller: Magic Soft
Quelle: Fachhandel
Preis: 59,95 DM

GEHT SO



Auf der Suche nach Verbrechern

Wo in aller Welt ist Carmen Sandiego?

Ständig werden die tollsten Verbrechen begangen! Helfen sie Interpol bei der Aufklärung.

Die Bande von Carmen Sandiego stiehlt ständig wertvolle Gegenstände aus allen Ländern der Welt. Sie als Spezialagent bekommen den Auftrag, den Täter innerhalb einer bestimmten Zeit zu fangen. Dazu können Sie sich in einer Großstadt des Landes, in dem Sie sich befinden, an verschiedenen Örtlichkeiten umhören. Hier erfahren Sie, wenn Sie an der richtigen Stelle sind, etwas über den nächsten Ort, wo der Dieb sich aufhält und etwas über seine Person. Diese Daten können Sie dann in den Interpol-Computer eingeben, der dann aus einer Liste von Verbrechern die falschen aussondert. Haben Sie nur noch einen Verdächtigen übrig, können sie ihn verhaften lassen und an den nächsten Fall gehen. »Carmen Sandiego« gibt es schon länger und in verschiedenen Teilen. Ziel des Spiels ist es eigentlich, das Wissen über die verschiedenen Länder der Erde zu erweitern, indem man aus den Aussagen

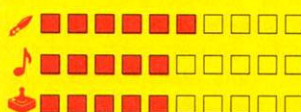
verschiedener Leute schließt, wo der Gangster sich aufhält. Ein Problem an dem Spiel ist, daß die Anleitung zwar in Deutsch, das Spiel und der Almanach aber in Englisch sind, so daß es fundierte Englischkenntnisse voraussetzt.

(Robert Marz/mm)

AMIGA DOS
Blitz ★ licht

Name: Wo in aller Welt ist Carmen Sandiego?
Hersteller: Broderbund
Quelle: Fachhandel
Preis: lag nicht vor

OK

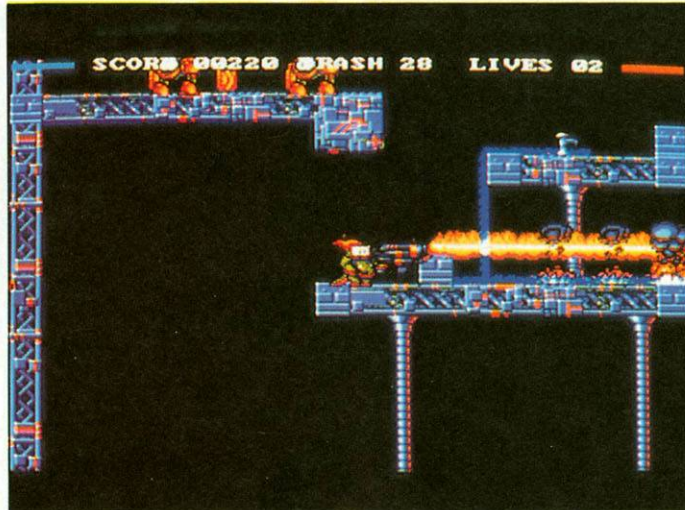


Das ist aber noch nicht alles. Quiffy ernährt sich nämlich von dem Müll, den die alte Zivilisation zurückgelassen hat. Da dieser reichlich vorhanden ist, hat er auch schon ein kleines Bäuchlein bekommen. Auf seiner Flucht kann er deshalb einen Raum nur dann mit dem Endteleporter verlassen, wenn er alles eingesammelt hat, was an Müll herumliegt. Hinzu kommt, daß Quiffy der letzte seiner Art ist und daß sich alle seine Verwandten gegen ihn stellen.

Quiffy macht sich also auf seinen Weg, der voller Gefahren steckt. Dabei durchquert er Höhlen, die von der Seite gezeichnet werden. Da Quiffy ein Blobble ist, kann er sehr gut klettern. Senkrechte Wände, sofern sie nicht vereist sind, und sogar die Decke stellen für seine Hände und Füße kein Problem dar. In den Höhlen begegnet er insgesamt elf verschiedenen Arten von Gegnern, die er mit gefundenen Waffen bekämpfen kann. An Waffen gibt es Bumerangs, Zeitbomben, deren Zündzeit sehr kurz ist, Handgranaten, einen Flammenwerfer und Wurfsterne.

Der Flammenwerfer ist eine recht wirkungsvolle Waffe, hat aber den Nachteil, daß er sich manchmal, statt zu schießen, in eine gackernde Gans verwandelt, was natürlich äußerst peinlich werden kann.

Damit ein wenig Spannung ins Spiel kommt, gibt es praktisch zwei Zeitlimits. Da wäre zum einen das Wasser. Es läuft aus Hähnen, die nicht abgestellt werden können und füllt die Levels langsam, aber sicher. Sammelt Quiffy einen Wassertropfen ein, wird die Fließgeschwindigkeit erhöht, ein Kolben verringert sie. Da er unter Wasser nicht atmen kann und Fett nach oben treibt, hat er leichte Probleme beim Tauchen, da er immer auf seine Luft achten muß und nur schlecht vorankommt. Die Waffen – sogar der Flammenwerfer – funktionieren auch unter Wasser. Das zweite Zeitlimit besteht aus dem Geist seiner knutschenden Tante, der sich nach einiger Zeit auf Quiffis Spuren setzt und sich von nichts ablenken läßt, ihm genau auf seinem Weg zu folgen, nur etwas schneller als er. Berührt er ihn, saugt er ihm mit



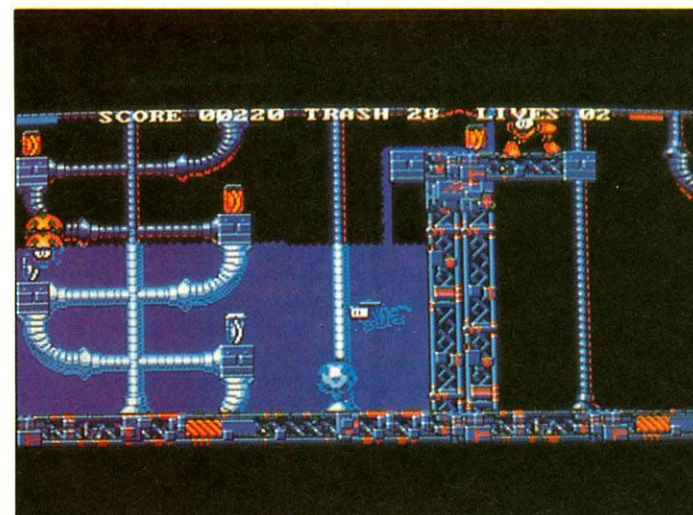
Quiffy in Aktion. Mit Taucherbrille und Waffen...



Quiffys unterirdisches Reich wird bedroht. Die Zapfen des Zorns erwachen aus ihrem Schlaf und lassen die Höhlen mit Wasser volllaufen. Es bleibt ihm also nur die Flucht.

seinen Küssen sehr schnell seine Lebensenergie aus. Da er ein Geist ist, läßt er sich auch nicht von Waffen oder sich bewegenden Wänden beeindrucken. In den Höhlen gibt es außer den Level-Endteleportern auch noch andere, die Quiffy fleißig durch die Gegend transportieren. Neben denen, die man sehen kann, gibt es auch unsichtbare Wesen, die natürlich etwas gemeiner sind. Aber ab und zu führen sie in eine Bonus-Höh-

le, in der man sich vor Extras und Waffen kaum retten kann. Es gibt zu fast jedem Level ein Paßwort, das meistens in einer dieser unsichtbaren Höhlen zu finden ist. In den Levels gibt es Hebel, die Aktionen auslösen; so kann man mit manchen Wänden bewegen und Passagen freimachen, man kann aber auch reinfallen und eine Dusche von Beady Balls, kleinen, springenden Monstern, oder eine Reihe von Kanonen damit auslösen. Außerdem



...kann eigentlich (fast) nichts passieren

gibt es Ballons, mit denen man fliegen kann, und Fallschirme, die einen sanften Fall ermöglichen. Minen sind unangenehme Gegenstände, verleiten sie einen doch meistens dazu, ins Unheil zu springen und vor allem auch Lebensenergie zu verlieren. Auch sonst gibt es eine Menge zu erleben. Dafür sorgen schon die vielen verschiedenen Monster: Neben den praktisch "normalen" gibt es Doktor Dustys, die mit Dynamit werfen, Psycho Teddys, die so gefräßig sind, daß sie Quiffy mit einem einzigen Biß töten können, aber auch bei der Müllbeseitigung helfen. Das Gegenteil stellen Bulbous Headed Vongs dar, die mit Müll nur so um sich schmeißen. »Flood« ist ein Jump'n'Run-Spiel, das so richtig begeistern kann. Alles paßt zueinander und ist aufeinander abgestimmt: witzige Animationen und lustige Musik, tiefes, bedrohliches Blubbern von Wasser und gute Effekte der Waffen. Die Spielbarkeit und Steuerung sind gut, wenn man sich erst einmal daran gewöhnt hat, daß man auch unter der Decke laufen kann. Auch das Paßwortsystem trägt dazu bei, daß man die 36 normalen und die 6 Zusatzlevels ohne Frust schaffen kann. Negativ ist eigentlich nur, daß zwischen den Levels sehr lange auf Diskette zugegriffen wird. Flood ist Bullfrogs würdiger Nachfolger auf »Populous«, auch wenn es einer ganz anderen Sparte entstammt.

(Robert Marz/ mm)



AMIGA DOS

Blitz ★ licht

Name: Flood
Hersteller: Electronic Arts
Quelle: Fachhandel
Preis: 89,- DM

SUPER



Das ist der Stoff, aus dem klassische Krimis und Sierra-Adventures gemacht werden.

Die Story beginnt eigentlich ganz friedlich im Park einer Universität in den Südstaaten der zwanziger Jahre. Laura Bow, ihres Zeichens Studentin und Heldin dieser Geschichte, wird von ihrer Freundin Lillian auf die Plantage ihres Onkels Henri eingeladen. Dort soll, sehr zur Verblüffung des gesamten Clans, ein Familientreffen stattfinden, das tödlich langweilig zu werden verspricht.

Tödlich wird das Treffen zwar, aber niemals langweilig. Nach einer Bootsfahrt durch die mondbeschiedenen Sümpfe erreichen die beiden Freundinnen den alten heruntergekommenen Familiensitz, der im zukenden Licht der Blitze eher wie ein Spukschloß aussieht. Der Onkel, Colonel Henri Dijon, entpuppt sich als ein griesgrämiger alter Despot, der von seiner hübschen französischen Krankenschwester im Rollstuhl durch die Gegend geschoben wird. Oh la la! Auch der Butler und die übrigen Angestellten wirken nicht gerade sehr vertrauenerweckend. Der Rest der Familie, die bereits zum Dinner Platz genommen hat, als Laura und Lillian ankommen, ist ein buntes Sammelsurium miesester Charaktere. Spieler, Glücksritter, Quacksalber, Schwindler, Flittchen und Trinker: Alles ist vorhanden und alle sind hinter Geld her wie der Teufel hinter der armen Seele. Jeder hat mindestens ein dunkles Geheimnis zu verbergen, und keiner scheut sich, einen anderen hereinzuwerfen. In diese nette Gesellschaft hinein platzt wie eine Bombe die Ankündigung des Colonels, nach seinem Tode sein Vermögen zu gleichen Teilen den dann noch lebenden Anwesenden zu vermachern. Während der nun folgenden Ereignisse reduziert sich die Zahl der Familienmitglieder auf geheimnisvolle Weise am laufenden Band, und Laura als Tochter eines Detektivs fühlt sich genötigt, den seltsamen Vorgängen auf die Spur zu kommen. Leichen, die in der Gegend herumliegen und wenig später spurlos verschwunden sind, stellen dabei noch das kleinste Pro-



Die Sümpfe halten eine reichhaltige Fauna bereit...

The Colonel's Bequest

Zuckende Blitze, Donnergrollen, Voodoo-Zauber, ein altes Herrenhaus voller Geheimgänge auf einer Insel im Sumpf, eine total zerstrittene Familie und ein riesiges Vermögen.

blem dar. »The Colonel's Bequest« ist als Theaterstück in acht Akten geschrieben, die jeweils eine Stunde dauern. Sobald eine Aufgabe erledigt ist, bewegen sich die Zeiger einer eingeblendeten Uhr um jeweils fünfzehn Minuten weiter. Wie in fast allen Sierra-Adventures steuert man auch hier die

Hauptfigur per Tastatur, Joystick oder Maus hübsch animiert durch die mit vielen Details ausgestatteten Räume. Alles kann untersucht, geöffnet oder verschoben werden. Lauras Aufgabe besteht vor allem darin, durch geschickte Fragen die Leute auszufragen, die meistens auch ganz

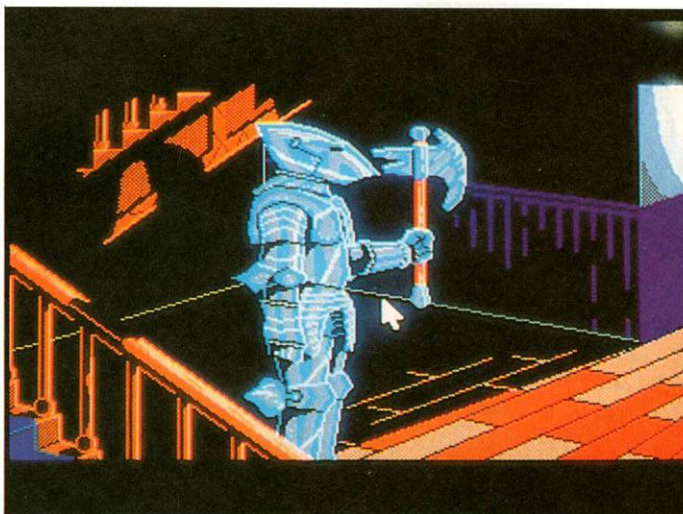
munter drauflos erzählen. Die bunte Story ist mit Anspielungen jeder Art nur so gespickt. Besonders hübsch sind die Hinweise und Gags aus verfilmten Krimis oder Gruselfilmen. Und nirgendwo stirbt man so »schön« wie bei Sierra-Adventures.

Wie bei fast allen Sierra-Spielen ist auch hier häufiges »Saven« der Schlüssel zum Erfolg. Nichts ist ärgerlicher, als eine schwierige Passage erneut spielen zu müssen, nur weil nicht rechtzeitig gesichert wurde.

Wie von Sierra bereits gewohnt, sind Grafik, Hintergrundstory und Atmosphäre auch beim »Erbe des Colonels« eine quasi Eins-zu-eins-Umsetzung des exzellenten PC-Originals. Flüssige Animation und an Details reiche Hintergründe sind eine Freude für die Augen. Beim Sound allerdings ist den Programmierern dieser Version dann wohl die Puste ausgegangen, denn der ist nun wirklich eines Amiga unwürdig.

Inhaltlich aber ist an dem Programm absolut nichts auszusetzen. »The Colonel's Bequest« ist ein äußerst spannender und kniffliger Krimi, an dem auch ein gewiefter Hitchcock-Kenner sich die Zähne ausbeißen kann. Da sich aber die häufigen und langen Ladevorgänge auf den Fluß des Spiels ziemlich negativ auswirken, ist das Programm eigentlich nur Besitzern von Harddisk und Speichererweiterung guten Gewissens zu empfehlen.

(A. Hink/mm)



Alte Häuser sind mit Vorsicht zu genießen

AMIGA DOS

Blitz ★ licht

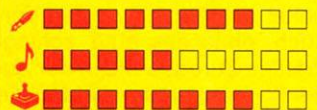
Name: The Colonel's Bequest

Hersteller: Sierra

Quelle: Fachhandel

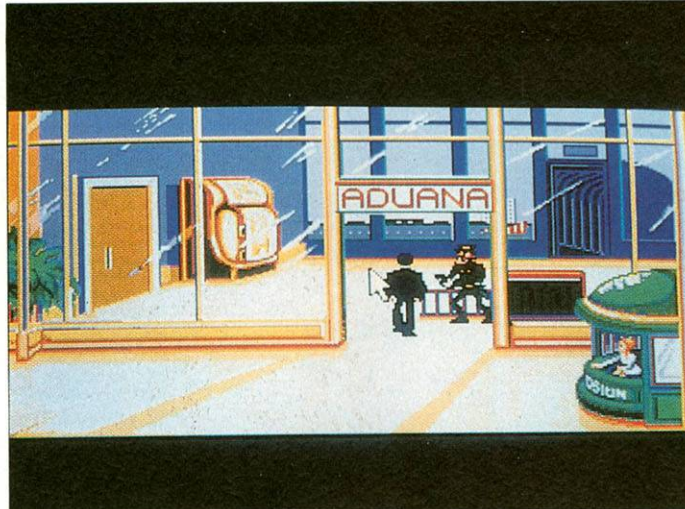
Preis: 129,95 DM

GUT



Die einzige Spur führt nach Santa Paragua, wohin sich John auch zunächst begibt. John ist ein überaus begabter Agent, der sich vor allem für Technik interessiert. Er hat schon viele Fälle auch mit anderen Agenten zusammen gelöst. Da er der Beste ist, wurde er mit dieser Aufgabe betraut. Ausgerüstet mit einem Spezialkofferchen des CIA, der ihm vom Chef persönlich überreicht wurde, macht er sich an die Arbeit.

»Operation Stealth« ist ein weiteres Adventure von Cinematique, ein direkter Nachfolger von »Future Wars, Time Travelers«. Entsprechend ähnelt sich auch der Aufbau der spannenden Adventures. Sogar die Ausgaben des Computers, wenn etwas nicht so funktioniert, wie man das gerne hätte, sind gleichgeblieben. Wie beim Vorgänger wird das gesamte Spiel mit der Maus gesteuert. Es wurden aber einige entscheidende Verbesserungen eingebaut. Drückt man den rechten Mausknopf, erscheint ein durchsichtiges Menü mit den möglichen Optionen, die da wären: »Examine«, »Take«, »Inventory«, »Operate«, »Use« und »Speak«. Wählt man eine Option mit dem linken Mausknopf an, bezieht sie sich auf einen Gegenstand im Raum. Mit der rechten Taste hingegen kann man ein mitgeführtes Objekt manipulieren. John wird bewegt, indem man mit der Maus auf den entsprechenden Ort klickt. Eine wesentliche Verbesserung ist, daß man den Mauszeiger auch auf kleine Objekte nicht mehr so genau plazieren muß, wie beim Vorgänger. Es ist jetzt wesentlich einfacher, zum Beispiel den Münzeinwurf eines Automaten zu bedienen. Mit einem MByte Speicher bekommt der Spieler die Texte in einer wirklich guten Sprachausgabe vorgelesen. Einmal mit dem Flugzeug in Santa Paragua angekommen, geht es dann los. Gleich zu Beginn trifft man auf einen alten Bekannten aus »Future Wars«, nämlich den Zeitungsautomaten. Nur scheint er hier noch zu funktionieren, denn er arbeitet schon beim ersten Versuch, ohne das gute, schlagkräftige Zureden wie in der Zukunft, wenn man die Münze aus dem Rückgabe-Slot einwirft. Da Santa Paragua nicht



John Glames gerät in manche bedrohliche Situation

Operation Stealth

Eine Stealth-Fighter ist von Unbekannten entführt worden. John Glames, Spezialagent des CIA, wird damit beauftragt, die Maschine wiederzubeschaffen.

gut auf die USA zu sprechen ist, sollte man auch nicht mit einem amerikanischen Paß durch die Kontrollen gehen. In der Zeitung steht dann aber glücklicherweise, mit welchen Land Santa Paragua diplomatische Beziehungen aufgenommen hat. Also nichts wie auf die Toilette, damit niemand was merkt, das Geheimfach geöffnet und mittels des Paßfälschergeräts einen Paß von diesem Land herstellen. Damit

kommt man dann auch ohne Probleme durch die Kontrolle. Die Frau am Informationsstand hat eine Nachricht von Johns Kollegen, der hier schon einige Zeit an diesem Fall arbeitet. In dieser steht, daß er sich um das Gepäck eines Mr. Martinez kümmern soll. In diesem Gepäckstück sind dann auch zwei Gegenstände, die Johns Ausrüstung, welche bislang aus dem Koffer und einem Säure verspritzenden Stift be-

steht, um eine Uhr und einen Rasierapparat bereichert. In der Uhr befindet sich ein 30 Meter langes Seil, und der Rasierapparat enthält ein Tonbandgerät. Der Apparat hat dann auch gleich eine weitere Nachricht für John, die er wieder auf der Toilette des Flughafens abhören kann.

Wenn er sich jetzt noch von dem verräterischen fremden Koffer befreit, indem er ihn einfach wegwirft, kann er sich in das Leben von Santa Paragua stürzen. »Operation Stealth« ist ein Adventure besonderer Art, das viele Stunden Knobelspaß verspricht. Viele Verbesserungen, auch im grafischen Bereich, machen das Spiel noch spielbarer als sein Vorgänger. Die Texte sind mit viel Humor und feinen Anspielungen geschrieben, und es gibt eine ganze Menge zu erforschen und viele Abenteuer zu bestehen. Da sich das Spiel jetzt über drei Disketten erstreckt, liegt die Vermutung nahe, daß es wesentlich umfangreicher geworden ist. Hier ist auch ein kleiner Nachteil versteckt. Das Spiel erkennt nämlich lediglich das interne Laufwerk an, auf einem externen kann nur gespeichert werden, so daß man häufig Disketten wechseln muß. »Operation Stealth« ist ein Adventure der Spitzenklasse, das seinen Konkurrenten von Sierra und Lucasfilm-Games durchaus Paroli bieten kann.

(Robert Marz/mm)



List und Tücke sind der Schlüssel zum Erfolg

AMIGA DOS

Blitz ★ licht

Name: Operation Stealth
Hersteller: Delphine
Quelle: Fachhandel
Preis: stand bei Redaktions-schluß noch nicht fest

SUPER





Ballern, was das Zeug hält – »Anarchy«

Anarchy

Und willst du nicht mein Bruder sein, dann schlag ich mit dem Laser drein...

Warum nur lassen uns die Aliens nicht mal zufrieden? Wohl deshalb, weil sonst ein großer Teil unserer Computerspiele vom Markt verschwinden würde. Also, Pseudo-Raumfahrer, frisch ans Werk, die Laserkanone geölt und ab ins Kampfgetümmel. Diesmal geht es um Teile der Ausrüstung, die auf einem entfernten Planeten liegengeblieben sind und nun aufgesammelt werden müssen. Und wie so oft ist da irgendein Störenfried, dem das überhaupt nicht paßt und der als Einspruch den Finger an den Abzug legt. So heißt es also, sich schießend zur Oberfläche durchzuschlagen und das Gesuchte einzuschleusen. Währenddessen tummeln sich diverse Gegner durch die oberen Luftschichten, und rauben einem bei Berührung Energie und Nerv. Nach der Methode "Wie du mich, kann ich dich" wird dann eben geballert, bis der Joystick glüht oder die lustige Weltraumfahrt unlustig endet. Diverse Zusatzwaffen können dabei gewinnbringend eingesetzt werden.

»Anarchy« unterscheidet sich von vielen andern Spielen seines Genres durch horizontales Scrolling. Der eigene "Flieger" ist allerdings manchmal etwas

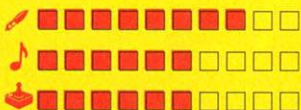
schwer in den Griff zu bekommen, weil er nach einem Richtungswechsel nicht erst stoppt, sondern sofort umkehrt. Fliegt ein Gegner genau dort, wo man selbst hin will, gibt es Energieabzug. Ansonsten ist »Anarchy« ein flottes, grafisch gut gelungenes Spiel, das allerdings von der Thematik her ein Kilo Staub mitschleppt. Wer Ballerspiele sammelt, wird sowieso zugreifen, ansonsten reißt es den mit Spielen dieser Art übersättigten Amiga-Fan nicht vom Hocker.

(jb)

AMIGA DOS Blitz ★ licht

Name: Anarchy
Hersteller: Psygnosis
Quelle: Fachhandel
Preis: 64,95 DM

GUT



Software-Noten und wie sie zustandekommen

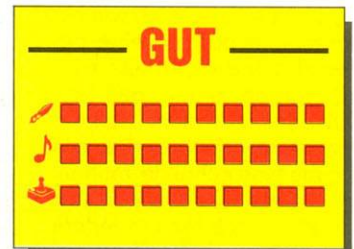
Daß Ihre AMIGA DOS bei den Spiele-Tests ein Urteil vergibt, haben Sie selbst sicherlich schon gesehen. Wie dieses Urteil aufgebaut ist, das soll hier noch einmal im Detail erklärt werden.

AMIGA DOS beurteilt jedes Spiel anhand verschiedener Kriterien: Grafik, Musik und Spielspaß. Die Noten dafür werden in LED-ähnlichen Diagrammen, die am Ende jeder Review zu finden sind, unter den Oberbegriffen Grafik, Sound und Motivation dargestellt. Diese Oberbegriffe werden durch einen Pinsel für Grafik, eine Note für Sound und einen Joystick für Motivation symbolisiert.

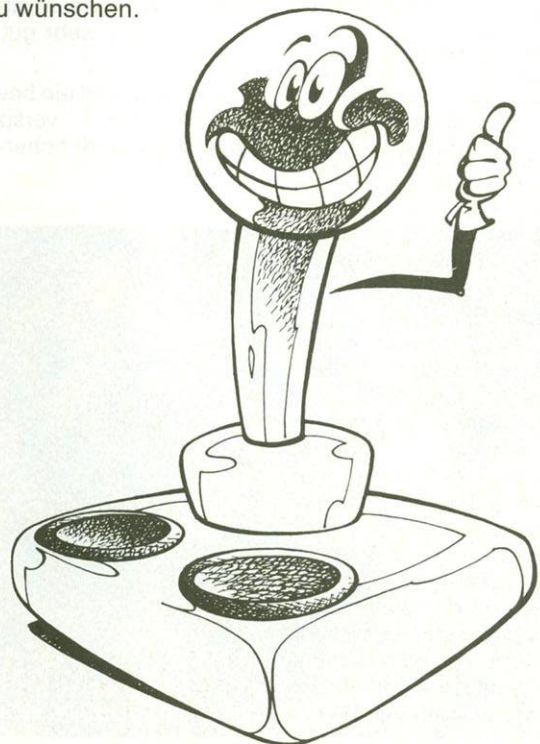
Die Note selbst richtet sich nach der Höhe des wiedergegebenen Wertes. Je mehr Einheiten des Diagramms ausgefüllt sind, desto besser ist die Beurteilung für den entsprechenden Oberbegriff. Zu diesen Noten gesellt sich ferner ein individuelles Urteil des Testers. Eine vierte Note also, die allerdings nicht wiedergegeben wird.

Aus diesen vier Werten bildet sich schließlich das Gesamturteil:

- SUPER
- GUT
- OK
- GEHT SO
- SCHLECHT



Die Bedeutung dieser Bewertungen erklärt sich von selbst. Uns bleibt noch, Ihnen viel Spaß mit dem AMIGA DOS Spielerteil zu wünschen.





»Rotator« präsentiert pixelorientiertes Drehen

Rotator

»Rotator« ist ein Strategiespiel besonderer Art. Hier wird Strategie mit Action verknüpft. Sie sollen eine Stromleitung von links nach rechts über das Spielfeld legen. Ihr Gegner versucht das ebenfalls, aber von oben nach unten. Allerlei Hindernisse sowie die Tatsache, daß die Kabel sich wie ein Gummiband nur auf eine bestimmte Länge dehnen können, sorgen dafür, daß der Spaß bestimmt nicht zu kurz kommt. Bei »Rotator« wurde erstmals eine schnelle Technik zum Drehen von Pixelbildern benutzt, so daß man in jedem Fall gespannt sein darf.

Buck Rogers

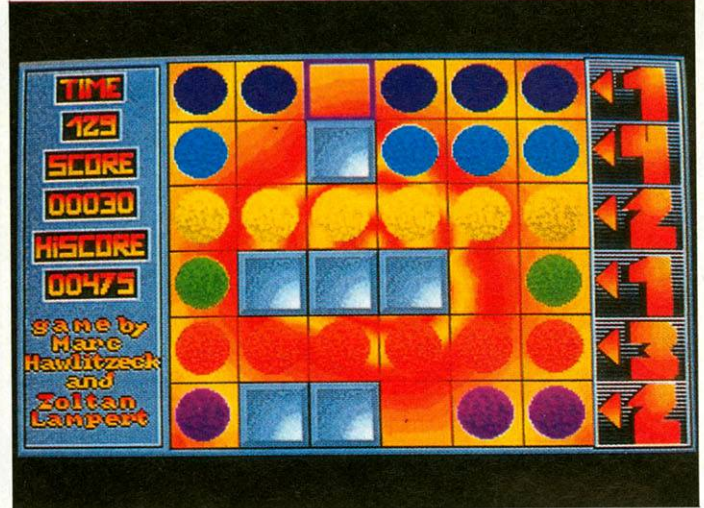
Brandheiß von SSI steht demnächst die Umsetzung des Rollenspiels Buck Rogers ins

Demnächst auf Ihrem Computer

Den Programmierern scheinen die Ideen niemals auszugehen. Neue Spiele aus allen Genres erwarten Sie in nächster Zeit.

Haus. Schon das papiergebundene Rollenspiel konnte bereits begeistern, die Umsetzung des Rollenspielspezialisten ist, wie man unschwer erkennen kann, sehr gut gelungen.

Gute Grafik und ein bewährtes Kampfsystem versprechen Spannung und hohen Spiel-



Bei »Krymini« führen Geschick und Logik zum Erfolg

spaß. Wir sind auf die Abenteuer des Weltraumhelden sehr gespannt.

Krymini

»Krymini« wird auf einem 6x6 Felder großem Spielfeld ausgetragen. Auf dem Feld befinden sich 32 Kugeln, von denen

jede einmal gezogen werden muß. Die einzelnen Kugeln müssen aber unterschiedlich weit bewegt werden, so daß die Level gar nicht so leicht zu schaffen sind.

Beast II

Nachdem Sie im ersten Teil darum gekämpft haben, die Kräfte des Beasts zu verlieren und wieder ein normaler Mensch zu werden, wünschen Sie sich diese Fähigkeiten sehnlichst wieder zurück. Zellek kann den Verlust seines Botschafters natürlich nicht hinnehmen und so entführt er Ihre kleine Schwester, um sie zu dem zu machen, was Sie einmal waren. Grafik und Sound sind noch besser, außerdem müssen Sie diesmal auch mit einigen Monstern kooperieren, handeln und reden.

(Robert Marz/jb)



Erste Screenshots von »Buck Rogers«



»Beast II« ist ein würdiger Nachfolger des ersten Teils

AMIGA DOS

SHOP

mit phantastischen Preisen

FANTAVISION

Lassen Sie Ihrer Phantasie freien Lauf mit diesem hervorragenden Animations- und Effektgenerator und Ihr Computer wird in wenigen Augenblicken zum Filmstudio.

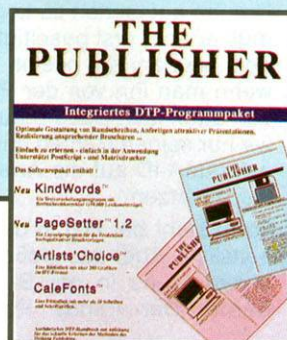
nur DM 49,95*



The Publisher AMIGA

Umfassendes Desktop-Publishing und Desktop-Presentation im Paket: Ein Textverarbeitungsprogramm, Layoutprogramm, eine Bibliothek mit mehr als 200 IFF-Grafiken und mehr als 35 Schriften.

nur DM 199,-*



AMIGA Spielebuch

Das umfassende deutsche Nachschlagewerk mit Tips & Tricks zu den beliebtesten AMIGA-Spielen. Dieses Buch zeigt Ihnen, was in Ihrem AMIGA steckt.

nur DM 29,-*



**AMIGA
Köln 90**
Wir stellen aus
09. - 11. November 1990
Halle 12, Stand 307

Speichererweiterung AMIGA 500

Mit 512 KByte zusätzlichem Speicher öffnen sich Ihrem Amiga ganz neue Welten! Mit externem Ein-/Aus-Schalter, integrierter Kalenderuhr und Batterie.

nur DM 149,-*



INSTANT MUSIC AMIGA

Musik für heiße Sohlen live aus dem Amiga! Ihre Musikideen erklingen absolut fehlerfrei, da sie automatisch an Takt und Tonart angepaßt werden.

nur DM 19,95*



* Unabhängig von der Anzahl der bestellten Produkte berechnen wir für das Inland 4,- DM bzw. für das Ausland 6,- DM Porto und Verpackung.

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag . Postfach 250 . 3440 Eschwege

DMV
Daten- und
Medienverlag

AMIGA DOS - Spieleletips

»**N**euromancer« ist endlich auch auf dem Amiga erschienen. Da es nicht gerade zu den leichtesten Spielen gehört, und immer wieder Leserfragen dazu eintrafen, haben wir und entschlossen, Tips zu diesem Spiel in die Helpline dieser Ausgabe mit aufzunehmen.

Hilfe für verzagte Cyberpunks

Da eine Schritt-für-Schritt-Lösung den ganzen Spaß am Spiel nehmen würde, gehen wir aber nur auf das Beseitigen einiger grundsätzlicher Probleme ein und drucken für diejenigen, die Probleme mit den Datenbanken haben und vor allem zur Orientierung im Cyberspace, Tabellen mit allen acht Cyberspace-Bereichen ab. Aber nun zu den Tips.

Geld, Geld, Geld...

Am Anfang ist zunächst das Hauptproblem die Beschaffung von ein paar Credits. Auch wenn das zunächst nicht so aussehen sollte, es gibt im Spiel eine ganze Menge davon, man muß nur wissen, wie man herankommt. Eine Möglichkeit ist zunächst einmal das Sparen. Man kommt mit dem UXB ganz gut über die Runden, solange man noch nicht in den Cyberspace kann. Erst dann wird ein gutes Deck, vor allem mit viel RAM wichtig, dann sollte man auch nicht kleckern, sondern klotzen. Die einfachste Methode, um an Geld zu gelangen, ist das Senden der eigenen BAMA-ID an General Armitage. Aber nur die ID und nichts anderes. Darauf sollte man sofort um 10000 Credits reicher sein, die natürlich auf den Credit-Chip geladen werden. Dann wären da noch die 10000, die man wöchentlich als Lohn erhält, wenn man in der Gehaltsliste von Hosaka geführt wird, was sich mit dem richtigen Paßwort leicht bewerkstelligen läßt. Man muß das Geld aber persönlich abholen. Nochmal 7500 bekommt man von Hosaka, wenn man ihnen ComLink6.0 beschafft. Für ein gutes Deck werden aber schon die 30000 Credits von der Bank-Gemeinschaft benötigt.

Komplexe Adventures sind nicht immer einfach zu knacken. Oft sind es nur Kleinigkeiten, die den gestreßten Spieler von wesentlichen Fortschritten trennen. Diesmal haben wir Hilfen zu »Neuromancer«, »Manhunter II«, und einige Shorttips zu bieten.

Die Source-Kontonummer erhält man von Lupus Yonderboy, dem Chef der Panther Moderns, der sich im Hinterzimmer von Larry Moe's Microsofts aufhält.

Um ihn aufsuchen zu können, muß er aber erst beseitigt werden, was am besten geht, wenn man ihn von der Polizei verhaften läßt. Dazu brauchen wir nur seinen Namen und seine BAMA-ID auf die Arrest-Liste zu setzen.

Larry steht bei Fuji in der Gehaltsliste. Aber das große Geld erwartet uns in der Bank of Berne. Der Funds Reserve

Account lautet 121519831200, der Master Code for Funds Transfer: LYMA1211MARZ. Vorher müssen wir aber ein eigenes Konto bei der Bank of Zürich eröffnet haben.

Bis dahin hilft uns ein flottes Schachspiel mit Battlechess 2.0 bei der World Chess Confederation über die Runden. Um kein unnötiges Geld bei Gerichtsverhandlungen zu verlieren, sollten wir den Richtern ruhig in das Hinterteil kriechen und vor allem auf den tollen Verteidiger verzichten. So kommt es immer nur zu Geldstrafen, die nicht abgebucht

werden können, wenn sich nichts auf dem BAMA-Konto befindet.

ROMs, Ersatzglieder und mehr...

Auf Ersatzglieder verzichten wir vollkommen, da sie die Einsatzfähigkeit im Cyberspace vermindern. Eine Ausnahme darf aber gemacht werden: Die Cybereyes aus Maas Biolabs. Dazu muß man aber erst vom Cyberspace aus die Türen entriegeln und sich bei Julius Deane die Gasmaske besorgen. ROMs sind eigentlich nur zu wenig zu gebrauchen, da sie ständig versagen. Bei Lupus Yonderboy kann man aber einen Security Pass für das Sense/Net-Gebäude kaufen, wo man mit der richtigen Nummer ein ROM mitgehen lassen kann. Die Nummer lautet:

1.Dixie Flatline
(Paul McCoyle):
0467839

Shorttips kurz und knackig

Von Beat Stadler aus St.Gallen erhielten wir einen Cheat zu »Turrican«:

Turrican

Man braucht genügend Punkte, um sich in die Highscoreliste eintragen zu können. Trägt man sich jetzt in der Liste unter »BLUESMOBIL« ein, ist es geschehen, was man zweifellos feststellt, wenn man sich seine Lives- und Granatanzeige anschaut.

Zu »Dogs of War« und »Leonardo« bekamen wir einen Tip von Andre Eckhof aus Grevelsberg:

Dogs of War und Leonardo
Durch Eingeben von »TIMBO« nach dem Ladevorgang kann man nicht mehr sterben.

Leonardo bekommt unendlich viel Leben, falls man als Paßwort »Freibier« eingibt.

Andre Kotbarski sandte uns Tips zu einer Reihe von Spielen ein, von denen wir uns nun »Rainbow Islands« und »P47« zuwenden wollen.

Rainbow Island und P-47

Bei »Rainbow Island« braucht man im Titelbild nur, wenn das Farbbrollen einsetzt, folgende Codes und als Abschluß ein Leerzeichen einzugeben, um ein paar Erleichterungen zu erzielen. Die Codes können aber nicht kombiniert werden:

BLRBJSB: Schneller laufen
RJSBJSBR: zwei Regenbögen
SSLLRRS: schneller Regenbogen
BJBJBJS: Tip1
LJLSLBS: Tip2
LBSJRWL: Continues
(???die RED.)

Wenn alles geklappt hat, erscheint unten im Bild ein Topf. Gibt man bei »P47« in der Highscore-Liste »ZEBEDEE« ein, kann man mit den Tasten [F1] und [F2] Optionen auswählen.

Die Karte und einen Tip zu »North Sea Inferno« erhielten wir von Stefan Miefert aus Osnabrück.

North Sea Inferno

Er schreibt, daß es eine Möglichkeit gibt, sich aus der Umzingelung von Feinden zu befreien. Man schießt sich eine

Seite frei und rennt in diese Richtung. Dreht man sich jetzt um, ist auch der Rücken wieder frei.

Gebrauchsanleitung für die Tabellen:

Ganz oben steht die Bereichsnummer und in Klammern der Ort, in dem sich der Cyberjack befindet. Darunter stehen die Namen der Basis, falls vorhanden, Inkkodes und Paßwörter. Falls es keine Paßwörter gibt, steht dort in Klammern das Wort »Sequencer«, was bedeutet, daß die Basis per Telefonlink nur mit dem Program Sequencer1.0 von der S.E.A. betreten werden kann. Der Punkt AI-Bekämpfung gibt an, mit welchem Skill begonnen werden muß, falls eine AI vorhanden ist. Unter dem Punkt Software/Sonstiges finden Sie die Liste der dort vorhandenen Software beziehungsweise eine Anmerkung, was man hier noch tun kann. Steht Software in Klammern, bedeutet dies, daß sie auch telefonisch abgerufen werden kann.

Tabelle 1: Bereich 0 (Cheap Hotel):						
Name:	Linkcode:	Paßwörter:	Koords:	ICE Stärke	AI Skill:	Software/Sonstiges:
Regular-Fellows	REGFELLOW	1. VISITOR	0/208/32	36	—	(Scout1; Battlechess2); Probe3
Consumer-Review	CONSUMER		0/320/64	—	—	
Asano-Computing	ASANOCOMP	1. CUSTOMER 2. VENDORS		0/16/112	72	Wertvolle Linkcodes
Worldchess Confederation	WORLDCHES	1. NOVICE 2. MEMBER	0/160/80	84	Logic	Battlechess4
Cheap-Hotel	CHEAPO	1. GUEST 2. COCKROACH	0/112/112	84	—	Rechnung editieren; Bestellungen
Panther Moderns	CHAOS	1. MAINLINE	0/224/112	132	—	(ComLink3); Blowtorch3; Decoder2; Thunderhead1; Cyberspace1
Psychologist	PSYCHO	1. NEWMO 2. BABYLON	0/96/32	96	Philosophy	Thunderhead1
Tabelle 2: Bereich 1 (Gentleman Loser):						
Name:	Linkcode:	Paßwörter:	Koords:	ICE Stärke	AI Skill:	Software/Sonstiges:
IRS	IRS	1. Taxinfo 2. Audit	1/272/64	149	—	Jammies1; Hammer2; Mimic1
Copenhagen University	BRAINSTORM	1. PERILOUS	1/320/32	149	—	ComLink4; Decoder1; Probe4; Jammies1; Doorstop1
Gentleman Loser	LOSER	1. WILSON 2. LOSER	1/416/64	150	—	Slow1; Hammer1; Probe3; Blowtorch1; Injector1
Eastern SeaboardFA	EASTSEABOD	1. LONGISLAND	1/384/32	150	—	(ComLink5); Thunderhead2
N.A.S.A	VOYAGER	1. APOLLO	1/448/32	150	Logic	Phyton2; Blowtorch4; Decoder4
Tozuku Enterprises	YAKUZA	1. YAK	1/480/80	120	—	(ComLink6; Blowtorch1-Decoder1); Drill2; Acid1; Blowtorch3.0
Justice	JUSTICE	(Sequencer)	1/416/112	150	—	—
S.E.A.	SOFTEN	1. PERMAFROST	1/352/64	149	—	(Sequencer1; ComLink4); Thunderhead2
Freematrix	CFM	—	1/352/112	Sophistry	—	—
Police	KEISATSU	1. WARRANTS 2. SUPERTAC	1/288/112	—	—	Hier kann man Larry Moe verhaften lassen
Tabelle 3: Bereich 2 (Musabori):						
Name:	Linkcode:	Paßwörter:	Koords:	ICE Stärke	AI Skill:	Software/Sonstiges:
Hitachi Biotech	Hitachibio	1. GENSPLICE 2. BIOTECH	2/32/192	260	—	In die Personalliste eintragen!
Fuji Electrics	FUJI	1. ROMCARDS 2. UCHIKATSU	2/112/240	260	—	Larry Moe's BAMA ID
Hosaka Corp.	HOSAKACORP	1. BIOSOFT 2. FUNGEKI	2/144/160	260	—	Concrek1; Injector2; Hammer4; Mimic2
Musabori Industries	MUSBORIND	1. SUBARU	2/208/208	260	—	Kuang Eleven
Tabelle 4: Bereich 3 (Bank of Berne):						
Name:	Linkcode:	Paßwörter:	Koords:	ICE Stärke	AI Skill:	Software/Sonstiges:
Free Sex Union	—	—	3/288/208	400	—	Phenomenology Informationen
Bank of Berne	—	—	3/336/160	400	Philosophy	Probe 10
D.A.R.P.O.	—	—	3/336/240	400	—	Thunderhead3; Injector3; Jammies2; Concrek2; Drill3
Turing Registry	—	—	3/432/240	400	—	Skill Upgrade; AI-Info
Screaming Fist	—	—	3/464/160	400	—	Slow3; Depth3; Pyton3 Easy Rider1; KGB1; AmorAll1
Tabelle 5: Bereich 4 (Sense/Net):						
Name:	Linkcode:	Paßwörter:	Koords:	ICE Stärke	AI Skill:	Software/Sonstiges:
Sense/Net	—	—	4/48/320	800	—	Rom Nummern
Gridpoint	—	—	4/160/320	800	—	Jammies3; Hammer5; Injector3; Thunerhead3 AmorAll2
Tabelle 6: Bereich 5 (Bank Gemeinschaft):						
Name:	Linkcode:	Paßwörter:	Koords:	ICE Stärke	AI Skill:	Software/ Sonstiges:
Bank Gemeinschaft	BANKGEMEIN	1. EINTRITT 2. VERBOTEN	5/304/320	1000	—	(Decoder1)
Bank of Zürich	BOZOBANK	(Sequencer)	5/36/368	1000	—	—
Bell Europa	—	—	5/384/288	1000	—	Thunderhead4; Acid5; Cyberspace1
Nihilist	—	—	5/416/368	1000	—	Python5; Slow4; Acid3; Logic Bomb3
I.W.S.A.	—	—	5/448/320	1000	—	AmorAll3; Hammer6; LogicBomb3; Injector5; Doorstop4
Tabelle 7: Bereich 6 (Nur mit EasyRider zu erreichen):						
Name:	Linkcode:	Paßwörter:	Koords:	ICE Stärke	AI Skill:	Software/ Sonstiges:
KGB	—	—	6/112/416	1100	Logic	Slow5; AmorAll4; Depth8; LogicBomb6; ComLink5; Injector5; Jammies4; Probe5
Maas Biolabs	—	—	6/112/480	1100	—	Sophistry Gebäudetüren entriegeln
Tabelle 8: Bereich 7 (Tessier-Ashpool):						
Name:	Linkcode:	Paßwörter:	Koords:	ICE Stärke	AI Skill:	Software/Sonstiges:
Phantom	—	—	7/320/464	2000	Battleches- s4	Hemlock1.0
Tessier Ashpool	—	—	7/384/416	2000	Phenome- nology	Spiel zuende!
Allard Technics	—	—	7/432/464	2000	AI:Neuro- mancer	...

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 Widerrufsschreibens genügt zur Fristwahrung.

**AMIGA
Köln 90**
Wir stellen aus
09.-11. November 1990
Halle 12, Stand 307

**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:

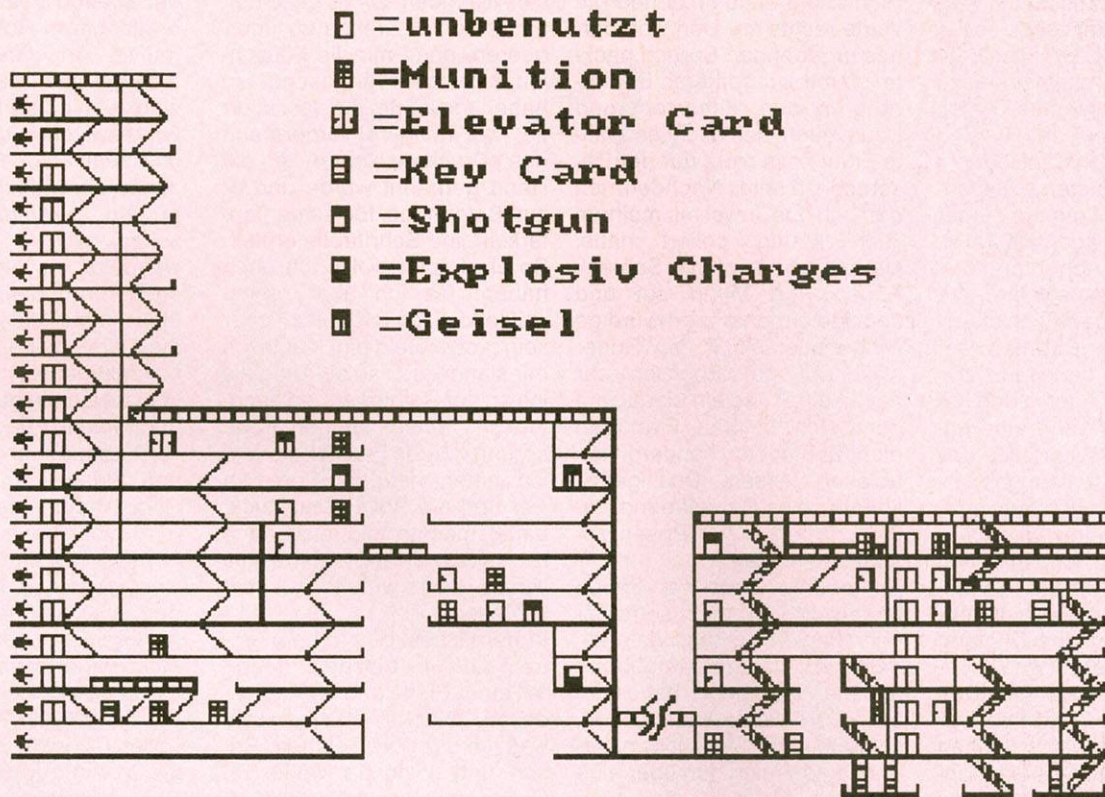
6 Ausg. = 60,- DM

12 Ausg. = 120,- DM

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV
Daten- und
Medienverlag



Karte zu »North Sea Inferno«

2.Toshiro Mifune:
6905984
3.Rombo:
5521426

Der Hochsicherheits-Code für die Tür an der Bankgemeinschaft lautet: BG1066.

ICE, Als und andere nette Kleinigkeiten...

Die Als sind gefährliche Gegner, die man nur mit Skills oder mit ganz speziellen Programmen vernichten kann. An eine AI sollte man sich aber nur mit gut ausgebildeten Skills wagen, da der Grad, wie gut man einen Skill benutzen kann, angibt, wie oft man ihn einsetzen kann. Also sollte man sich erst an Als wagen, wenn man bei Turing war. Zuerst stellen wir mit dem Psycho-Skill die Schwächen der AI fest. Die Wirkung der Treffer läßt sich auch bei Als am Stärkebalken rechts außen feststellen. Sollte ein Skill blockieren, schießen wir einfach mit einem anderen weiter, wichtig ist nur, daß wir mit dem richtigen anfangen. Die Als Phantom und Greystoke brauchen wir nicht zu bekämpfen, sie zerstören sich, wenn man ihnen spezielle Programme aushändigt (Phantom: Battlechess 4.0; Greystoke:

Hemlock1.0). Im Kampf gegen das ICE sollte man auf folgendes achten:

Alte AmorAll-Programme kann man in Verbindung mit neuen weiterbenutzen.

Als Reihenfolge im Kampf hat sich folgendes bewährt:

- 1.Der ICE-Breaking Skill
- 2.Ein Slowprogramm
- 3.Ein Viruschen.
- 4.ICE-Breaking Programme
- 5.Zwischendurch AmorAll.

Das war's von »Neuromancer«, viel Spaß beim Knacken der Datenbanken.

Manhunter - die Jagd auf Phil geht weiter...

...diesmal in San Francisco. Nach meiner Bruchlandung sammelte ich aus den Überresten des nun toten Manhunter seine Ausweiskarte und sein MAD auf. Zur Tarnung führte ich dessen Auftrag weiter.

Außer dem Ausweis und dem MAD erhielt ich natürlich auch gleich eine Wohnung, die ich erst einmal inspizierte. Die gründliche Untersuchung förderte aber lediglich ein Juwelentuch zu Tage. Nach einem schnellen Blick aus dem Fenster besuchte ich die Orte, die mir der Tracker zeigte. In der

Bank lag neben einem toten Mann eine Karte, mit der ich erst einmal nichts anfangen konnte. Der Zeitung auf dem Fußboden konnte ich nur dem Namen eines Wissenschaftlers entnehmen. Dafür erfuhr ich an der Glastür den Namen des Büroinhabers, der dort in Spiegelschrift stand: »TADTIMOV«. Das MAD bescherte mir dazu eine Adresse ganz in der Nähe der Bank. Draußen lag neben der Bank dann noch eine Leiche. Der Mann – mit dem gleichen seltsamen Zeichen auf der Hand wie die Leiche in der Bank – ist an einer vermutlich von einem Tier durchbissenen Kehle gestorben. In der Wunde steckte noch ein Zahn.

Mein nächstes Ziel war das Lagerhaus. Nachdem ich mich an den Robotern vorbei bis ins Büro durchgekämpft hatte, fand ich dort einen Holzhammer und auf dem Schreibtisch eine mit Blut geschriebene Nachricht. Sie war an einen gewissen Zac gerichtet und handelte von einem Verräter namens Mic.

Hoch im Norden am Hyde Street Pier war die Ratte im Kanaleinstieg nicht besonders erbaut davon, bei ihrem Mittags-schlaf gestört zu werden, des-

halb kletterte ich lieber den angeknackten Pfosten hoch, um die Medizinkisten zu untersuchen. Da mich dies nicht weiterbrachte, begab ich mich zum Fährgebäude, wo außer einem wenig aufschlußreichen Plakat nichts zu sehen war.

Im Embarcadero-Brunnen gab es dann aber wieder Action: Zuerst schlidderte ich im großen Bogen in die Öffnung und metzelte dann jede Menge Ratten und Fledermäuse nieder. Im Hinterzimmer lag ein Führerschein auf dem Boden und auf dem Bett ein angenagtes Individuum, dem die verschüttete Flasche mit Gegengift offensichtlich auch nicht mehr helfen konnte. Die nun leere Flasche nahm ich mit.

Auf die Eingabe »MIC STONE« reagierte das MAD positiv, mit den Informationen ließ sich jedoch nicht viel anfangen. Genauso wenig ergiebig war ein Besuch in Tad Timovs Wohnung, wo ich lediglich zu Hundefutter verarbeitet werden sollte. Dafür aber fanden die Orbs, daß ich genügend Zeit für meinen Auftrag hatte, wollten Resultate sehen und schickten mich anschließend nach Hause zum Schlafen.

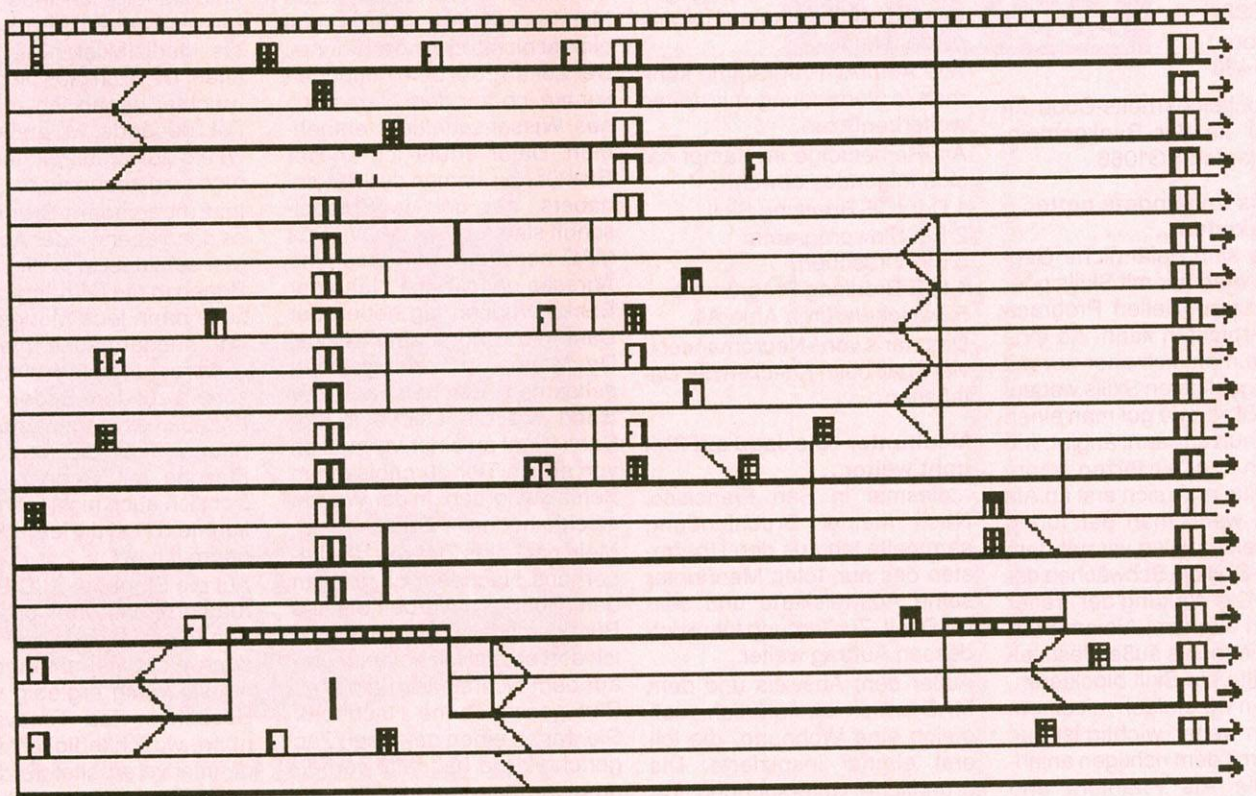
Der zweite Tag...

...versprach hektisch zu werden. Mein Tracker zeigte sechs verschiedene Leute an, die alle irgendwie miteinander zu tun zu haben schienen. Also fing ich am Pier 5 an. Hier lag auf einem verkohlten Boot ein übel zugerichteter Riesenhund, dem nicht nur ein Finger fehlte, sondern auch ein Zahn, was sich beim Abnehmen des Maulkorbes herausstellte. Den Maulkorb benutzte ich sofort nach dem Betreten der Wohnung und der Hund zog sich beleidigt wieder in den Kamin zurück. Ich fand eine Kamera und ein Wandbild, das mir schwer zu denken gab. Im Tempel, den ich nun aufsuchte umzingelten mich vier Ninjas, sobald ich mir den Schild in der Mitte des Raumes nahm. Nachdem ich ihre müden Attacken eine Zeit lang abgewehrt hatte, zogen sie gelangweilt wieder ab. Der Buddha gab mir den Weg in die Tiefen beziehungsweise Höhen des Tempels frei. Das Treppenhaus erinnerte mich an die Drachenkarte aus der Bank.

Also sah ich mir die dritte Figur rechts, die erste links und die vierte rechts an. Den sich daraus ergebenden Spruch packte ich mir zur späteren Erinnerung in den Hinterkopf und freute mich, daß auch die zweite Figur links nicht nur der Dekoration diene. Nachdem ich nämlich das Juwel mit meinem Tuch kräftig poliert hatte, klappte der Kopf vor Schreck Augen und Mund auf und spuckte ein paar merkwürdige Dinge aus: ein F, ein Ruder (was auf gut Englisch oar heißt), ein P und ein Lineal, auf dem sich in San Francisco nicht Zentimeter, sondern inch ablesen lassen. Das ganze könnte ein Rebusfreund zu FOar Plnch = Vier Prisen zusammen setzen. Am oberen Ende der Treppe begeisterte ich mich so für einen Papierdrachen, daß ich nicht auf meine Umgebung achtete. Die Ninjas hatten inzwischen Verstärkung geholt, schnitten mir den Rückweg ab und jagten mich nun über eine schmale Brücke, die auch noch mit glühenden Kohlen

gepflastert war, zu ihrem Meister. Nachdem ich es geschafft hatte, den Säuresee zu überqueren, ohne mir die Fußsohlen allzu sehr angesengt zu haben, schaute ich fasziniert zu, wie meinem Vordermann das Drachenzeichen in die Hand gebrannt wurde und er zur Belohnung für seine Tapferkeit eine Schriftrolle erhielt. So ein Papier wollte ich auch haben. Da ich aber genau wußte, daß ich in punkto Feuer kein großer Held bin, klaute ich mir stattdessen so ein Ding als ich an der Reihe war und hechtete mit einem kühnen Kopfsprung durchs Fenster. Sobald ich unten heil angekommen war und die Rolle eingesteckt hatte, machte ich mich nach rechts aus dem Staub, da die Ninjas recht wütend zu sein schienen. In dem Haus, das ich nun betrat, saß ein einsamer Mann mit einer Pfeife und fünf Tabakdosen. Nachdem ich alles, was ich über die Zeichen, Prisen und Visionen wußte im Geiste miteinander verknüpfte hatte, kam ich zu dem Ergeb-

nis, daß vier Prisen Tabak aus der zweiten Dose von links ein brauchbares Resultat bringen müßte. An diese Dosierung hielt ich mich dann auch und steckte mir die Pfeife anschließend zwischen die Zähne. Die im Rauch erscheinende Vision zeigte ein Mädchen, das sich in ein häßliches, quadratisches, grünes Monster verwandelte. Ihr Vater war davon so gerührt, daß er mir dafür eine kleine Hasenstatue gab, die ihm seine Tochter geschenkt hatte. Nun tat ich ein gutes Werk, begab mich zur Trans-America-Pyramide und befreite einen Sklaven, indem ich den Roboter ablenkte. Dazu mußte ich mich im Kreis um ihn herum- und in dem Moment weiter bewegen, in dem sich der Laser gerade auf die eigene Position gedreht hat. Das alles dauerte seine Zeit. Bis ich wieder draußen war, war der Typ längst über alle Berge. Lesen Sie in der nächsten Ausgabe von AMIGA DOS, wie Phils Abenteuer weitergehen. (mm)



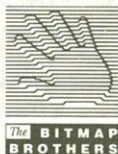
Zweite Karte zu North Sea Inferno

Besonders empfehlenswert!

CādaveR™

**AMIGA
Köln 90**
09.-11. November 1990
Halle 12, Stand 422

... ist CADAVER nicht nur weil es mit dem gleichnamigen POWER-PLAY-Prädikat ausgezeichnet wurde. Nein, da gibt es noch andere Gründe. Zum Beispiel die Bitmap Brothers, die hiermit ihr erstes Fantasy-Rollenspiel präsentieren. Aber auch der hohe Grafikstand und die Herausforderung, eine „fast“ unlösbare Aufgabe übernehmen zu müssen. Immerhin gibt es 5 verschiedene Level mit jeweils 70 Räumen. Vollgepackt mit Rätseln, Fallen, Gegnern, Zaubersprüchen und natürlich viel Arcade-Action. Wer möchte da nicht mitmischen? Kein Problem: Jetzt erhältlich für Commodore Amiga und Atari ST.



Michael Hengst in Power Play 10/90:

„Diese Mixtur aus Action-Adventure und Rollenspiel ist absolut suchtverdächtig. vor allem die teuflischen Rätsel bereiten dem Spieler schlaflose Nächte.“

Volker Weitz in Power Play 10/90:

„Cadaver kann man vorbehaltlos jedem empfehlen, der Spaß an intelligenten Computerspielen hat.“

Heinrich Lenhardt in TOS 10/90:

„Dank hohem Komfort, einfallsreichem Spielprinzip und fantastischer Grafik wird Sie das ideenreiche Programm gnadenlos fesseln.“



United Software
Das relevante Programm

ABAG *Andreas Bergmann & Adalbert Geyer GdR*

Amiga krank?

Unsere Heilmethoden sind anerkannt, die Therapiepreise günstig. Außerdem: 24-Stunden-Notdienst! Wir lassen Sie und Ihren Amiga in den schwersten Stunden nicht allein.



Rufen Sie uns an!

ABAG-Computer

☎ 0621/78 74 01 • Fax 0621/78 74 09

Schönaucr Str. 17b
6800 Mannheim 31

Amiga Software in Köln

Public Domain PD:

Einzeldiskette 3,50 DM, 10-20 Disketten 3,20 DM, ab 20 Disketten 2,90 DM. Wir führen jeweils 5 Spiele oder Anwenderpakete zu je 29,- DM. Wir führen alle Serien wie z. B. Cactus, ACS, RPD, FISH, Kickstart, Taifun, Killroy (ab 18 Jahren), Ruhrsoft u.v.a. Ebenfalls veröffentlichen wir monatlich 10 Disketten auf unserer eigenen Serie COLONIA PD, die bis jetzt 55 Disketten umfaßt. Versand per Nachnahme mit UPS. Versandkosten trägt der Besteller.

ACHTUNG PROGRAMMIERER:

Wir kaufen Ihre Programme, Spiele oder Anwender zu vernünftigen Honoraren für die kommerzielle Vermarktung oder unsere Serie COLONIA PD. Rufen Sie uns an!

"DON MANOLO" - eine Pacman-Variante mit insgesamt 63 Level • Preis: **DM 19,90**

Öffnungszeiten: Mo. - Fr. 10.00 Uhr bis 18.30 Uhr
Samstag 10.00 Uhr bis 14.00 Uhr

Pielago Software

Olperer Straße 438 • 5000 Köln 91
Telefon 0221-8903162

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, Wizard of Sound, CLI-Pack, Virus Stop, Werner-Spiel, Latein, R.O.M., StarTrek, Alf, Core Wars, Label, Amiga-Paint, Giroman, Blizzard, VirusControl, Tetric, Moria, Battleforce, Peters Quest, Super-Bilder, Billard, Einkommensteuer, DSort III, Fix-Disk, Universaldatei, Quickmenü, Diskey, Mandelbrot, Silver Bilder, Astronomie, Super-Print, Calc, Atlantis, Schach, Labelpaint! Fast alle mit dt. Anleitungen!

Alle zusammen nur **DM 79,-**

***** GRATIS-KATALOGDISK *****

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, Quizmaster, Pingball, Database-Wizard, Jazzbuchi!

PATRICK PAWLOWSKI, Software-Service
Ellerbruch 19, 2177 Wingst, 04778/7294
Versandkosten: Vorkasse DM 3,50/NN DM 6,00

YC-FARBSPLITTER YC-GENLOCK



Zwei Geräte für den Profi.
Garantiert beste Qualität und Verarbeitung. Bei OPTIVISION inklusive Service und Beratung. Fordern Sie ausführliche Unterlagen an. OPTIVISION. Der AMIGA & VIDEO Spezialist.

OPTIVISION

Heckenerstr. 16 D-5469 Windhagen
☎ 02645/4424 Fax 02645/3395

GNE

5.25" TEAC Profilaufwerk extern 279,-

1,67 MB unformatiert, 880 KB im AMIGAFORMAT, 100% kompatibel zu internem 3,5" Diskchange, abschaltbar, 4080 Track unschaltbar original standardmäßiger durchgeschalteter Bus, Amiga-farbenes extrem robustes Stahlblechgehäuse, direkt MS-DOS + PCXT/AT Karten kompatibel, mindestens 80 cm langes zugestrichenes Rundkabel, Stromversorgung über Amiga, passender BOOTSELEKTOR im Preis von 279,- enthalten, inklusive WRITE PROTECT Schalter, qualitätsgeprüft, ausführliche deutsche Bedienungsanleitung

5.25" TEAC intern A2000 239,-

Technische Daten wie 5.25" extern, wahlweise DF1 oder DF2. DF2 kostet 259,- wegen zusätzlicher Modifikationsplatine! unkomplizierter Einbau innerhalb 2 Minuten OHNE LÖTEN!!!

3.5" TEAC Profilaufwerk extern 229,-

Technische Daten wie 5.25" Laufwerk extern, Write 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! unkomplizierter 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 698,-

STAR XB 24-10 Farbband NEC P2200: 8,50 DM 1398,-

NEC P6 PLUS Farbband NEC P6/P6+: 9,- DM 849,-

Umschaltplatte mit Kick 1.3 1349,-

Umschaltplatte mit Kick 1.3 98,-

Wir liefern nur deutsche Geräte mit Seriennummer und Super-Service! 129,-
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 % errorfree)

	von Sentinel	von SONY/Collossus
bis 99 Stück	1,50 DM	1,85 DM
ab 100 Stück	1,30 DM	1,70 DM
ab 500 Stück	1,15 DM	1,50 DM

Laufwerke mit allen Extras

3,5" intern.....	155,- DM
3,5" extern, abschaltbar, Busdurchführung.....	189,- DM
5,25" extern, wie 3,5" + 40/80-Trackumschaltung.....	249,- DM
Sin City (dt.).....	79,- DM
B.A.D.....	77,- DM
DPaint III (dt.).....	240,- DM
Zoetrope 1.1.....	189,- DM
GFA-Basic 3.5.....	208,- DM
GFA-Compiler 3.5.....	129,- DM
512 KB-Erw. (A500).....	149,- DM
2 MB-Erw. (A500).....	498,- 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 #

JAMIGA Astrol. Kosmogramm

Nach Eingabe von Namen, Geburtsort (geogr. Lage) + datum werden errechnet: Sternzeit, Aszendenz, Medium Coeli, Zodiakgradienten, 12 Objektpositionen im Tierkreis, Koch/Schaeck-Häuser, Aspekte * Allgem. Persönlichkeitsanalyse mit Ideal-Partner Skala, Bild-/Druckerausgabe 3 DIN A4-Seiten, Horoskop-Diagramm - Alle Planeten + Sonne/Mond, Mondknoten - Minutengenaue Berechnung - Sommerzeiten + Koordinaten-Einlesung DM 78,-

JAMIGA BIOKURVEN

Wissenschaftl. Trendbestimmung der biologischen/seelisch/geistig/körperlichen Rhythmik - Monitor-Ausgabe monatsweise vor- + rückschreitend, Ausgabe Drucker beliebig lang mit täglicher Analyse und Kennzeichnung kritischer Tage - Absolut-+Mittelwerte - Ideal für Partnervergleich - Text-Editor für Zusätze - Wissenschaftl. Grundlagen DM 58,-

JAMIGA Kalorien-Polizei

Erstellung von Diätplänen und personenbezogene Bedarfsrechnung auf Eingabe von Größe, Gewicht, Geschlecht, Leistung - Energiebilanz nach Fett, Eiweiß, Kohlenhydraten - Ideal-/Über-/Untergewicht Einlesung Vitalstoffe, Kalorien-Lebensmittel-Tabelle, Aktivitäten-Verbrauch - Bildschirmdrucker - Drucker Ausgabe auf einzigen DIN A4 DM 58,-

JAMIGA Etikettendrucker

Druckt 40 gängige Haftetiketten-Formate nach Gestaltung in jeweils passender Bildschirm-Maske + Bestimmung der Auflagenhöhe - Ablage auf Disk für sofortige Neu-Auflage - Schriftartenwahl DM 88,-

JAMIGA IDEE-SOFT-Programme

- Exzellent in ihrer Struktur - alle Programme in Deutsch -

JAMIGA Registrierkasse

+ Normaldrucker, Beleg auf Tab. Papier 145mm - Kasenförmige Führung auf Disk für Ausdruck-Unterbrechung - Artikel/Dienstl. von Disk abrufbar - Einbindung von Firmendaten, Werbeslogans o.ä. - m/o MwSt. - Ideal für alle Gewerbe mit Bareinnahmen DM 148,-

JAMIGA GESCHÄFT

Editor für Formular-, Adressen-, Artikel-Dienstleistungsdateien - Optionen: Angebot/Kosten-Voranschlag, Auftrag/Bestellung, Auftr.Bestätigung, Rechnung, Lieferschein, Mahnung, Eingabe Hand o. Jatei - 20 Positionen/DINA4 Durchrechn. über Menge, Preis, Aufschlag/Rabatt, MwSt., Skonto - Texteditor für Zusätze - Kein Verbund zu Lager-/FIBU - Schnell, übersichtlich, Userfrdl. DM 198,-

JAMIGA Inventur, Fibu-gerecht

Kontinuierl. Bestandsverwaltung m. Bildmoment u/o Listenauswertung - Neu-Inventur durch Streichen, Ändern, Hinzufügen - Gruppeninventur nach Code - 1000 Positionen/Liste - Blätteraddition DM 118,-

JAMIGA Provisionsabrechnung

Editor für Vertreter, Kunden-, Formulardaten - 25 Positionen/DINA4, Eingabe Hand/Datei - Satz 0.01 - 99.99% - Storno, Spesenüberschr. - Durchrechnung zum Endbetrag, m/o MwSt. - schnell! DM 118,-

JAMIGA TIPIST

AMIGA als elektronische Schreibmaschine mit zeilenweisem Ausdruck und 15zeiligem Bildschirmdisplay - Je nach Druck bis zu 30 Schriftarten - Fülle auf Disk - Kopie-Ausdruck - Super! DM 88,-

JAMIGA IDEE-SOFT-Programme

- Exzellent in ihrer Struktur - alle Programme in Deutsch -

JAMIGA GELD

30 Routinen für Umgang mit Geld: Anlage - Vermögensbildung - Rentensparen - Rendite - Kredite - Lasten - Zinsen - Hypothek - Laufzeit - Amortisation - Raten - Gleichklausel - Nominal/Effektiv Zins - Akonto+Restverzinsung - Diskont - Konvertierung - kpl. Tilgungspläne Bild/Druck DM 98,-

JAMIGA DATEIVERWALTUNG

Datenfelder von je 8 Zeilen a 33 Zeichen, je Datei max. 1000 - Suchcode von max. 33 Zeichen, mit jedem mehr die Zielgruppe einengend - Optionen: Code, Nummer, alle, Blatt vor/zurück, Streichen, Ändern (zeilenweise), Hinzufügen - Druck: 80-Zeichen/Blockliste, Seitenvorschub, Etiketten, Datentafel-Maske - Gezielte Aufgaben, superschnell-übersichtlich, bedienerfreundlich, ausgesteuert

JAMIGA Adressen 68,- Galerie 118,-

JAMIGA Bibliothek 118,- Lager 118,-

JAMIGA Briefmarken 118,- Personal 118,-

JAMIGA Diskothek 78,- Stammbaum 118,-

JAMIGA Exponate 118,- Videothek 78,-

JAMIGA DEFIN DATA zum Selbstdefinieren

der Inhalte DM 148,-

Versandkosten pro Sendung: Nachnahme DM 5,70, Ausland DM 10,70, Vorkasse DM 3,-

Liste gegen adressierten Freumschlag DIN A5/DIN 1 - Händler sehr erwünscht.

I. DINKLER Am Schneiderhaus 7

Tel. 029 32/329 47 D-5760 ARNSBERG 1



6000 PD

AMIGA PD incl. 3,5" MF2DD
schon ab 1,60 DM je Diskette

Profitieren auch Sie von unserer Erfahrung
Wir versenden noch am Tag des Bestelleingangs.
Wir haben 5 Jahre AMIGA Erfahrung,
Ultraschnell - Preiswert und stets Top Aktuell.
ABO Möglichkeit, Serienz.B. Fish, Kickstart, ANTARES usw..

AMIGA PD incl. 3,5" MF2DD
1 - 79 Je PD Disk 1,90 DM
80 - Je PD Disk 1,80 DM
AMIGA PD Disk incl. 5,25" Disk
10 - 99 Je PD Disk 1,20 DM
100 - Je PD Disk 1,00 DM

2 Katalogdisketten 5,- DM, bei Vorkasse (in Briefm.)
oder fordern Sie unser Info-Material an 1,- DM
Bestellannahme Mo.-Fr. 10 - 20 Uhr, Sa. 11 - 14 Uhr
Vorkasse + 4,- DM, Nachnahme + 8,- DM,
Ausland + 16,- DM bei Vorkasse.

Versand-Service
Bis 13 Uhr bestellt
bis 16 Uhr versandt



R. Dombrowski
Postfach 710462
2000 Hamburg 71
Tel: 040/6428225
FAX: 040/6426913

Public Domain Köln-PD

Wir bauen auf unsere ca. 3-jährige Erfahrung mit PD und kopieren deshalb nur mit Verity auf 3,5" 2 DD Disketten.

bis 9	Disketten	3,45 DM
ab 10	Disketten	3,30 DM
ab 50	Disketten	2,99 DM
ab 100	Disketten	2,59 DM

Pakete (je 10 Disks)

1. Einsteigerpaket I - Spiele, Anwendungen, Grafik u.a.
2. Einsteigerpaket II - Noch mehr des Guten
3. Spiele I - ausgesuchte Spiele vieler Bereiche
4. Spiele II - denn spielen kann man immer mal
5. Grafikpaket - DBW-Render, Malprogramm, Dias u.a.
6. Anwenderpaket - Textverarbeitung, Videodatei u.a.
7. Soundpaket - Sonixsounds mit Player

1 Paket	33 DM	5 Pakete	140 DM
3 Pakete	90 DM	7 Pakete	190 DM

DON MANOLO
Der große Bruder von Pacman mit 63 Level, 2-Spieler-Modus
abspeicherbarer Highscore-Liste, ausführliche Beschreibung
19,90 DM

Infoliste gegen Rückporto. 5 Info-Disketten - 11,- DM + Porto (siehe unten). Erotikdisketten können wir nur gegen Altersnachweis liefern.

Vorkasse/Scheck: 4,- DM
Nachnahme Inland: 7,- DM
Nachnahme Ausland: 18,- DM

PETER KEIM
Vogelsanger Str. 34
5000 Köln 30
Telefon: 0221/520765

COMPUTER SHOP

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?... P.V. Computer-Shop
Bahnhofstr. 278, 8623 Wetzikon (ZH)
Tel. 019 30 79 54
Schweiz

HARDWARE • SOFTWARE • BERATUNG

Daten- und Organisationssysteme
Hard- und Softwarevertrieb

Ihr AMIGA-Fachhändler
im Bergischen Land!



Wir halten eine große Auswahl von HARD-
WARE, SOFTWARE, PD-Programmen
(über 2000 Disketten) sowie Zubehör für
alle Amiga-Rechner für Sie bereit!

Sie können sich in Ruhe alles ansehen,
kommen Sie einfach vorbei oder rufen Sie
an!

Wir würden uns über Ihren Besuch in einem
unserer Ladengeschäfte sehr freuen.

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

HARDWARE SOFTWARE ZUBEHÖR

TELEFON: 02 162 / 12073

Software		Software	
688 Attack Submarine	Sim 59,50	Legend Of Faerghail	Rol 64,50
All Dogs Go To Heaven	Act 64,50	Might And Magic 2	Adv 64,50
Back To The Future 2	Act 64,50	Ninja Warriors	Act 49,50
Battlechess 2	Str 64,50	Okolopolis	Sim A.a.
Champions Of Krynn	Rol 64,50	Oil Imperium	Str 49,50
Dan Dare 3	Act 47,50	Pirates	Adv 64,50
Dark Century	Act 64,50	Police Quest 2 (1mb)	Adv 89,50
Debut	64,50	Q 8 Team Ford	Sim 58,50
Distant Suns	Adv 119,50	Railroad Tycoon	Sim A.a.
E Motion	Str 58,50	Shad. o. L. Beast 2 & T-shirt	Act 84,50
E.s.s.	Sim 69,50	Space Rogue	Act 69,50
F 16 Combat Pilot	Sim 59,50	Their Finest Hour	Sim 69,50
F 16 Falcon	Sim 74,50	Ultima 5	Rol 69,50
F 16 Falcon Mission Disk	Too 53,50	X-copy, Hardw. Kopierprg	Too 58,50
F 16 Falcon Mission Disk 2	Too 49,50	X-out	Act 51,50
F 19 Stealth Fighter	Sim 69,50		
F 29 Retaliator	Sim 59,50		
Ghost N Goblins	Act 47,50		
Ghostbusters 2	Act 58,50	512kb int. A500, Akku-uhr absch.	149,00
Hammerfest	Act 64,50	3.5 Einbau Laufwerk für A2000	159,00
Holes Of Montezuma	Adv 64,50	3.5 ext. Markendrive, Bus, absch.	189,00
Impenium	Str 64,50	5.25 ext. Markendrive, Bus, absch.	249,00
Indiana Jones Adventure	Adv 64,50	Megamax 2000 - RAM-Card 2MB	679,00
J. Niklaus Courses 1	Too 34,50	1.5 Mb Intern für A 500	499,00
J. Niklaus Courses 2	Too 34,50	Medusa St-emulator für A500	459,00
J. Niklaus Golf	Sim 59,50	A-mos	39,50
Khalan	Adv 64,50	A-max Mac Emulator	349,00
Klax	Str 49,50	Hi A500 - 20 Mb Für A500	828,00

Versandbedingungen: Bei Vorkasse + DM 5,00, bei Nachnahme + DM 7,00,
Ausland nur Vorkasse, EC-Scheck + DM 12,00.
Bestellungen telefonisch: HOTLINE Mo-Fr 14.00 - 19.00 Uhr + Tel. 02162 / 12073
Anrufbeantw. 24 Std. täglich + Fax 02162 / 12074
Bestellungen schriftlich: HAMO K. Rüdges • Rahnerstr. 235 • 40680 Viersen 1

Speichererweiterungen

ANRUFEN !!!! LOHNT SICH !!!!
TAGESPREISE !!!!

Amiga 500	- 512 KB intern, abschaltbar	ab DM 129,-
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 Speicherentwicklungen sind autokonfigurierend, 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: 028 34 / 12 49 ; Fax: 028 34 / 69 79

DELTA-SOFT

6000 PD-DISKETTEN !!!
WIR FRIEREN DIE PREISE EIN !!!
JEDE DISK NUR DM

1,70

4 Katalogdisketten gegen 4,- DM Rückporto!

HOTLINE TEL.: (02 28) 66 54 43
F. Krüger - Postfach: 450 023 - 5305 Aifter 1
DELTA-SOFT - IM TANNENBUSCH 38-5300 BONN 1

JEDES TEIL WOANDERS EINKAUFEN ? -
BEI SCHAEFER BRAUCHST DU NUR EINMAL LAUFEN!

AMIGA-FLACHTEIL-SCANNER DM 948,00
Kann als BILDERFASUNGSGERÄTKopierer und Thermodrucker eingesetzt werden. Scandichte
200 Punkte/Zoll, Scanzzeit 10 Sekunden. Abgabe des ganzen Bildes im IFF. Auflösungen 320x200,
640x512 werden unterstützt. Ausschnittvergrößerungen möglich. Binär u. 16 Grau-Darstellungen.

VIDEOTECH-DECODER DM 298,00
Man kann das VIDEOSIGNAL eines Recorders, ScartTV oder Tuners dazu verwenden, den freien Service
des Teletextes im IFF- oder ASCII-Format abzulegen. Super-Grafik-Darstellung.

MINI MAX 500 DM 248,00
Erweiterung bis 2 MB
Preise für Aufrüstsätze siehe Gigatron-Inserat

AMIGA-ACTION-REPLAY DM 189,00
(Eurosyste) Erweiterungsmodul für den AMIGA 500
- Programmfreier, Spieltrainer, Zeitlupe, Virus-Detektor, Spielreiter, Statusanzeige und vieles mehr.
ZWEITLAUFWERK "Senator 3,5" (mit Trackanzeige DM 259,00) DM 239,00
ZWEITLAUFWERK 5,25" DM 259,00
MIDI-MASTER / Midi-Interface (Amiga-Typ angeben) DM 99,00
PRO-SAMPLER und DATEL-JAMMER DM 169,00
GENIUS oder GOLDEN-IMAGE-MAUS - voll Amiga-kompatibel DM 69,50
SYNCRÖ-EXPRESS incl. neueste Hardware DM 99,00

- Weiterentwicklung des Burst-Nibbler bei
Benutzung externer Laufwerke
DEVPACK-ASSEMBLER (M+T-Verlag) DM 149,00
AMIGA-DEUTSCH - Deutsche DOS-Befehle einlesen DM 29,95
**SPRACHENKIT - Vokabel-Trainer in Deutsch, Latein,
Engl., Französisch, Spanisch, Italienisch, Japanisch !** DM 49,95
**AMIGA-Ersatzteilliste wegen starker Schwankungen
derzeit nur auf Anfrage.**

Alle Bücher von Markt und Technik - umfangreiche Software, Hardware und Ersatzteile auf An-
frage - Versand nur gegen Vorkasse + DM 5,00 oder Nachnahme + DM 8,00.

(CLS) - COMPUTERLADEN SCHAEFER
Klingelholz 111, 5600 Wuppertal-2, Tel.: 0202/508121
Geschäftszeiten: Mo.+Di., Do.+Fr. 14-18,30 Uhr, Samstags 10-13 Uhr

AMIGA AUSTRIA
Studio für Video- & Computerdesign - Groß- & Einzelhandel - Roland Schreiff

Profisysteme
Turbo-Opt. Platten-Netzwerk

DTP VIDEO
Eigener
Reparaturservice
PD-Versand

Distributer in Österreich

NewTek GVP Lamm
Computer + Video

Fordern Sie unsere Preisliste Händleranfragen willkommen
Video- & Computerdesign - Roland Schreiff
Postfach 257- A-6010 Innsbruck / Dorfstr. 5 - A-6074 Rinn
Tel. 05223 / 88 96 Fax: 05223 / 88 97

über 9000 DISKETTEN AMIGA - PD

kopiert auf unsere 3,5"-2DD 5,25"-2DD - Disk

Disk 1 - 9	je DM 2,80	je DM 1,70
Disk 10 - 49	je DM 2,70	je DM 1,60
Disk 50 - 99	je DM 2,65	je DM 1,50
ab 100 Disk	je DM 2,60	je DM 1,40

PD-Katalog-Set

5 vollgepackte 3,5"-Disketten (kpl. DEUTSCH!)
und die neueste Ausgabe der TIME. Stets aktuell
durch Update-Service. -- DM 20,- (Vorkasse !)

Powerpacker professional

der Super-Cruncher mit diversen Möglichkeiten !
Mit ausführlicher deutscher Anleitung! DM 39,-

A.P.S. -electronic-
Sonnenborstel 31
D-3071 Steinhilke
Tel.: 05026/1700
FAX 05026/1615



OMEGA Datentechnik Tel.: 0441 / 71109

Junkerstr. 2, 2900 Oldenburg, Commodore Fachhandel

AMIGA 3000 - 16MHz Umrüstung auf 20 MHz möglich
AMIGA 3000 - 25MHz wahlweise mit 40MB und 105 MB Festplatte
Wir führen die spezial RAMs für AMIGA 3000 und A2620 / 30

AMIGA 2000 C	1695,- DM
AMIGA 500	849,- DM
AMIGA 500 C mit 1 MB ChipRAM und Uhr	949,- DM
A 2088 PC-XT Karte mit Laufwerk	575,- DM
A 2286 PC-AT Karte mit Laufwerk 10 MHz	Preis auf Anfrage
A 2630 2 MByte Prozessorkarte optional 4 MB RAM	3598,- DM
A 2091 mit 40 MB Quantum	1498,- DM
A 1950 Multisync Monitor	1098,- DM
A 590 20 MB Festplatte für AMIGA 500	849,- DM
AMIGA 2000 8 MB Karte MicroBotics "8Up" / 2 MB	679,- DM
MegaBit RAMs 511000-80ns zum Nachrüsten	14,- DM
AMIGA 500 512 kB Erweiterung, m. Uhr, abschaltbar	115,- DM
128 kB RAM Erweiterung auf 640 kB für A 2088	129,- DM
ALF2 Filecard 60 MB, 25ms, RLL	1198,- DM
ALF3 Filecard 84 MB, 25ms, SCSI 2	1698,- DM
ALF3 Filecard 180 MB, 20ms, SCSI 2	2798,- DM
Syquest Wechselplatte mit 44 MB Medium	1398,- DM
Profisampler 56 kHz V2	125,- DM
NEU Professional 68020 Board für AMIGA 500 / 2000	849,- DM
Professional 68030 Board für AMIGA 500 / 2000	1298,- DM
PageStream 1.8 DTP Software	298,- DM
NEU O.M.A. Assembler Version 1.8	149,- DM

Test AMIGA Magazin 8/90 gut Seite 122/123
A.D.D.A. V 16 Soundsampler in 16 Bit Technik für AMIGA 2000
Bitte erfragen Sie die aktuellen Tagespreise !!!

Century PD

R. Arnold Abt. ADOS

PF 10 02 07, 4300 Essen 1, Tel. 0 20 54/69 89

24 h Schnellversand !!

Wir liefern alle gängigen Amiga-Serien !!

Einzel-Disk.....4,- DM	Preise inkl. 3,5 DD Markendisk
ab 10 Disk.....3,60 DM	- mit Qualitätsgarantie
ab 20 Disk.....3,20 DM	- wir kopieren mit doppeltem Verify
ab 40 Disk.....3,00 DM	- alle Disk 100% Virus- und Errorfrei
ab 100 Disk.....2,80 DM	- alle Disk etikettiert

Einsteiger!

10 Disketten, die den Einstieg erhebl. erleichtern,
Inhalt: Spiele, Util., Erklärungen etc. nur DM 39,-

Komplettpaket !

20 Top PD Disk. mit Textverarbeitung, Kontoführung,
Spiele, Schallplatten-Video-Datenbank,
Adreßverwaltung, Haushaltsbuch, Anti-Virus-Disk,
Schach, Util. etc. nur DM 79,-

3 Katalogdisks nur 10,- DM.

Dazu gibt es wahlweise gratis von uns:
Super-Kopierdisk oder Super-Spiele-Disk.
Bei Bestellung bitte angeben.

+ Versandkosten: Inland 4,00 DM bei V.-Scheck, Nachn. 7,00 DM
Porto + Verpackung: Ausland 6,00 DM bei Eurocheck, Nachn. 12,00 DM

AURIGA Mainaustr. 38 TECHNOLOGIE 8000 München 60 Tel.: 089/8203651

Tel. Bestellannahme Mo. - Fr. 8.00 bis 18.00
unter Tel.: 0821/462511 bei Fa. Angerer

Speichererweiterung A2000:

Amiga 2000 interne Speicherkarte bis 8MB, auf-
rüstbar in 2MB Schritten mit 411000 oder 511000,
autoconfigurierend, Testsoftware, Handbuch
2MB bestückt: DM 589,- 4MB: DM 829,-

FileCards Amiga 2000:

ALF2 SCSI Controller, autoboot ab Kickstart 1.3,
hohe Datentransferrate, einfache Bedienbarkeit
durch mitgelieferte ALF2 Software
mit Quantum P40S, 19/11ms, 42MB 1398,-
mit Seagate ST1096, 24ms, 83MB 1499,-

LIVE!-Echtzeitdigitizer A2000 DM 998,-

CFA-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,-
Einzelblatteinzug SWIFT24	DM 245,-
FontCard SWIFT24, je Stück	DM 99,-

Amiga Hard & Software

512 KB für Amiga 500 mit Uhr, Test Amiga 1/90 "gut"	139,-
2 MB für Amiga 500, mit Uhr, intern, abschaltbar	548,-
2 MB für Amiga 2000, aufrüstbar bis 8 MB	598,-
3,5"-Laufwerk mit allen Extras	179,-
Medusa Atari-ST Emulator	475,-

Alles mit 6 Monaten Garantie.

Amiga Public Domain
jede Disk 1,90
Nur Versand, 24 Std. Bestellannahme.

Volker Maczlejok

Hebbelstr. 6, 4130 Moers 1
Tel. 0 28 41 / 4 12 58

Falls unser Büro gerade nicht besetzt ist,
bitten wir Sie, auf Band zu sprechen.

HORST WITTE COMPUTER



DIGITAL-TECHNIK

URBANSTRASSE 132 • 1000 BERLIN 61
TELEFON 030/693 1022-24
BTX. 693 1022-24 • TELEFAX 694 1385

DAS COMMODORE- FACHGESCHÄFT

Verkauf & Versand von
Ersatzteilen • Neuanlagen
Reparatur nach § 3, 2 How

Ihre Hotline für AMIGA-DOS-Tip

Telefon
089 / 4 39 10 87 bis -89

Telefax
089 / 4 39 10 80

IMMER UP-TO-DATE AMIGA PD BTX AB SOFORT FAX

"Pakete"

DTP-	200 Fonts, 600 Clips, 75 Druckertr. Dokumentation u. Illustration der einzelnen Fonts und Clips
Musik-	DME 2.0, Beatmaster, 25 Sounds, 450 Instrumente und Samples
Assembler- Spiele-	A68k, BLINK, DME, GOMF 10 Disk. mit den besten Spielen aus dem PD-Pool
Animation-	10 Disk. mit Animationen der Spitzen- klasse

Dem Assembler- und Musik-Paket, liegen
umfangreiche DEUTSCHE Anleitungen zu den
einzelnen Programmen bei.

D.Ptak
Pfarrgasse 23
6670 St. Ingbert
Tel.: 06894/381331
FAX: 06894/4435

24 Std.
Versand

A.Fuchs
Theodorstr. 11
5500 Trier
Tel.: 0651/28294
BTX: 065128294



Pro Com Arts

Computer- & Software-Fachversand
Pöckelstraße 101 • 85000 München 10 • Telefon (089) 535519
* Wir sind autorisierte Commodore Systemhändler *

AMIGA 500 Grundgerät 512KByte DM 780,-
AMIGA 500 512 KByte mit Uhr DM 110,-
AMIGA 500 2 MB mit Uhr nur DM 398,-
AMIGA 2000 2 MB Speicher nur DM 698,-
LAUFWERK 3,5" extern abschaltb. DM 189,-
LAUFWERK 5,25" ext. 40/80 Tracks DM 235,-
FILECARD 30 MB mit Controller DM 869,-
AMIGA 3000 AUF ANFRAGE!
Nutzen Sie Bitte auch unseren Preisgünstigen und sehr schnellen Reparaturservice!
DR T's Tiger Cub mit dt. Anleitung DM 165,-
DR T's KCS Level V 3.0 nur DM 549,-
Datamat Professional Datenbank DM 349,-
Datamat Plus Datenbank DM 140,-
Excellence! DTP in Deutsch DM 299,-
Lattice 5.05 Developer C-Compiler DM 569,-
Loerdisketten:
No Name 3,5" 10 Pack DM 9,50
Rainbow 3,5" 10 Pack 5 Farbig DM 13,-
No Name 5,25" 10 Pack DM 5,50
PD Service alle Serien ab DM 1,49

Selbstverständlich ist dies nur ein kleiner Auszug unseres umfangreichen Gesamtprogrammes. Fordern Sie bitte kostenlos und unverbindlich per Post oder Telefon (0911) 6335519 Informationsmaterial an.

HANDLERANFRAGEN WILLKOMMEN

PUBLIC DOMAIN CENTER
Postfach 3142
5840 Schwerte

Volltreffer ...

Jede nur

1,90 DM

Info anfordern!

ALSDORFER PUBLIC DOMAIN CENTER

PD Serien	Public Domain Preise		
ACS	Stück	3,5"	5,25"
Amok	1-50	2,50 DM	1,50 DM
AmSel	51-100	2,40 DM	1,40 DM
Auge	ab 101	2,30 DM	1,30 DM
Bavarian			
Cactus			
DDD			
FAUG			
Fish			
Franz			
Zeeka!			
Kickstart			
Platz			
Pudomix			
RPD			
Ruhr			
Saar			
Tailun			
TBAG			
u. v. a.			

Public Domain Sonderreihen

TuC	Stück	4,00 DM
Ollis Games	Stück	4,00 DM
Germany	Stück	4,50 DM
Zeeka!	Stück	5,00 DM

Hardware und Prolossoftware auf Anfrage, 48 Std. Versand
Versandkosten 8,00 DM + NN, Ausland nur Vorkasse + 15,00

WALTER KAMINSKI

Geilenkirchener Straße 4, 5110 Alsdorf
Tel. 02404/22963 Fax 02404/24285

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

Endlich!!

gibt es Shareware, wo die Autoren
an jeder Disk beteiligt sind. Von

SHARE

bekommt der Autor pro Disk, die
verkauft wird, 2,- DM
Unterstützen Sie unsere Idee.
2 Katalogdisketten komplett
deutsch + Antivirendisk +
1 Spiel erhalten Sie gegen 15,- DM
Vorkasse bei:

Herrmanns & Kommelter
Vom-Bruck Platz 45
Telefon 021 51 / 39 98 33
4150 Krefeld 1

Btx- Btx- Btx- Btx-

IBM-kompatible PCs XT/AT,
C 64, AMIGA und Atari ST

Der große Versender Deutschlands, aus BTX

Programme ab: **10,- DM**

Ihr Ansprechpartner für: * Spiele *

* Stengel

Wir kaufen und verkaufen:
Gebraucht-Spiele (nur Originale)

Info unter Tel.: 0521/763918
ab 17 Uhr anfordern
BTX: 0521/763918 - 0001/2

Computershop H.-J. Stengel

Brakhofstraße 21B, 4800 Bielefeld 16

★ EINZIGARTIG ★

Easy Work 2.0

die Reparaturanleitung

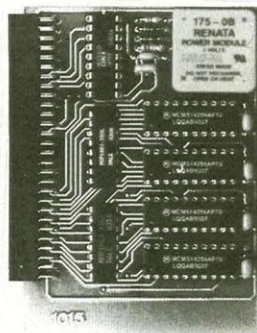
für den Commodore AMIGA 500
mit Meßwerten und
Oszilloskop-Bildern,
System-Fehlerbeschreibung.
Leicht verständlich, bestens
geeignet für Anfänger und Profis.

Preis 49,90 + NN + Versand.
Bestellen Sie gleich!

Peter Schmidramsl

Jagdfeldring 45
8013 Haar (bei München)
Telefon 089 / 46 43 83

AMIGA 500 512 k Speichererweiterung



- * Megabit Technologie
- * Abschaltbar inkl. Schalter
- * Uhr und Batterie

DM 119,-

ohne Uhr DM 98,-
2 MB A 500 DM 459,-
2 MB A 2000 DM 598,-
Kickstart Umschaltplatine DM 29,-
3,5"-Laufwerk extern DM 178,-

Hardware Design Udo Neuroth
Postfach 525, 4250 Bottrop
Tel. 020 41 / 204 24 ab 16 Uhr

Spiele für Amiga

SPICHERERWEITERUNG AMIGA 500

auf 1 MB, inklusive Uhr, abschaltbar 178,00 DM
mit dem Spiel DUNGEON MASTER (dtsch.) 248,00 DM

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

NORDIC POWER	TITEL	PREIS
Der Freezer für den Amiga 500!	Aquanaut	74,80 DM
	Fighter Bomber	89,80 DM
	It came from the Desert	84,80 DM
	Kaiser	109,00 DM
Fordern Sie ausführliche Unterlagen an!	Little Computer People	19,80 DM
	Treasure Island Dizzy	19,80 DM
	Rockstar eats my Hamster	19,80 DM
	Space Harrier II	59,90 DM

X-Copy II + Cyclone + Hardware nur 69,00 DM

Versandkosten:

5,00 DM bei Warenwert unter 100,00 DM
Warenwert über 100,00 DM Versandkosten freil!

COMPYSHOP

Gneisenastr. 29
4330 Mülheim/Ruhr
0208-497169 + 496178



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 und donnerstags von 16.30 bis 18.00 Uhr unsere Hotline an. Dort stehen wir Ihnen mit Rat und Tat zur Seite, wenn Sie eine der folgenden Nummern wählen: 056 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, senden Sie diese unter dem Kennwort "Problemecke" ein.

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.

PD preiswert?

Ich möchte Euch gerne mal schreiben, was uns vom Computerclub "A.U.G.E. 4000" an Eurer Zeitschrift stört:

Überall steht, daß PD-Software eine sehr gute Sache ist. Leider kommt aber zu dem Wort gut auch immer das Wort "preiswert". Doch was ist eigentlich preiswerte PD-Software? Wenn ein Händler 5 DM und mehr für eine Diskette nimmt und dann auch noch behauptet, dies decke nur seine Unkosten, dann hat er den

Sinn der Public Domain nicht verstanden. Wir von "A.U.G.E. 4000" veranstalten zweimal jährlich eine PD-Copyfete, um solchen Händlern den Wind aus den Segeln zu nehmen. Dort kann sich jeder bis zu zehn Disketten kopieren lassen. Wer seinen eigenen Computer mitbringt, kann aus unserem PD-Pool (zirka 3200 Disketten) soviel kopieren, wie er/sie möchte. Bei der letzten Copyfete im April sind sogar Amiga-User aus dem Raum Frankfurt extra nach Düsseldorf gekommen. Es hat sich jeder zirka 250 Disketten kopiert. Unsere nächste Copyfete findet am 20./21. Oktober bei folgender Adresse statt:

**Haus der Jugend
Lacombletstr. 10
4000 Düsseldorf**

Während der Fete finden auch noch lehrreiche Kurse für Anfänger statt (zum Beispiel »CLI«, »DPaint«,...). Aus dem geringen Eintrittsgeld von 5,- DM erweitern wir fortlaufend unseren PD-Pool. Wer PD-Disketten zum Tauschen mitbringt oder uns ein neues selbstgeschriebenes Programm für unsere PD-Serie zur Verfügung stellt, zahlt keinen Eintritt.

Wolfgang Erler (A.U.G.E. 4000),
Düsseldorf

Grundsätzlich ist gegen das Kopieren nicht-kommerzieller

Programme nichts einzuwenden. Die meisten PD-Anbieter dürften allerdings seriös genug sein, um für das geforderte Geld auch gute Ware zu liefern.

(Red.)

Nachtrag zum »NoClick«

Auf der Fish-Disk 246 befindet sich das Programm »NoClick 3.6«; dieses Programm in das Verzeichnis »C« kopieren, anschließend in die Startup-Sequenz eintragen, das wär's. Einige Laufwerke reagieren leider nicht auf »NoClick3.6«. Wenn das so ist, sollte man sich die Fish 312 besorgen, dort finden Sie das Programm »TrackSalve«, das auch das Klicken abstellt. Zum Problem »Libraries und crunchen« ist zu sagen mit dem »PowerpakerVxx« können sehr wohl Include- und Library-Dateien gepackt werden. »PowerPaker« findet sich in der PD-Serie »Tbag«.

Die »gecrunchten« Dateien werden von unserem Amiga-DOS erkannt.

Abschließend erlaube ich mir, eine Bitte zu äußern. Ich befinde mich zur Zeit in Haft, ich würde gerne mit einigen Amiga-Usern(innen) über unsere Maschine Briefwechsel führen. Können Sie meine Bitte mit meiner Anschrift veröffent-

lichen? Ich bin 47 Jahre alt und Fan vom »Devpac-Assembler«.

Horst B. Munke
Landesgerichtstr. 4
A-7000 Eisenstadt

Vielen Dank für den Tip. Interessierte Leser sollten sich bitte mit einem frankierten Rückumschlag melden, da Herr Munke sicherlich nur einen eingeschränkten Etat besitzt.

(Red.)

MS-DOS und Amiga

Wie lassen sich MS-DOS-Disketten mit dem Amiga lesen und beschreiben? Es gibt da wohl einige Programme, mit einem davon habe ich das Problem nicht in den Griff bekommen.

Wie läuft der Datenaustausch zwischen zwei Computern (auch mit XTs) über die serielle Schnittstelle ab, wie muß das Kabel gelötet werden, wie wird das softwaremäßig gesteuert? Wie kann man den Amiga für die Videoverarbeitung, insbesondere als Schnittcomputer benutzen?

Christoph Arnemann,
Fischerhude

Zur ersten Frage: Es gibt in der Public Domain ein Programm, das MS-DOS-Disketten lesen und beschreiben kann (Test er-

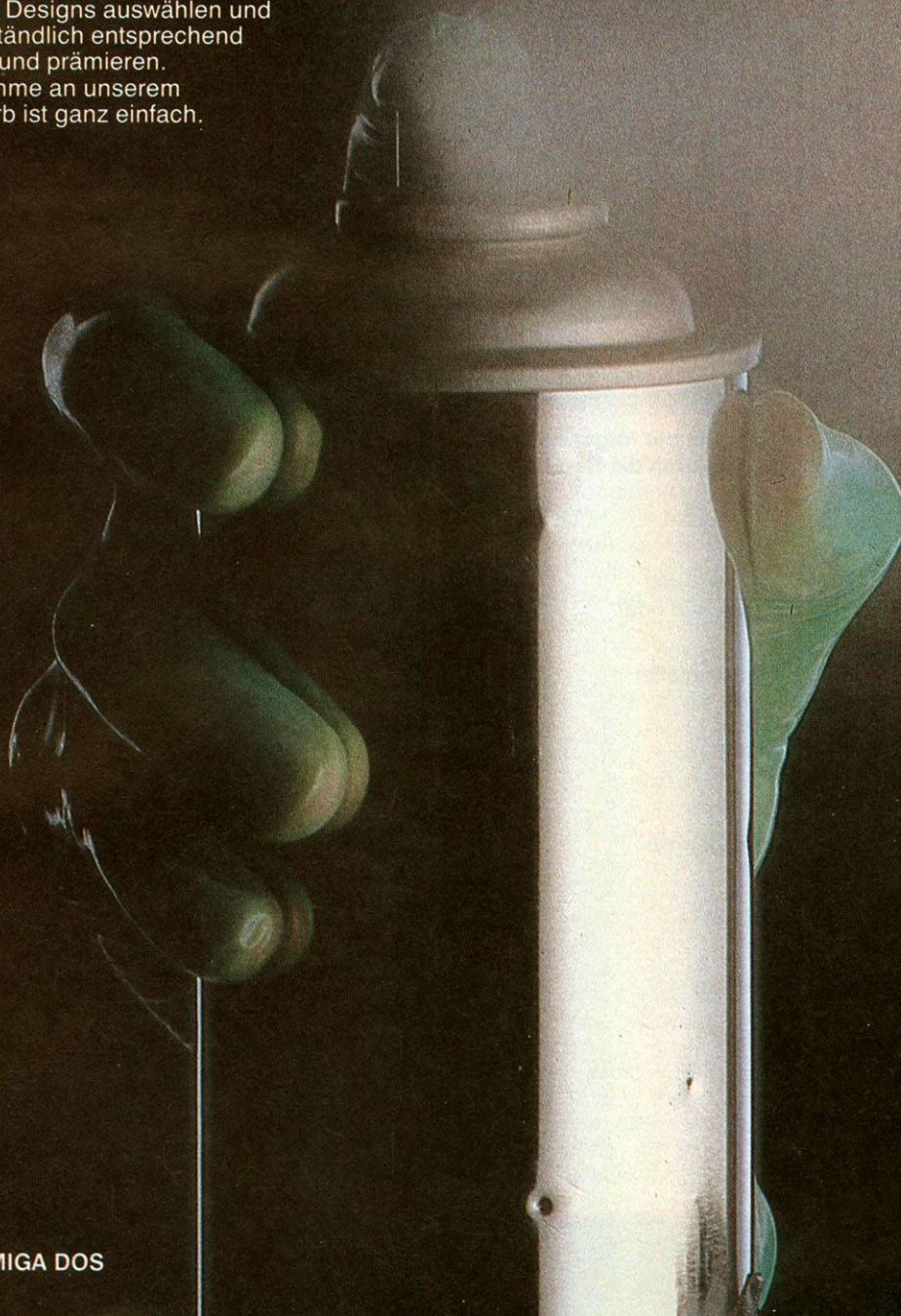
Wir suchen den schönsten AMIGA

Sie sind Amiga-Fan. Sie lieben Ihren Computer, weil er anders ist als der MS-DOS-Einheitsbrei, eben kreativ und bunt. Mal ehrlich, eigentlich ist es doch schade, daß eine so tolle Maschine äußerlich so attraktiv ist wie die berühmte "graue Maus".

Werden Sie doch einfach mal selbst tätig, und geben Sie Ihrem Amiga eine persönliche Note. Wie? Das bleibt ganz Ihnen überlassen. Von (A)irbrush bis (Z)eichenstift ist alles erlaubt, was A500 bis A3000 verschönert. Eins müssen wir aber hier einschränkend anmerken: Sollte Ihr schöner bunter Amiga, nachdem Sie wieder aus Ihrem Farbrausch erwachen, nur noch bunt sein, können Sie uns natürlich nicht regreßpflichtig machen. Aber auch hier gilt, wie überall in der Computerei, ohne eingehende Vorplanung gibt's nur Chaos.

Natürlich soll Ihr Kunstwerk nicht im stillen Kämmerlein bleiben. Wir wollen die schönsten Designs auswählen und selbstverständlich entsprechend vorstellen und prämiieren. Die Teilnahme an unserem Wettbewerb ist ganz einfach.

Wenn Sie Ihrem Amiga ein neues Gesicht gegeben haben, schicken Sie uns ein möglichst formatfüllendes Foto von mindestens 9x13 cm Größe; dann können Sie davon ausgehen, daß wir in der Lage sind, Ihr kreatives Genie auch entsprechend zu würdigen.



Senden Sie das ganze bis zum 10. November (Datum des Poststempels) an den

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

Wie bei allen unseren Wettbewerben dürfen natürlich auch hier DMV-Mitarbeiter und deren Angehörige leider nicht teilnehmen.

Wir meinen, Sehenswertes muß mit ebenso Sehenswertem belohnt werden. Deshalb wartet auf den Gewinner eine Reise zur PCS '91 in London! Neben Flug und Hotel bieten wir Ihnen gute Betreuung und selbstverständlich einen Blick hinter die Kulissen. Als weitere Preise winken Drucker, Zusatzkarten und jede Menge guter und brandaktueller Software. Also nichts wie ran, Ihr Amiga wartet schon!

(jb)



Das Superding

Großer PD-Wettbewerb

Public Domain – ein Begriff, der wohl jedem Amiga-Besitzer etwas sagt. Und ein Begriff, der Ihnen Glück bringen kann.

Und das gibt es zu gewinnen:

- eine Serie »Fish« von der Nummer 1 bis Nummer 360 im Wert von 1800,- DM, gestiftet von der Firma Hermanns & Kommelter,
- von Digital Marketing eine Serie der eigenen PD-Reihe im Wert von 200,- DM,
- die Time-Serie im Wert von 50,- DM,
- 20 Antivirendisketten im Wert von 100,- DM und
- diverse Time-Special-Disketten im Wert von 100 - DM, bereitgestellt von der Firma A.P.S.,
- drei PD-Pakete von der Firma Donau-Soft,
- von der Firma Nordsoft eine optische Maus,

- ein Abonnement der Kickstart-PD-Serie für ein Jahr von der Firma Software Empire,
- von der Firma Kaminski 5 mal 5 Disketten nach Wahl,
- von der Firma Dombrowski eine Serie der Antares-Disketten,
- ein Superpack 50 und ein Superpack 2, zur Verfügung gestellt von der Firma Pawlowski,
- die ersten 50 United-Disks von der Firma Hieske

Ihre Postkarte richten Sie bitte an folgende Adresse:

DMV-Verlag
Redaktion AMIGA DOS
Kennwort: PD-Wettbewerb
Postfach 250
D-3440 Eschwege

Eine unüberschaubare Anzahl von Programmen bietet sich dem Amiganer an. Auf dem großen Amiga-PD-Markt findet man alles, angefangen bei Grafik und Sound über Programmiersprachen, Tools, Utilities bis hin zu kompletten Anwendungsprogrammen.

Sie haben nun die Chance, eines von mehreren Komplett-Paketen reiner PD-Software zu gewinnen. Was Sie dafür tun müssen, ist ganz einfach: Schreiben Sie uns eine Postkarte, auf der Ihr liebstes, am meisten genutztes oder das Ihrer Meinung nach am besten programmierte PD-Programm steht. Wir machen daraus eine Hitparade der zehn besten PD-Programme.

Alle Einsender nehmen dann an der Verlosung teil.

Wir bedanken uns bei folgenden Firmen für die Mitarbeit am Wettbewerb durch die Bereitstellung von Preisen:

Digital Marketing
 D. Mückter
 Krefelder Str.16
 5142 Hückelhoven-Baal
 Tel.: 02435/2086, 428 oder 1295

Herrmanns & Kommelter
 Vom-Bruck-Platz 45
 4150 Krefeld 1
 Tel.: 02151 / 399833

Nordsoft
 Schwenecker & Behnke
 Heidbergstr. 75
 2800 Bremen 21
 Tel.: 0421 / 611430

Kaminski, Walter
 Geilenkirchener Str. 4
 5110 Alsdorf
 Tel.: 02404/22963

Pawlowski, Patrick
 Software Service
 Ellerbruch 19
 2177 Wingst
 Tel.: 04778 / 7294

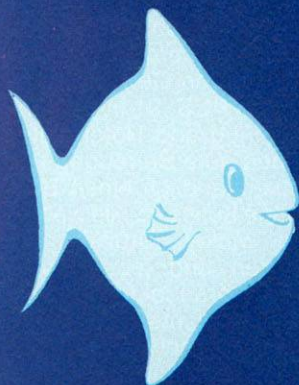
A.P.S. Electronic
 Sonnenborstel 31
 3071 Steimbke
 Tel.:05026/1700

Donau Soft
 Maik Hauer
 Postfach 1401
 8858 Neuburg
 Tel.: 08431/49798

Software Empire
 Lange Str. 118
 2880 Brake (Utw)
 Tel.: 04401 /-71348

Dombrowski, Rüdiger
 Postfach 71 04 62
 2000 Hamburg 71
 Tel.: 040/6428225

Hieske, Dieter
 Schillerstr. 36
 6700 Ludwigshafen
 Tel.: 0621 / 673105



Mitarbeiter des DMV und Ihre Angehörigen dürfen leider (wie immer) nicht teilnehmen; der Rechtsweg ist ausgeschlossen. So, und nun bleibt uns nur noch, Ihnen viel Glück zu wünschen, vielleicht sind Sie der oder die Glückliche, der (die) demnächst "in PD-Disketten baden kann".

(vb/jb)

folgt in dieser Ausgabe!). Dieses Programm finden Sie auf der Fish-Diskette 327.

Zur zweiten Frage: Um einen Datenaustausch zwischen zwei Computern in den Griff zu bekommen, brauchen Sie ein serielles Kabel (im Fachhandel für wenig Geld zu kaufen) und ein Programm, das die Daten hin- und herschiebt. Das beste und billigste Programm ist »Kermit«, das es in der Public Domain gibt. Wenn Sie die Kabel selber löten wollen, dann müssen Sie sich zwei 25polige SUB-D-Stecker und Kabel besorgen. Eine genaue Bauanleitung finden Sie in der Amiga DOS 5/90, Seite 30. Dort finden Sie noch weitergehende Informationen über »Kermit«. Um etwas über Videoverarbeitung und Amiga zu erfahren, kann auch in der Ausgabe 5/90 nachgelesen werden.

(Red.)

Defekte Workbench?

In der letzten Ausgabe blieben doch einige Fragen offen. So zum Brief »Fragenkatalog«; das Rattern von leeren Diskettenlaufwerken kann man mit dem Programm »NoClick« abschalten (Anm. d. Red.: siehe Anfang der Leserbriefe).

Zum Brief »Kann nicht lesen, kann nicht schreiben«. Wie ich aus dem Brief entnehmen konnte, treten die »Read/Write-Errors« nur bei der Workbench-Diskette des Herrn Lowinski auf, da er schreibt, 1001 Virenkiller an die Workbench herangelassen zu haben. Es ist eigentlich kaum möglich, daß beide Laufwerke über kaputte Schreib-/Leseköpfe verfügen, da das Kopieren mit diversen Kopierprogrammen ja zu funktionieren scheint. Viel eher ist anzunehmen, daß die besagte Workbench-Diskette defekt ist. Eine fehlerhafte DOS-Checksumme (läßt sich mit einem Diskettenmonitor, zum Beispiel DiskX von Steve Tibbet, leicht korrigieren) kann beim Kopieren einzelner Dateien oder Verzeichnisse mit dem Befehl »Copy« des »CLI« zu einem »Read/Write-Error« führen. Kopierprogramme ignorieren diesen »Fehler« sehr häufig und kopieren ihn einfach mit. So kommt es scheinbar zu fehlerfreien Kopien. In hartnäckigen Fällen können Pro-

gramme wie »DiskSalve« und »DiskFix« weiterhelfen. Alle genannten Programme sind in der Fish-Serie zu finden. Nun zu einer anderen Frage. Was ist mit dem Spiel »Damocles« (Mercenary II) los? Ende letzten Jahres bereits angekündigt, scheint es jetzt noch immer nicht auf dem Markt zu sein. Ist Novagen pleite? Oder hat Paul Woakes sich mit der Programmierung des Spiels übernommen?

Michael Herrmann,
Steinfurt

Mercenary II finden Sie als Test in der Ausgabe 10/90

(Red.)

PC-Dateien kopieren

Ich besitze einen Amiga 2000 mit PC-Karte und Festplatte. Davor hatte ich einen Amiga 500 mit Speichererweiterung und einem zweiten Laufwerk. Jetzt habe ich das Problem, daß einige Programme aus der Zeit des A500 (V1.2) auf meinem Amiga 2000 (V1.3) gar nicht oder nur noch teilweise laufen! Daß Programme gar nicht mehr laufen, kann ich mir aufgrund Eures Artikels (7/90) jetzt erklären, aber nicht, warum Programmteile nicht mehr laufen. Zum Beispiel nehmen wir mal DPaint III. Auf dem A500 lief das Programm ohne Probleme, und jetzt auf dem Amiga 2000 läuft der Animationsteil nicht mehr.

Er wird im Menü nur noch in Geisterschrift aufgeführt und ist nicht mehr zu bedienen. Die gleiche Diskette läuft aber auf meinem alten A500 korrekt! Und das will mir nicht in den Kopf.

Eine weitere Frage: Es besteht ja mit den Janus-Software-Befehlen »Aread« und »Awrite« die Möglichkeit, Dateien von der Amiga-Seite auf die PC-Seite zu kopieren und umgekehrt. Es gibt aber auch Software, mit der es möglich ist, die 3,5-Zoll-Amiga-Laufwerke als PC-Laufwerke zu nutzen (zum Beispiel PC-Handler). Nehmen wir mal an, man besitzt (natürlich gekauft) diese Software, so wie ich, und möchte damit von der PC-Seite auf die Laufwerke zugreifen, so muß man das ja mit den obengenannten Befehlen tun.

Klappt ja auch hervorragend, wirklich (zum Beispiel »Aread NW1: Autoexec.bat c:\autoexec.bat /b«).

Nun habe ich aber festgestellt, daß die beiden Befehle nicht mit Jokerzeichen arbeiten, beziehungsweise nur bedingt, was bei mehreren Dateien Stunden dauern kann. Wie läßt sich das leichter bewerkstelligen?

Kai Heidemann,
Stadland

Zur Frage 1.: Daß DPaint III nicht mehr auf dem A2000 läuft, hat meistens nur eine einzige Ursache, und zwar Speicherplatzmangel. Durch verschiedene Erweiterungen (zum Beispiel Laufwerke, Festplatte, PC-Karte) wird Speicherplatz weggenommen, und manche Programme laufen nicht mehr richtig, insbesondere trifft das auf das speicherplatzfressende DPaint zu.

Zur Frage 2.: Um viele Dateien zu kopieren, empfehlen wir das PD-Programm »Kermit« und ein serielles Kabel. Schauen Sie sich bitte dazu unsere Ausgabe 5/90 an.

(Red.)

Lösungen

Ich habe einige Lösungen zu Fragen aus der Problemecke

1. Frage von Herrn M. Röhm, Iserlohn, AMIGA DOS 9/90, Seite 76.

Herrn Röhm kann auf folgende Art geholfen werden:

Ein Hardware-Eingriff ist notwendig. Man muß die Select-Leitungen an dem für die Floppy-Laufwerke zuständigen Port-Baustein ändern.

– Für Amiga 1000 ist dies Port-Baustein U6S

– Für Amiga 500 ist dies Port-Baustein U8

– Für Amiga 2000A ist dies Portbaustein U11

– Für Amiga 2000B ist dies Portbaustein U301

Die Pins 13 bis 16 entsprechen DF0: bis DF3:

Wenn DF2: als DF1: angemeldet werden soll, muß demzufolge Pin 14 mit Pin 15 vertauscht werden.

Hardwaremäßig ist dies am besten zu lösen, indem man den Baustein aus der Fassung nimmt, zwei Fassungen übereinander steckt und die beiden Pins an einer Fassung abzwickt und mit Kabel ver-

tauscht anlötet. Jetzt den Port-Baustein wieder oben draufstecken.

Genauso ist auch ein Bootselektor realisierbar (beispielsweise Vertauschen von Pin 13 mit Pin 14 über einen Umschalter. Jetzt kann wahlweise von DF0: oder DF1: gebootet werden.

2. Antworten zu den Fragen von Herrn Gottschalg, Krefeld, AMIGA DOS 9/90, Seite 76.

Zu Frage 1 und 2: Beide Laufwerke sind über die Janus-Software ansprechbar. Man kann MS-DOS-Files lesen und schreiben, jedoch ist meines Wissens eine Formatierung von Disketten nicht möglich.

Zu Frage 3: Eine Erweiterung auf 640 KByte ist ohne Steckkarte nicht möglich. Eine Erweiterung auf dem Motherboard ist nur auf 512 KByte möglich.

Zu Frage 4: Der RAM-Speicher des Amigas kann vom Sidecar als RAM-Disk benutzt werden. Auch hier führt der Weg über die Janussoftware.

Zu Frage 5: Das 5 1/4-Zoll-Laufwerk ist vom Amiga aus nicht nutzbar.

Zu Frage 7: Solange der Requester »Software-Failure« sichtbar ist, kann man problemlos noch stundenlang im PC-Window mit dem Sidecar arbeiten. Erst wenn der »Guru« da ist, geht nichts mehr.

Zu Frage 9: Der Grad der Kompatibilität ist sehr hoch, es läuft praktisch jede Software, die auf einem normalem XT auch läuft. Nur bei Software, die direkt auf die Hardware zugreift (zum Beispiel Druckerschnittstelle), kann es Probleme geben. Beispielsweise läuft ein Programm, welches mir die Daten vom PC über die Druckerschnittstelle an einem Sharp-Pocketcomputer überspielen soll, nicht. Der Grund liegt vermutlich in der Software-Emulation des Amiga. Ausdrücke mit den bekannten Textverarbeitungsprogrammen funktionieren jedoch problemlos.

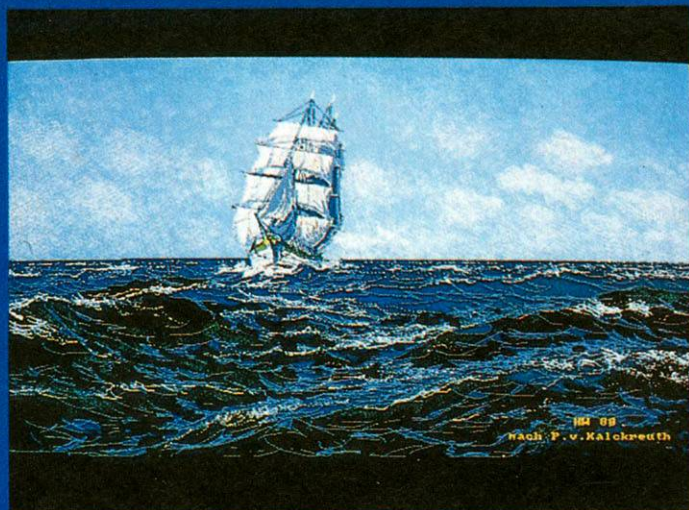
Jens Lange,
Konstanz

Die Redaktion behält sich vor, Leserbriefe in gekürzter Form wiederzugeben.

AMIGA
DOS

Pixel-Panorama

Der Phantasie der Amiganer sind nur wenige Grenzen gesetzt, der Amiga macht vieles möglich, das ohne diese Maschine wahrscheinlich nie realisiert worden wäre.



»Auf hoher See« von H. Wiesendanger

Wer von Hause aus schon mit Grafik und Musik zu schaffen hatte, experimentiert mit Genuß an den neuen Möglichkeiten herum, die ihm der Amiga eröffnet.

Wie die Nutzungsmöglichkeiten des Amiga reich gefächert sind, so breit spannt sich auch das Spektrum der Motive, die hier zu Computerbildern umgesetzt wurden.

Segelschiffe und das Meer sind ein beliebtes Motiv von H. Wiesendanger aus Zürich. Von ihm stammt das Bild »Auf hoher See«.

Den HAM-Modus bemühte U. Drücke für sein Bild »Orchidee«. Obzwar recht gegenständlich, wirkt das Bild surreal und ist von einer phantastischen Stimmung umgeben.

Raytracing der originellen Sorte präsentiert C. Mommer. Seine Fraktalgalerie benutzt Apfelmännchengrafik in einer völlig neuen Art und Weise.



»Silver Galerie« von C. Mommer



»Orchidee« von U. Drücke

Haben Sie ein wenig Lust bekommen beim Pixel-Panorama mitzumachen? Nichts leichter als das: 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 Softwarepaketes im Wert von zirka 100 DM.

(hs)

Impressum

Herausgeber
Christian Widuch

Redaktionsleitung
Stefan Ritter

Chefredaktion
Markus Matejka (mm)

Chef vom Dienst
Matthias Bloß

Leitender Redakteur
Jürgen Borngießer (jb)

Redaktion
Heinrich Stiller (hs), Claus Daschner (cd),
Vera Brinkmann (vb), Thomas Baum (tb)

Freie Autoren dieser Ausgabe
Robert Marz, Michael Anton, Andreas Polk,
Franz-Josef Reichert, Antje Hink, Edgar Meyzies,
Robert Münch, Ottmar Röhrig, Oliver Gumtau,
Jürgen Seibel, Timo Siebert, Ingmar Reyer,
Guido Coenen, Garry Glendown, Oliver Wagner,
Bernd Rudolf, Holger Lubitz, Kay Bolay,
Siegfried Rings, Bert Schwarm, Willi Bäcker,
Thorsten Budesheim, Matthias Kühn

Redaktionsassistent
Anke Kerstan-Müller (ke), Susanne Reckelkamm (re)

Produktionsleitung
Gerd Köberich

Bereichsleitung
Claudia Ebbrecht (Fotosatz/Lektorat),
Margarete Schenk, Helmut Skoupy
(Montage/Reprografie)

Layout
Patricia Reifenhausen, Petra Kuch

Fotodesign
Achim Schulte

Fotosatz
Marcus Geppert

Montage/Reprografie
Manuela Eska, Peter Gajewski

Werbegestaltung
Mohamed Hawa

Anzeigenleitung
Wolfgang Brill

Anzeigenverkauf
DMV-Verlagsbüro München
Zaunkönigsweg 2c, 8000 München 82
Telefon (0 89) 4 39 10 87, Telefax 0 89/4 39 10 80
Leitung: Britta Fiebig
Anzeigenverkauf: 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) 8 09-0,
Telefax (0 56 51) 8 09-333

Vertrieb
Verlagsunion Erich Pabel-Arthur Moewig KG (VPM),
Friedrich-Bergius-Straße 20, 6200 Wiesbaden

Druck
Druckerei Jungfer, 3420 Herzberg

Bezugspreise
»AMIGA DOS« erscheint monatlich.
Einzelpreis DM 6,50/sfr. 6,50/6S 52, -

Abonnementpreise
Die Preise verstehen sich grundsätzlich einschließlich
Porto und Verpackung.

Inland:
12 Ausgaben: DM 70,-
6 Ausgaben: DM 35,-

Europäisches Ausland:
12 Ausgaben: DM 100,-
6 Ausgaben: DM 50,-

Außereuropäisches Ausland:
12 Ausgaben: DM 120,-
6 Ausgaben: DM 60,-

Bankverbindungen:
Postcheck 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 geben nicht
in jedem Fall die Meinung der Redaktion wieder.

Die Inserenten

ABAG.....	152	Jochheim Computertuning.....	73
ALPHA 2000.....	79	Keim.....	153
Alsdorfer PD Center.....	155	Kirschbaum.....	93
Ami Shows Europe.....	31	Kramer.....	153
A.P.S. Elektronik.....	2,154	Kupke Computertechnik.....	49
Auriga.....	154	Mac-Soft.....	99
B + S Computer.....	153	Maczijek.....	154
Bonito.....	101	Markt + Technik.....	121
Century.....	154	Masoboshi.....	53
CIK-Computertechnik.....	155	Maxon Computer.....	19
CLS Computerladen Schäfer....	153	Neuroth Hardware-Design.....	155
Com Data.....	43	Nordsoft.....	93
Computing.....	93	Nürnberger PD-Studio.....	79
Compy Shop.....	155	Omega Datentechnik.....	154
CSV Riegert.....	43	Optivision.....	152
Delta Konzept.....	41	Pawlowski.....	152
DELTA SOFT.....	153	PD Versand.....	154
DIT.....	97	Pielago Software.....	152
DMV.....	22,23,134,145,148	Pro Com Arts.....	155
Dombrowski.....	153	PUBLIC DOMAIN CENTER.....	155
Donau-Soft.....	101,152	Pulsar Vertrieb.....	79
Edotronik.....	108	P.V. Computershop.....	153
FSE Strauß-Elektronik.....	51	ROSSMÖLLER.....	39,125
Fujitsu Deutschland.....	166,167	Schmidramsl.....	155
German-Soft.....	153	Schrettl.....	154
GFA Systemtechnik.....	45	SKY WARE.....	93
GNE.....	152	Soft Art + Hardware Tute.....	101
Gold Disk.....	11	Stengel Computershop.....	155
Hagenau Computer.....	165	G. v. Thienen.....	99
HAMO.....	153	Tröps + Hierl.....	93
HERMES Software.....	59	United Software.....	151,168
Herrmanns & Kommelter.....	2,155	Vesalia Computer.....	63
HK-Computer.....	4	WENNGATZ.....	25
IC-InterComputing.....	27	Wittes Digitaltechnik.....	154
IDEE Soft.....	152	Wolf Computertechnik.....	97
IDS Hard- + Software.....	33		

Die ABO-Auflage enthält eine Beilage des Schneider-Verlages.

In punkto Neuvorstellungen auf dem Büchermarkt schreitet die Entwicklung nicht so schnell voran, wie im Soft- und Hardware-Bereich. Wir haben einige Exemplare herausgepickt, die Lesenswert sind.

Nur wenige Themen rund um den Computer werden so kontrovers diskutiert wie Kunst und Computer. In dem vorliegenden Band wird ein Versuch unternommen, sich dem Thema auf verschiedenen Wegen anzunähern. Zum einen beschreibt der Autor recht global Mittel und Wege, mit welchen er Computerbilder erarbeitet. Zum anderen werden in einem Galerieteil Kunstwerke präsentiert, deren Aussagen durch Lyrik bekannter Autoren unterstrichen werden sollen.

Freie Malerei auf AMIGA

Die Schilderung der Arbeitsweise wird an fertiggestellten Projekten des Autors festgemacht, so daß immer ein unmittelbarer Bezug zur tatsächlichen Arbeit am Computer gegeben ist. Jedoch ist der Band weit davon entfernt, eine Anleitung zur Benutzung von Grafikprogrammen zu sein. Dabei steht dem Autor die Beschreibung des kreativen Prozesses, der einem Bild vorausgeht, im Vordergrund.

Freie Malerei auf dem Amiga ist wohl das erste Kunstbuch rund um diesen Computer. Obwohl vielfach Arbeitsweisen beschrieben werden, sollte der Band als Bilderbuch betrachtet werden, in dessen Verlauf der Autor den Versuch unternimmt, anhand eigener Bilder die Faszination des Mediums zu beschreiben. So wurde aus Freie Malerei auf dem Amiga ein gediegen gestalteter Kunstband, der sich an Kunstliebhaber richtet. Für den Anwender bringt der Band, von gelegentlicher Inspiration einmal abgesehen, nur wenig.

(hs/vb)

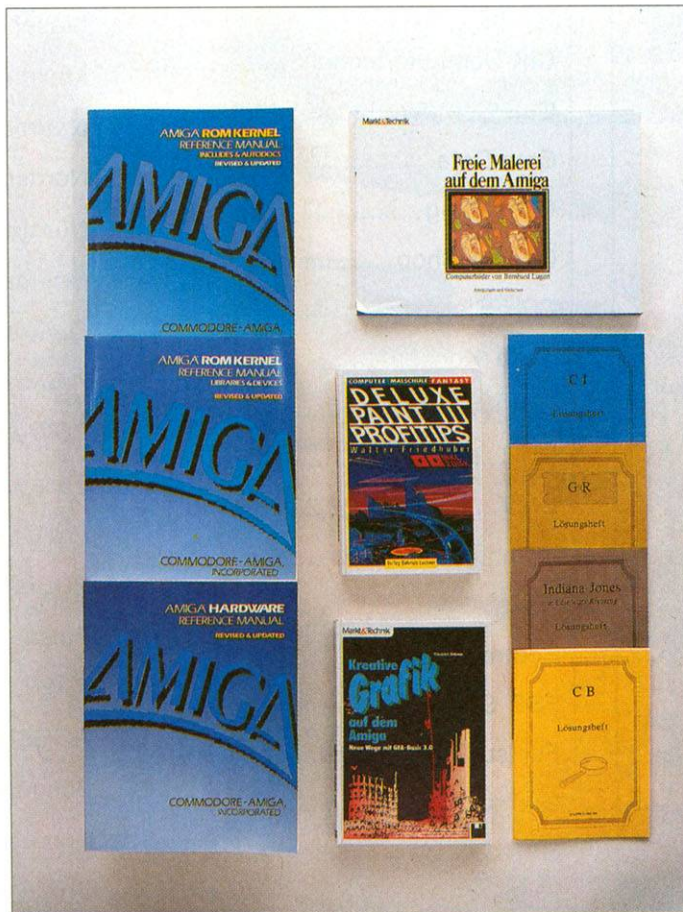
Name: Freie Malerei auf dem Amiga
Autor: Bernhard Lugert
Verlag: Markt & Technik
ISBN: 3-89090-233-2
Preis: 98,- DM

Für die eingefleischten Programmierer ist das Lesen der Reference Manuals schon längst zur Pflicht geworden. Wie sieht es aber mit den normalen Anwendern aus? Sind auch für sie diese Bücher unverzichtbar? In der neuen Version umfaßt das Gesamtwerk drei Bücher. Das erste befaßt sich mit der Programmierung von Libraries und Devices. Die Include-Files und die Autodocs sind Gegenstand des zweiten Buches. Ganz im Zeichen der Hardware steht das dritte Buch.

Manuals nicht unbedingt als Lehrwerk aufzufassen sind. Vielmehr stellen sie ein kompetentes Nachschlagewerk dar. Für den Einsteiger sind diese Bücher wenig geeignet, da vom Leser die Beherrschung der Sprachen C und Assembler in allen drei Büchern vorausgesetzt wird. Außerdem sollten Englischkenntnisse mitgebracht werden, da die Reference Manuals nur in englischer Sprache vorliegen. Auf insgesamt 900 Seiten werden Definitionen erläutert und anhand von Beispielen ver-

Als Ergänzung zum ersten Werk kann das Manual Includes&Autodocs angesehen werden. Jede einzelne Funktion des Amiga-Betriebssystems wird einschließlich aller Übergabeparameter und den möglichen Ergebnissen erklärt. Es stellt somit ein reines Nachschlagewerk dar und ist die ideale Ergänzung zum Manual Libraries & Devices. Dort wird zwar die Programmierung der einzelnen Komponenten beschrieben, es wird jedoch keine Auflistung der einzelnen Befehle gegeben. Neben den Funktionen zu den Libraries und Devices sind die gesamten Include-Files für die Sprachen C und Assembler abgedruckt. Damit bleibt dem Anwender ein mühseliges Durchforsten der Files auf Diskette erspart. Auch auf die Linker-Libraries wird eingegangen. Die Syntaxangaben der Befehle beziehen sich voll und ganz auf die Sprache C. Für Assemblerprogrammierer werden die Register angegeben, in denen die Parameter übergeben werden müssen. Zusätzlich findet man in diesem Band noch ein Kapitel über das IFF-Format. Die Autoren gehen dabei nicht nur auf das IFF-ILBM-Format, sondern auch auf die anderen nicht so gängigen Formate, wie Text und Musik ein.

Die Hardware des Amiga steht im Mittelpunkt des dritten Manuals. Sämtliche Beispiele sind in Maschinensprache abgefaßt. Kenntnisse dieser Sprache sind demzufolge unumgänglich. Nach einer kurzen Einleitung wird gleich ein sehr interessantes Thema behandelt: die Programmierung von Grafik. Auf rund 130 Seiten wird beschrieben, wie der Copper manipuliert wird, wie man eigene Playfields aufbaut oder Sprites über den Bildschirm huschen läßt. Im Gegensatz zum Manual Libraries & Devices findet der Anwender hier keine Listings am Ende eines Kapitels. Vielmehr werden innerhalb der Kapitel Teilausschnitte von Listings vorgestellt, die dann zu einem ablauffähigen Programm erweitert werden. Dies macht den Umgang mit diesem Buch nicht gerade einfach. In diesem Stil wird dann die Programmierung der Audio-, der Blitter-, der System-Control- und der Interface-Hardware



Die Amiga-Manuals

Für den Systemprogrammierer ist wohl das erstgenannte eine wichtige Informationsgrundlage. Auf den ersten 230 Seiten erfährt der Anwender alles über die Programmierung von Intuition. Jedes Kapitel behandelt ausführlich ein bestimmtes Thema. Jeweils direkt im Anschluß findet der Anwender ein oder mehrere Beispiel-Listings, die den (oft trockenen) Stoff festigen. Hierbei ist anzumerken, daß die Reference

deutlich. Auf spezielle Themen wie Exec oder Graphic wird ebenso vertieft eingegangen, wie auf die verschiedenen "kleinen" Libraries. Der zweite Teil des Buches beschäftigt sich mit der Programmierung von Devices. Es wird alles vom Audio Device bis hin zum Trackdisk-Device ausführlich erläutert. Abgerundet wird das Buch durch ein Kapitel über Resources. Was der Anwender in diesem Buch nicht finden wird, ist ein Kapitel zum Thema DOS. Hier werden lediglich Hinweise auf andere Informationsquellen gegeben.

abgehandelt. Das Buch wird abgerundet durch einen Anhang, in dem Adressen und Listen aufgeführt sind. In knapper Form wird noch auf das Thema Autoconfig und CIA eingegangen. Alle drei Bücher liefern dem Anwender fundierte Kenntnisse über das System, wenn auch manchmal in einer etwas schwer zu verstehenden Form. Die Kenntnisse der englischen Sprache und der Programmiersprachen C und Assembler sind aber umgänglich.

(Andreas Polk/vb)

Name: Amiga Manuals
Autor: diverse
Verlag: Addison-Wesley
Poppelsdorfer Allee 32
5300 Bonn 1
Best.Nr.: 553-51461-6
Preis: 210,- DM

Mit dem Buch Kreative Grafik auf dem Amiga erweitert Markt&Technik seine Reihe der GFA-BASIC-Bücher. Wer GFA-BASIC 3.0 besitzt und schon immer mehr über Computergrafik wissen wollte, der sollte einen Blick auf dieses Buch von Friedrich Belzner werfen.

Kreative Grafik auf dem Amiga

Der Autor versteht es, dem Leser fundiertes Wissen zu vermitteln, ohne daß dieser den Eindruck hat, auf der Schulbank zu sitzen und "zu büffeln". Mundgerecht bekommt der Interessierte Grundlagen zur Verfügung gestellt, um nach dem Lesen des Buches eigene Projekte angehen zu können. Gute und nicht zu lange Beispielprogramme reizen einen zum weiteren Erforschen der wunderbaren Grafikwelt auf dem Amiga. Da alle Listings auf Diskette mitgeliefert werden, ist ein flüssiges Durcharbeiten des Buches gewährleistet.

Der Autor versteht es nahezu perfekt, sowohl Anfänger als auch Fortgeschrittene mit seinem Buch anzusprechen. So wird im zweiten Kapitel auf relativ einfache Dinge, wie das Öffnen von Screens und Windows eingegangen. In weite-

ren Kapiteln wird ein Programm entwickelt und besprochen, das zur Bildschirm-Manipulation dient und sehr komplex ist. Daß in ein Buch über Grafik auch Bilder gehören, ist keine Frage. Markt & Technik hat dieses Buch besonders reichhaltig grafisch ausgestaltet. Dies regt natürlich zum Weiterlesen an. Für ein Buch nicht alltäglich sind die Hardware-Anforderungen, die die abgedruckten und mitgelieferten Programme benötigen. Der Autor empfiehlt 1MByte RAM als Grundvoraussetzung. Wer nur 512 KByte RAM sein eigen nennen kann, der muß mit einigen Einschränkungen rechnen. Abschließend bleibt nur zu bemerken, daß das Buch Kreative Grafik auf dem Amiga (Neue Wege mit GFA-BASIC 3.0) rundherum gelungen ist. Die durch den Titel gesetzten Erwartungen, werden voll erfüllt.

(Ingmar Reyer/vb)

Name: Kreative Grafik auf dem Amiga
Autor: Friedrich Belzner
Verlag: Markt & Technik
ISBN: 3-89090-227-8
Preis: 79,- DM

Der Softwareversand Safer Games hat eine ganze Reihe von Lösungsheftchen zu Spielen, hauptsächlich Adventures, herausgebracht. Stellvertretend für die Masse schauen wir uns vier Hefte an, die alleamt einen ordentlichen Eindruck machten.

Lösungshefte zu verschiedenen Sierra-Spielen

Allen gemein ist, daß sie in einem lockeren Schreibstil verfaßt, und wenn möglich und erforderlich, mit Karten aufgepeppt wurden. Eine Einleitung ermöglicht einen schnellen Einstieg in die Lösung. Die Lösung selbst ist dann jeweils in viele kleine Kapitel und Absätze unterteilt, so daß man gezielt nach der Lösung für ein bestimmtes Problem innerhalb des Spiels suchen kann. Im

Anhang befindet sich meistens eine Liste aller Orte, die im Spiel vorhanden sind, verknüpft mit stichwortartigen Angaben zu den Aktionen, die dort ausgeführt werden müssen.

Wir fanden diese Heftchen eine nette Idee, stellen sie doch eine Alternative zu den teuren Original-Lösungsheften dar, die es noch dazu nicht zu jedem Spiel zu kaufen gibt. Diese sind meist in englischer Sprache gehalten und ersparen das oft stundenlange Suchen nach Tips zu den Spielen in Computerzeitingen. Ein Punkt fiel dann aber doch noch negativ ins Auge: Bei den Sierra-Spielen steht nicht der komplette Name auf dem Titel, sondern nur eine Abkürzung. Auch sonst ist nirgends im Heft der Name des gelösten Spiels zu finden. So kann es eine Weile dauern, wenn man mehrere der Heftchen hat, bis man die passenden Spiele zugeordnet hat, wer kommt schon sofort auf »Codename Ice-man«, wenn er die Buchstaben »Cl« sieht. Aber das ist nur ein kleiner Fehler, der den Eindruck der Heftchen nur unwesentlich trübt, die alles in allem eine runde Sache sind.

(Robert Marz/ vb)

Name: Lösungshefte
Autor: Susanne Rupp, Ingo Krause, Claus Vester
Verlag: Safer Games
Preis pro Heft: 19,- DM

Das Zeichen- und Animationsprogramm »DeluxePaint III« findet aufgrund seines weiten Anwendungsspektrums immer mehr Zuspruch. So lassen auch die Dokumentationen nicht lange auf sich warten.

Deluxe Paint III Profitips

Mit DeluxePaint III Profitips will der Autor nicht den Weg gehen, das Malprogramm und seine einzelnen Funktionen zu erklären. Er will vielmehr dem Leser vermitteln, wozu »DPaint III« fähig ist. Dies tut er anhand eines ausführlich gegliederten

Grafikkurses, dessen Übungen zur näheren Veranschaulichung auf zwei Disketten fertige Zeichnungen beiliegen. Da das Buch die Funktionen von »DPaint III« nur minimal beschreibt, sollte der Käufer dieses Buches mit dem Umgang des Programms schon vertraut sein. Am Anfang des Buches werden einige Begriffe aus der Kunst erklärt, welche bei Zeichnungen mit »DPaint III« eine Rolle spielen. Dann steigert sich der Schwierigkeitsgrad langsam von der einfachen Farbgebung über Objektstudien bis hin zu komplexen Animationen. Dabei wird der Leser aber nicht nur passiv belehrt, sondern kann mit einzelnen Übungsaufgaben sein Können im Bereich Kunst/Computer erweitern. Das fertige Ergebnis kann dann mit Hilfe der zwei beigelegten Disketten verglichen werden, auf denen alle Aufgaben von professionellen Grafikern erledigt wurden. Die Stilrichtung des Buches geht mehr in die Richtung traditionelles Malen, also Portrait oder Landschaftszeichnungen. Käufer, die meinten, mit diesem Buch liege Ihnen der Schlüssel zur idealen Ballerspielgrafik in der Hand, werden sich schwer enttäuscht sehen. Spielegrafiken und -schriften werden in diesem Buch so gut wie gar nicht behandelt. Dieses Buch bricht ein neues Thema an – den Bereich Computer und Kunst. Wie weit diese Stilrichtung mit dem Amiga zu verbinden ist, sei dem Einzelnen selbst überlassen. Das Buch versucht, dem User diese Richtung in kleinen Schritten verständlich zu machen, was ihm nicht nur aufgrund der einfachen Schreibweise recht gut gelungen ist. Für Interessierte auf jeden Fall zu empfehlen!

(Timo Siebert/vb)

Name: Deluxe Paint III Profitips
Autor: Walter Friedhuber
Verlag: Gabriele Lechner
ISBN: 3-926858-24-9
Preis: 98,- DM



Im nächsten Heft

Von A bis Z – Zubehör für den Amiga

Alle Jahre wieder – freut man sich auf geschenktes Zubehör für seine Lieblingsbeschäftigung. Bevor es zu spät ist, geben wir Ihnen eine Übersicht über Nützliches, Notwendiges und kleine Nettigkeiten rund um den Amiga.



Super-Grafik erleben Sie mit dem Spiel
»Beast 2«

WShell

Die neue »WShell 1.2« soll die Mängel der Amiga-Shell und des CLI ausmerzen.

VGA-Karte für den Amiga

Eine alternative Grafikausgabe ermöglicht die Hard-/Software-Lösung HAM-E.

AmigaBASIC, die zweite

Achtung Einsteiger, Fortgeschrittene und überhaupt alle Freunde kostenloser Sprachen: Wir starten ab der Ausgabe 12'90 einen neuen AmigaBASIC-Kurs.

CLI-Tools

Zwei neue CLI-Werkzeuge zieren seit kurzem den Softwaremarkt: »LoGG« der Schweizer Firma Hermes und der »DOS-Manager« aus der Markt&Technik-Schmiede.

Arriba... Arriba...

Nein, es geht mit Sicherheit nicht um die schnellste Maus von Mexiko, aber mit Schnelligkeit hat es schon etwas zu tun und in erster Linie mit der Größe. Wir haben für Sie die kleinste Einbaufestplatte der Welt getestet: die Arriba.

Und was sonst???

Wie immer leistungsfähige Programme, intelligente Tips&Tricks, aktuelle Infos zu den neuesten Produkten in den Bereichen Hardware, Anwendung und Spiele.

PD-Workshop

Nach dem Motto "Heraustrennen und Sammeln" stellen wir Ihnen nützliche Tools und Anwendungen aus dem großen PD-Pool vor.

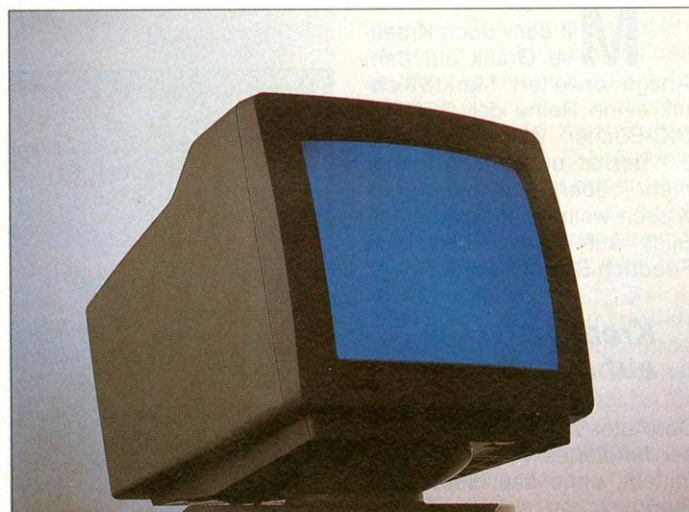
Die nächste

AMIGA DOS

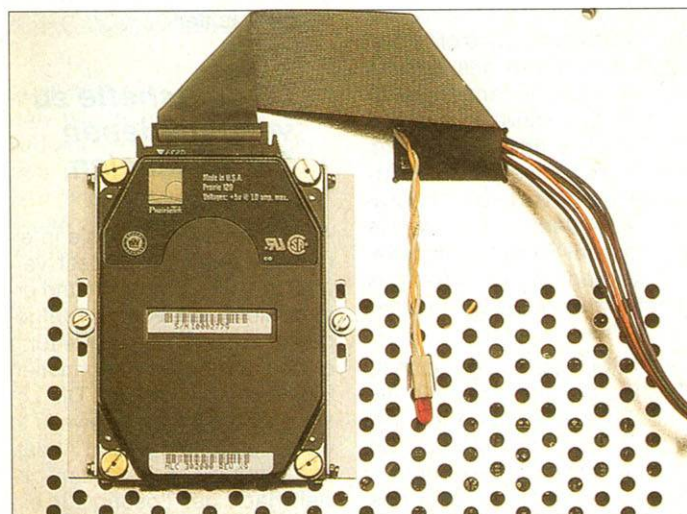
finden Sie ab

14. November '90

bei Ihrem Zeitschriftenhändler



Brillante Bilder mit dem Highscreen Monitor



Die kleinste interne Festplatte der Welt für den A500: Arriba

Jetzt geht es noch besser

Deluxe VIEW

THE ULTIMATE VIDEODIGITIZER FOR PAL-AMIGA COMPUTERS

Wir stellen aus
09.-11. November 1990
Halle 10, Stand 625

Testsieger

Amiga Special 1/90
Amiga Magazin 7/89

Hardware des Jahres

Amiga Extra 1/90

AMIGA-Test

sehr gut

10,8

von 12

GESAMT-
URTEIL
AUSGABE 7/89

PHILIPS

PHILIPS

- ☆ Color-Modus einstellbar von 2-4096 Farben (in allen Auflösungen, abhängig von den Möglichkeiten der Amiga-Hardware).
- ☆ SW-Modus einstellbar von 2-16 Farben (in allen Auflösungen).
- ☆ Folgende Bildschirm-Auflösungen können gewählt werden: LoRes, MedRes, HiRes und Interlaced.
- ☆ Alle genannten Auflösungen wahlweise auch in Overscan.
- ☆ Frame-Modus = frei definierbarer Frame (Gummiband-Rahmen) zum Digitalisieren oder Nachbearbeiten bestimmter Bildschirmausschnitte.
- ☆ Alle Auflösungen bzw. Color- und SW-Modi können im laufenden Programm gewechselt werden (ohne Neustart).
- ☆ Alle Digitalisier-Modi sind auf Schnellmodus (Fast-Mode) umschaltbar.
- ☆ Bestmögliche Bildbegrenzung durch Spezial-Algorithmen.
- ☆ Alle nur erdenklichen nachträglichen Softwareberechnungen sind möglich: Helligkeit, Kontrast, Farbsättigung, Rot-, Grün-, Blauanteil, Bildschärfe, Negativ, Dithering usw. wahlweise mit oder ohne Neuberechnung der Farbpalette.
- ☆ Sehr umfangreiches Palette-Menü für alle Farb- und SW-Modi mit folgenden Reglern und Funktionen: RGB, HSV, Palette von SW oder Color austauschen bzw. kopieren, Palette-Restore, Farben spreizen "Spread" oder kopieren "Copy to". Einzelne Farben können manuell gesperrt werden. Paletten können aus bereits digitalisierten Bildern eingeladen werden. Automatisches Erzeugen der Jim Sachs-Palette mit "Autopalette" kein Problem.
- ☆ Sehr umfangreiches Druckermenü mit voller WB 1.3-Unterstützung.
- ☆ Sehr umfangreiches Diskmenü mit Harddiskunterstützung.
- ☆ Abspeichern von Masterbildern (RGB-Auszüge) kein Problem.
- ☆ Alle Bilder werden im IFF-Format gespeichert (wichtig für die Nachbearbeitung mit Amiga-Malprogrammen).
- ☆ Bester Bedienungskomfort durch flimmer- und flackerfreie Menübildschirme auch in den Auflösungen "Interlaced" und "HiRes".
- ☆ Problemloser Wechsel in allen Auflösungen und Modi ohne das Programm verlassen zu müssen.
- ☆ Fast alle Funktionen sind zusätzlich auch auf F-Tasten gelegt.
- ☆ Interne RAM-Verwaltung jetzt mit "Dynamic Allocation".
- ☆ Im Lieferumfang sind enthalten: Digitizer-Hardware, Steuer-Software, ein 47seitiges deutsches Handbuch, sowie 2 Zusatzprogramme (Slide-Show und Animation).
- ☆ Update-Service (bereits bekannt von DE LUXE SOUND).



Die abgebildeten Bildschirmfotos wurden mit
DE LUXE VIEW digitalisiert

Neu DLV 4.1 für A500/2000	nur 398,- DM
Neu DLV 4.1 für A 1000	nur 398,- DM
Neu DLV 4.1-Demo	nur 15,- DM
2 Disketten mit Animationsdemo	



**hagenau
computer**

g
m
b
h

Alter Uentroper Weg 181 * 4700 Hamm 1
Bestellservice: Tel. 02381/880077
Bestellservice: Fax 02381/880079
Händleranfragen willkommen

VON DEN
PROGRAMMIERERN VON...

FUTURE WARS
FUTURE WARS

FUTURE WARS
FUTURE WARS

TRÄGER DES "IN DIN" - PREISES FÜR DAS BESTE ÜBERSEE-SPIEL

EIN WEITERES

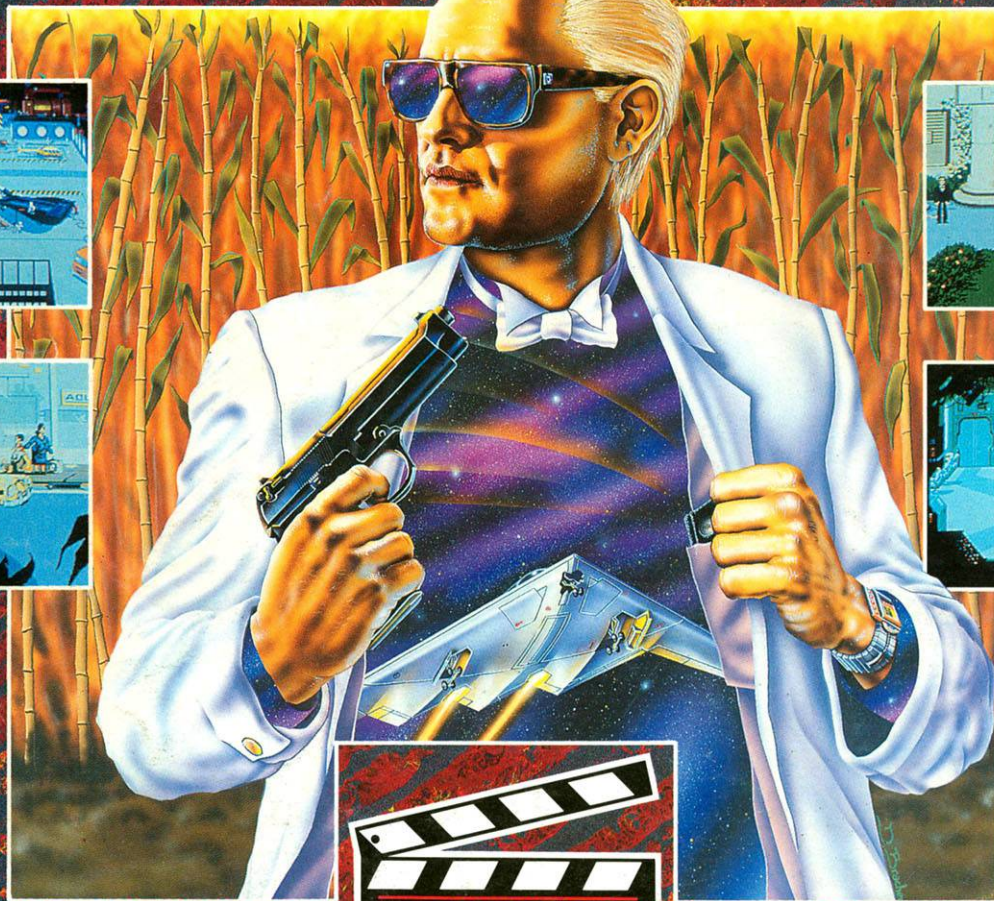


SPITZEN PRODUKT...

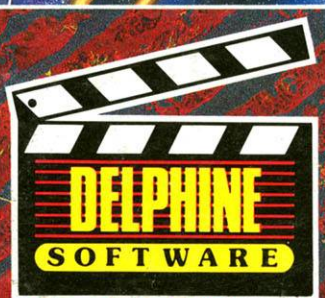
OPERATION STEALTH



Bilder der
ST Version



Bilder der Amiga-
Version



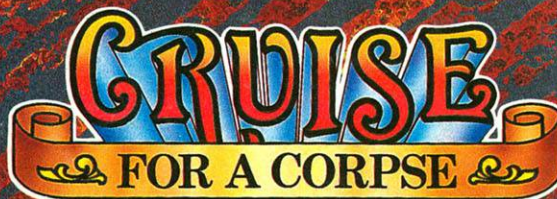
Operation Stealth stürzt Dich in die Welt internationaler Spionage, ein brandneues, interaktives Cinématique™-Adventure. Operation Stealth ist das zweite Spiel, das das einzigartige Cinématique™-Betriebssystem, seit Future Wars weiterentwickelt und verbessert, nutzt.

- Vollständig mouse-gesteuert
- Einfaches Point-and-Click-Interface
- Kompatibel mit AdLib- und Roland-Sound-Card (MS-DOS-Version)
- 256 Farben in der VGA-Version unter MS-DOS.
- Super-Soundtrack aus den Delphine-Studios

DEMNÄCHST...



U.S. Gold Ltd. Units 2/3, Holford Way,
Holford, Birmingham B6 7AX.
Telephone: 021-625-3388.



© 1990 DELPHINE SOFTWARE.
Alle Rechte vorbehalten. Cinématique
ist eingetragenes Warenzeichen der
Delphine Software.

ERHÄLTICH FÜR
ATARI ST • AMIGA
MS/PC-DOS KOMPATIBEL