

KICK START

COMPUTER MAGAZIN

KICK IT OUT!

Der Linkvirus-Killer

GOME 3.0

Der Gurubuster

DRAGONS LAIR

Die Lösung

LISTINGS

Übersetzungsprogramm

3D-Realtime-Animation

Debugger



**Workshop: Icons • Photon Video • Spiele • News
Schrifterkennung • s/w-Multisync • Tips & Tricks**



IntroCAD Version 2.1
 Sehr leistungsfähiges CAD Programm. Hervorragend geeignet für Abbildungen, Diagramme, Grundrisse, Service-Bücher, Werbegrafiken u.v.m.... Eigene Druckroutinen bieten nahezu Laserqualität auf Matrixdruckern.
 Komplett deutsch 189,- DM

DISKMASTER
 Das optimale Disk-Utility-System



Diskmaster
 Das beste Utility Programm. Alle Fähigkeiten des Amigas durch einen Klick mit der Maus. Ersetzt vollkommen das CLI! Spielt IFF Musik- und Instrumentendateien, zeigt IFF Bilddateien mit einem Klick!
 Komplett deutsch 119,- DM

PIXmate
 KOMPLETTES BILDBEARBEITUNGSSYSTEM



Pixmate
 Neue deutsche Version 1.1. Das Profi-Programm zur digitalen Bearbeitung von Grafiken. 3000 Spezialeffekte mit Konvertierung aller Amiga-Grafikformate. Sehr schnell durch neue 'Hyper-Slice Technology'.
 Komplett deutsch 149,- DM

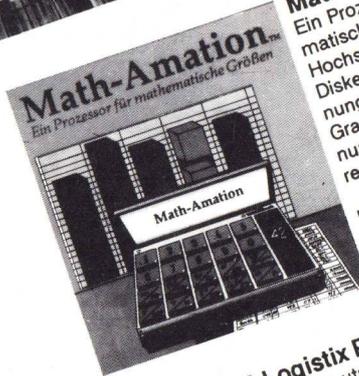


Pro Sound Designer
 Digitalisieren Sie Klänge und Musik um diese dann mit hunderten von Effekten digital zu bearbeiten. Pro Sound Designer beinhaltet neben einer zusätzlichen MIDI Software auch einen Stereo Digitalisierer.
 Komplett deutsch 299,- DM

Precision Software
 Professionelle deutsche Amiga-Software



Superpic
 PAL-Echtzeit Digitalisierer in Farbe mit integriertem Genlock. Digitalisiert von jeder Videoquelle 32768 Farben oder 64 Graustufen. Machen Sie aus Ihrem Amiga eine digitale Videomaschine. Für alle Kameras und Videorecorder mit Videoausgang.
 1795,- DM



Math Amation
 Ein Prozessor für mathematische Größen. 4 Jahre Hochschulstudium auf einer Diskette. Einheitenrechnung, Statistiken, Algebra, Graphiken, Geometrie, Rechnungen, Polynome, Matrixrechner, etc.
 Komplett deutsch 159,- DM



Logistix Professional
 Neue deutsche Version mit voller Maussteuerung für Amiga, Pull-Down Menüs und Bedienungsfeldern (ähnlich Superbase). Die einzige Tabellenkalkulation mit voller Kompatibilität zu Superbase Dateien durch Arexx Unterstützung.
 Komplett deutsch 449,- DM

Precision Software

Am Marktplatz 10
 8033 Planegg/München
 Telefon: 089/8573014
 Telefax: 089/8599970

Poststraße 25
 6200 Wiesbaden-Bierstadt
 (0 61 21) 56 00 84
 fax (0 61 21) 56 36 43



Werbung und EDV GmbH

MICROTRON
 COMPUTERPRODUKTE
 Postfach 69, Bahnhofstr. 2
 Tel. 032 872429 Fax 032 872482
 CH-2542 PIETERLEN

Bitte senden Sie mir kostenlos Ihre Gesamtinformationen

Name: _____

Straße: _____

Ort: _____

Der AMIGA ist wegen seiner Hardware-Eigenschaften geradezu prädestiniert für Software, die auf dem grafischen Sektor zu Hause ist. Ob das jetzt Zeichen-, Animations- oder Raytracing-Programme sind. Aber in dieses Genre fallen mit Sicherheit auch Spiele, die die grafischen Fähigkeiten des AMIGA mehr oder weniger ausnutzen. Doch hier liegt der Hase begraben! Leider handelt es sich oft nur um mehr oder weniger gute Umsetzungen von anderen Rechnern, die im grafischen

MASSE statt KLASSE

Bereich mit dem AMIGA nicht mithalten können. Wenn das Spiel wenigstens noch eine neue und interessante Spielidee hervorbringt, möchte ich keine größere Kritik verlauten lassen.

Doch kommen wir zum Kern der Sache: Die KICKSTART hat in jeder Ausgabe einen recht großen Spieleanteil. Jedesmal stellen wir die neuesten Spiele vor; jedoch haben sich manche Leser zurecht gewundert, warum fast alle Spiele eine positive Benotung erhalten haben. Dies hat einen einfachen Grund: Die Redaktion möchte den Lesern natürlich hauptsächlich die guten und anspruchsvollen Spiele vorstellen, die weniger guten bzw. schlechten werden meistens nicht getestet. Der Leser kann dadurch schnell desorientiert werden und glauben, die Herrn Redakteure würden alle Spiele gut testen. Weit gefehlt. Besonders in den letzten Ausgaben ist Ihnen vielleicht aufgefal-

len, daß Spiele auch einmal eine schlechte Note bekommen haben. Doch wir wollen wie bisher in der Hauptsache bessere Spiele ausführlich vorstellen und nur zu einem geringeren Anteil schlechte präsentieren.

In der Rubrik "DAS NEUESTE VOM SPIELEMARKT" bekommen des öfteren Spiele ihr Fett weg. Eine kurze Vorstellung genügt ja auch in den meisten Fällen.

In dieser Rubrik, die sich hauptsächlich neuen Produkten widmet, die z.Z. noch in der Entwicklung stehen oder erst nach Redaktionsschluß eintrafen, werden wir in Zukunft auch eine kleinere Bewertung mit einbauen, sofern das möglich ist, versteht sich. Denn oft werden neue Entwicklungen von Softwarehäusern vorgestellt, die erst in naher Zukunft käuflich zu erwerben sind.

Noch ein Wort zu Kriegsspielen: Die KICKSTART-Redaktion behält sich vor, reine Kriegsspiele zu ignorieren, also gar nicht erst zu erwähnen. Oft sind die Grenzen jedoch fließend, so daß eine Einordnung recht schwer fällt. Auch gehen die Meinungen der Redakteure über ein Spiel oft auseinander. Gegen Action- oder reine Ballerspiele haben wir jedoch nichts, sofern sich die Handlung auf fiktive, nichtmenschliche Kreaturen oder Maschinen bezieht.

Andreas Krämer

INHALT

AKTUELLES

NEWS

AMIGA-Neuigkeiten	6
Spiele-NEWS	115

SOFTWARE

GOMF 3.0

The Gurubuster	14
----------------------	----

Workshop:

ICONLAB	27
---------------	----

Photon-Video

Und sie bewegen sich doch	32
---------------------------------	----

HARDWARE

Optische Maus

Eine Maus, die sehen kann	16
---------------------------------	----

Multisync s/w

Monitortest	71
-------------------	----

A2000 Steckplatz am A500

Der Draht zur Außenwelt	79
-------------------------------	----

GRUNDLAGEN

Batch-Dateien

Teil 2: Fortgeschrittene Anwendungen	19
--	----

Hardwareprogrammierung

Teil 3: Sprites	51
-----------------------	----

Wunderland BTX

Hoch auf dem gelben Kabel	68
---------------------------------	----

Floppy-Kurs

Teil 5: Lese- und Schreibzugriffe	99
---	----

Schrifterkennung

Die Theorie	110
-------------------	-----

DRAGONS LAIR...

...ist mit Sicherheit ein Grafik-Highlight auf dem AMIGA. Was die Programmierer geleistet haben, ist schon beeindruckend. Doch sind viele an der Aufgabe, die Prinzessin aus den Klauen des Drachens zu befreien, gescheitert. Für alle geplagten AMIGA-Spieler bieten wir die Lösung aller Sequenzen. Ab Seite

62



GOMF 3.0

Die neue Version 3.0 des 'GURU-Jägers' GOMF ist diesmal mit Hardware ausgestattet. Was es damit auf sich hat und was sich gegenüber der alten Version geändert hat, verrät ein ausführlicher Test ab Seite

15

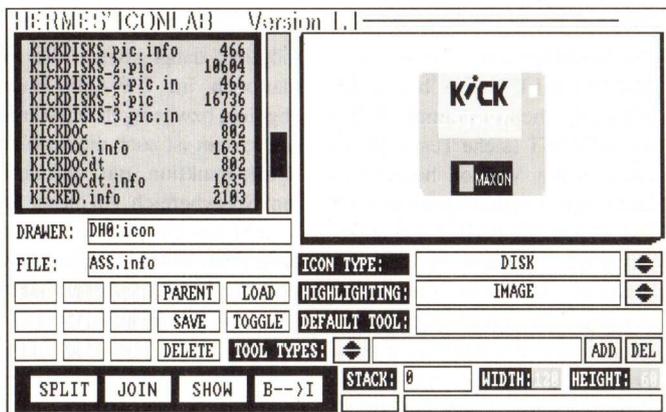


7/8 '89

WORKSHOP- ICONLAB

Jeder AMIGA-Anwender arbeitet mit der Workbench, demzufolge sind ihm Icons nicht unbekannt. Die kleinen Bilder, die Dateien oder Laufwerke auf dem Bildschirm repräsentieren, sind oft einfach und schmucklos. Mit dem Programm ICONLAB können Sie die Icons vielfältig verändern und manipulieren. Was das Programm zu leisten vermag, können Sie anhand eines Workshops selbst nachvollziehen. Ab Seite

19



KICK IT OUT - der Linkvirus-Killer

Viren erschrecken zurecht viele AMIGA-Anwender, sind es doch heimtückische kleine Programme, die mitunter ungeahnte Folgen haben können. Der IRQ-Linkvirus ist bisher nur schlecht oder mit viel Mühe zu finden und zu entfernen. Wir liefern Ihnen das Antibiotika gegen den Link-Virus.

35

PHOTON VIDEO

Die PHOTON VIDEO-Serie von MICRO ILLUSIONS soll genau das richtige für den AMIGA-Video-Amateur sein. Was sich dahinter verbirgt, wo die Schwächen und Stärken der Programme PHOTON VIDEO Cel Animator und PHOTON VIDEO Transport Controllers liegen, lesen Sie ab Seite

32

LISTINGS

KICK IT OUT!

Antilink - der Linkvirus-Killer 35

Esperantomat

Der Übersetzungskünstler 40

KICKS FÜR INSIDER

Krypt 84

Prozessordoktor 88

Vektorcheck 92

Gib Gas 94

Realtime 96

SPIELETIPS

Dragons Lair

Die Lösung 62

SPIELE

Spiele-News 115

TOP 12 117

Blood Money 118

Cybernoid II, The Duel 119

Run the Gauntlet, I Ludicrous 120

Nightdawn, Leonardo 121

Battletech, Datastorm 122

Oil Imperium, Space Pilot 89 123

The Real Gostbusters, Skweek 124

Voyager 125

RUBRIKEN

Tips&Tricks 73

Einkaufsführer 81

Inserentenverzeichnis 125

Vorschau 130

NEWS

KICKSTART 7/8'89

PHOTON PAINT 2.0



Wunderschöne Bilder lassen sich mit PHOTON PAINT realisieren.

Das amerikanische Softwarehaus MICRO ILLUSIONS bringt das HAM-Zeichenprogramm PHOTON PAINT (siehe Test 6.88) in einer neuen Version heraus. So lautet die Versionsnummer jetzt 2.0. PHOTON PAINT 2.0 unterstützt alle AMIGA-Auflösungen inklusive Overscan. Gegenüber der alten Version hat sich einiges getan. Die Brushoptionen sind erweitert und verbessert worden. Beispielsweise kann ein Brush jetzt über ein 3D-Objekt gezogen werden (Contour Mapping). Objekte können mit einer Schattenoption räumlicher gestaltet werden uvm. Animationen sind mit PHOTON PAINT ebenfalls

möglich, dabei können die Animationen im IFF-Anim-Format abgelegt bzw. eingeladen werden. Interessant ist auch die neue Air-Brush-Funktion mit definierbarem Spraybereich. An dieser Stelle sind nur ein paar Neuerungen aufgeführt, alle aufzuzählen, würde den Rahmen dieser kleinen Meldung sprengen.

Anbieter:

CSS
Auf der Warte 46
6367 Karben 1
Tel. 06039/5776

Preis: ca. 300.- DM

Speichererweiterungen in Megabittechnik für

AMIGA 500/1000

3-State Computer Technik bietet für den AMIGA 500 bzw. 1000 eine neue 512 kB-Speichererweiterung an. Die A502 RAM-Expansion wird in den Erweiterungsschacht des A500 gesteckt und ist mit Megabit-RAMS bestückt. Sie ist abschaltbar und benötigt lediglich 18% des Stroms einer herkömmlichen. Die Zugriffszeit beträgt nur 120 bzw. 100 ns. Eine Uhr ist problemlos nachrüstbar, wobei keine Lötarbeiten vorgenommen werden müssen.

Für den AMIGA 1000 ist die A1002-Speichererweiterung gedacht. Sie wird intern eingebaut und ist sowohl hardware- als auch softwaremäßig abschaltbar. Auch

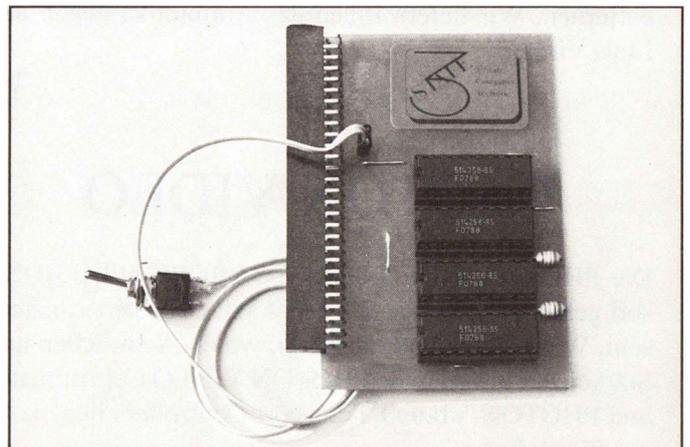
hier findet die Megabittechnik Verwendung. Der Einbau ist bei allen AMIGA 1000 mit PAL-Hauptplatine problemlos. Die Erweiterung ist autokonfigurierend durch eine Mega-Kickstart, die sich im Lieferumfang befindet.

Anbieter:

3-State Computer Technik
Schaumburgstr. 17
4350 Recklinghausen
Tel. 02361/17979

Preise:

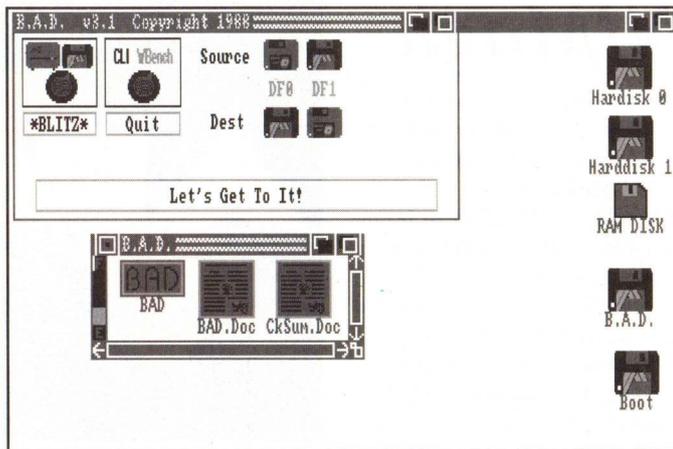
A 1002 398.- DM
A 502 259.- DM
A 562 79.- DM (Uhrenmodul)
A 502 + A 562 329.- DM



Die A 502-Speichererweiterung rüstet Ihren AMIGA 500 auf 1 MB auf, wobei die neue Megabit-Technik Verwendung findet.

B.A.D....

...hat nichts mit dem gleichnamigen Michael Jackson-Song zu tun, sondern steht für 'Blitz A Disk'. Das Utility-Programm beschleunigt den Disketten- bzw. Festplattenzugriff. Dabei handelt es sich um keinen Hardwarezusatz, vielmehr um eine reine Softwarelösung. B.A.D. reorganisiert das Speichermedium, d.h. es verkettet die Blöcke einer Datei neu, so daß sie schneller geladen werden. Im Extremfall kann es ja vorkommen, daß Dateien derart gesplittet vorliegen, daß das Laufwerk dauernd die Spur wechseln muß, was natürlich eine erhöhte Ladezeit mit sich bringt. Besonders störend ist die lange Ladezeit beim Ausgeben eines Verzeichnisses, Wartezeiten von 15 und mehr Sekunden sind keine Seltenheit.



B.A.D. reorganisiert Ihre Festplatte oder Diskette, so daß eine Geschwindigkeitssteigerung beim Laden einer Datei oder Anzeigen eines Verzeichnisses verbucht werden kann.

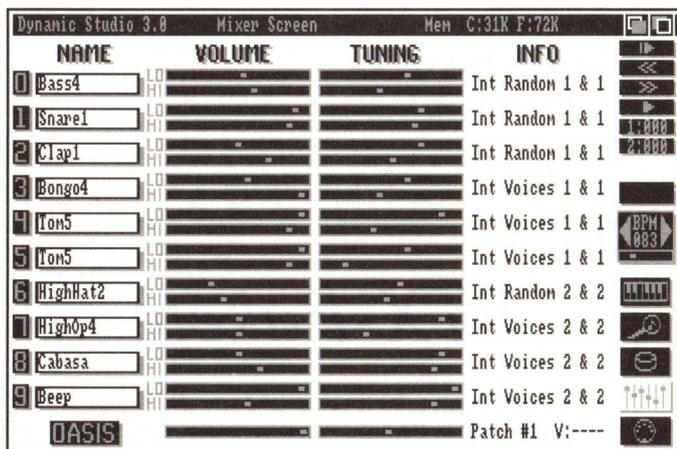
In diesem Zusammenhang haben die Programmierer von B.A.D. zwei unterschiedliche Reorganisationsmodi eingebaut. Die Optimierung kann sich entweder auf die Workbench oder auf das CLI beziehen. Im ersten Fall werden die 'info'-Dateien für das Anzei-

gen eines Disketteninhalts unter der Workbench optimiert, im zweiten Fall die globale Verzeichnisstruktur unter dem CLI. Der Anwender sollte sich vor der Optimierung im klaren sein, welchen Modus er einstellen will, bzw. welcher für ihn günstiger ist.

Die Bedienung von B.A.D. ist denkbar einfach, dazu stehen verschiedene Gadgets bereit (siehe Bild), die mit der Maus angewählt werden können. Die Geschwindigkeitssteigerung ist enorm, wurden für das Anzeigen eines Verzeichnisses bisher 20 Sekunden benötigt, sind nach der Optimierung nur noch 5 Sekunden erforderlich. Auch das Laden von Programmen geht wesentlich schneller. Allerdings braucht eine Reorganisation ihre Zeit. Für eine Diskette sind, je nach Ausnutzung, 3-5 Minuten vonnöten, bei einer Festplatte steigt die Zeit natürlich dementsprechend. Der Preis von B.A.D. liegt bei 79,-DM, mit deutscher Anleitung.

Anbieter:
CASABLANCA
Nehringskamp 9
4630 Bochum 5
Tel. 0234/411994
Preis: 79,- DM

DYNAMIC STUDIO 3.0



Das Programm DYNAMIC STUDIO 3.0 beinhaltet die Module Sequenzer, Drum Machine und Event Editor.

Sequenzer, Drum Machine und Event Editor beinhaltet die neue Version 3.0 von DYNAMIC STUDIO. Gegenüber der 'alten Version' hat sich einiges getan. Die Multitasking-Handhabung ist jetzt wesentlich verbessert worden, auch arbeitet das STUDIO jetzt mit anderen Produkten aus dem Hause NEWWAVE SOFTWARE zusammen.

Neben diesen globalen Änderungen hat man etliche neue Features implementiert, beispielsweise sind die ganzen MIDI-Optionen überarbeitet worden. Auch der Aufbau des Programms ist verändert, am rechten Bildrand kann jetzt auf Mausclick in die ver-

schiedenen Programm-Modi gesprungen werden. Die Bedienung ist gegenüber der alten Version gleichgeblieben, die Maus ist in ihrem Element. Die vielen Features des Programms fordern aber ihren Tribut, 1 MB RAM ist das Mindeste, was der musikbegeisterte AMIGA-User an freiem Speicher haben sollte.

Anbieter:
CSS
Auf der Warte 46
6367 Karben 1
Tel. 06039/5776
Preis: ca. 350,- DM

BROADCAST TITLER...

...ist ein Programm für den professionellen Videomarkt. Die maximale Auflösung eines Screens wird mit 2160*1440 Pixel angege-

ben. Spezielle Libraries für Super-Hi-Res-Fonts in verschiedenen Größen und Attributen stehen zur Verfügung. Mehrfarbige und nor-

male AMIGA-Fonts können genutzt werden. 4096 Farben, 16 Farben pro Zeile, erlauben eine detailgetreue Darstellung. Erwähnenswert ist noch der Super Smooth-Effekt. Momentan ist der BROADCAST TITLER nur in der amerikanischen NTSC-Norm zu erhalten. Auch läuft er nur auf einem NTSC-AMIGA. Bei der

europäischen PAL-Norm versagt das Programm seinen Dienst. An einer Anpassung wird allerdings gearbeitet. Des weiteren sind mindestens 1.5 MB vonnöten, allerdings sind 2 oder 3 MB anzuraten.

AEGIS ANIMagic

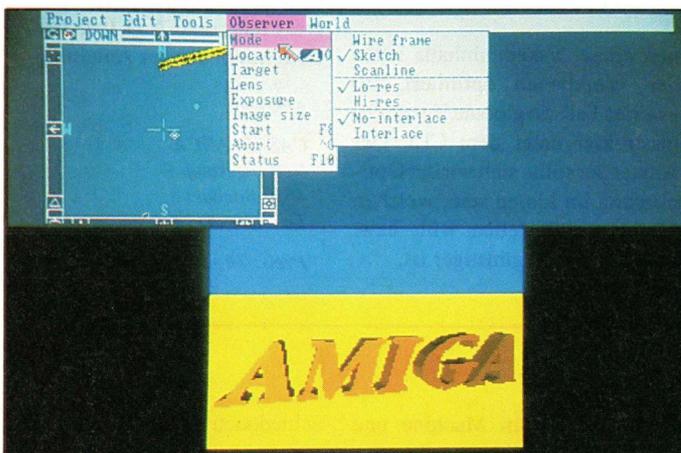
Das Softwarehaus AEGIS hat sich bereits durch einige Programme einen guten Namen gemacht, ob das jetzt das Strategiespiel PORTS OF CALL oder das Animationsprogramm Videoscape 3D ist. Mit ANIMagic kommt ein weiteres Animationsprogramm aus diesem Hause. Es handelt sich dabei um ein Programm zum Erstellen von Videovorspannen, Demofilmen und ähnlichem. Die Bedienung erfolgt mit der Maus

und ist sehr komfortabel. Dabei werden alle AMIGA-Auflösungen unterstützt, inklusive dem HAM-Modus.

Die Animationsvielfalt ist beeindruckend. Das im Bild gezeigte Aegis-Logo zoomt beispielsweise weich in den Vordergrund. Für Besitzer einer 68020-Karte mit FPU steht des weiteren eine spezielle Programmversion zur Verfügung, die die nötigen Berechnungen enorm steigern.



ANIMagic hat eine spezielle 68020-Version.

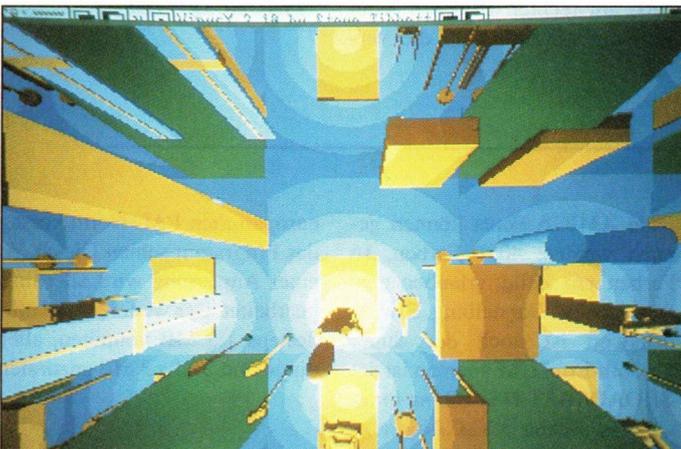


SCULPT ANIMATE 4D JUNIOR ist eine abgespeckte Version von SA 4D.

C-LIGHT II

Ein neuer Raytracer kommt aus England und nennt sich C-Light II. Eine ältere und stark eingeschränkte Version ist auf KICKSTART PD Nr. 103 zu finden. Zum Lieferumfang gehören zwei Disketten, auf denen sich der Tracer und ein Animationsprogramm zum Animieren der berechneten Bilder befinden.

Außerdem gibt es noch etliche Demos. Die Features sind vielfältig: Real-Time 3D-Animation, viele unterschiedliche Lichtquellen, spielgelnde Oberflächen, Fischaugeneffekte und etliches mehr. Die Bedienung erfolgt mit der Maus, ein komfortabler 3D-Editor ist ebenfalls implementiert.



Ein mit C-Light berechnetes Bild zeigt eindrucksvoll dessen Möglichkeiten.

SCULPT ANIMATE 4D Junior

Das komplexe Raytracing-Programm SCULPT ANIMATE 4D hat einen kleinen Bruder bekommen. Der Name ist gleichgeblieben, jedoch besitzt das neue Programm das Anhängsel 'JUNIOR'. Gegenüber der großen Version, die immerhin über 1000.- DM kostete, ist das 'JUNIOR'-Programm preiswerter und schlägt nur mit 349.- DM zu Buche. Natürlich ist es etwas abgespeckt, beispielsweise stehen nur 3 Bildermodi gegenüber 7 zur Verfügung.

Auch die Menüs Tool, Edit und Observer sind um einige Punkte

gekürzt worden. Allerdings können mit dem 'JUNIOR' ebenfalls fantastische Bilder bzw. Animationen erstellt werden. SCULPT ANIMATE 4D JUNIOR schlägt eindeutig in die Bresche, die SCULPT 3D hinterlassen hat, und ist für den semiprofessionellen Käufer konzipiert.

Anbieter:

CASABLANCA
Nehringkamp 9
4630 Bochum 5
Tel. 0234/411994

Preis: 349.- DM

BTX-Pakete

Ein interessantes BTX-Paket bietet eine Berliner Firma an. Das BTX-Paket I beinhaltet ein HAYES-kompatibles 1200 Baud-Modem (Discovery 1200C+), ein geeignetes Anschlußkabel für den AMIGA, eine PD-Diskette mit verschiedenen Terminal-Programmen und den BTX-Softwaredecoder MULTITERM Delu-

xe. Andere BTX-Pakete unterscheiden sich davon nur durch ein anderes MODEM.

Anbieter:

DFÜ-Shop
Kolonnenstr. 33
1000 Berlin 62

Preis:

BTX-Paket I 398.- DM
BTX-Paket II 569.- DM

AMIGA BASIC Profibuch

Für Theoretiker

Dieses Buch gibt Ihnen einen hervorragenden Überblick über die wichtigsten Möglichkeiten und Methoden der Systemprogrammierung unter Amiga-BASIC. Neben einer detaillierten Beschreibung und Analyse der wichtigsten Funktionen finden Sie in diesem Buch eine ausführliche Erläuterung der verschiedensten Systemunterroutinen sowie den Aufbau des Gesamtsystems.

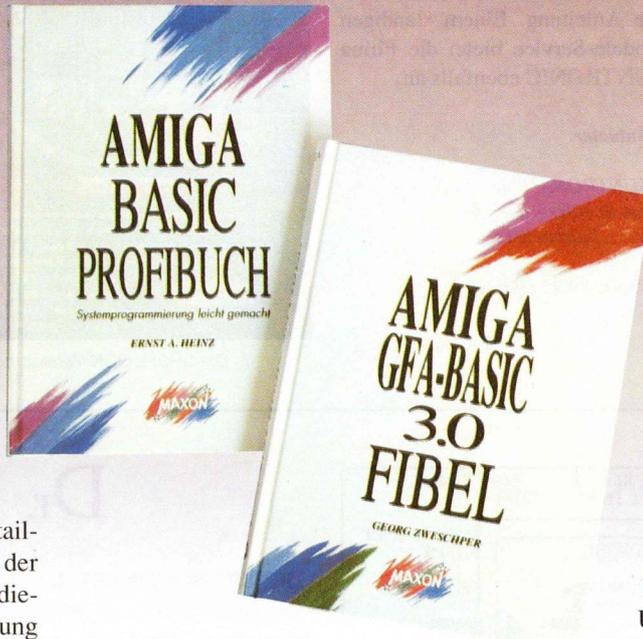
Für Praktiker

Im Hinblick auf die praktische Anwendung in eigenen Programmen dürfte die Sammlung von über 60 (!) neuen Amiga-BASIC-Unterroutinen zur Systemprogrammierung ein absoluter Leckerbissen für jeden BASIC-Programmierer sein. Mit diesen neuen Unterroutinen werden die fantastischen Möglichkeiten des Amiga nun endlich auch all denjenigen in einfacher Weise zugänglich gemacht, die selbst nicht tiefer ins System eindringen möchten.

Für jeden

Blitzschnelle und variationsreiche Textausgabe, die Benutzung beliebiger Textfonts, die Erzeugung von Pull-Down-Menüs, die Gestaltung eigener Requester sowie die komplette Fenstersteuerung werden ab sofort für jeden BASIC-Programmierer leicht realisierbar.

DM 59,90 MIT DISKETTE



AMIGA GFA-BASIC 3.0 Fibel

Alles

Editor, Variablentypen, Befehle, Funktionen und Operatoren sind vollständig aufgeführt und beschrieben. Tastaturbelegung, Füllmuster und Fehlermeldungen sowie eine Liste aller implementierten Betriebssystem-Funktionen des Amiga befinden sich im Anhang.

Dieses Buch eignet sich deshalb für alle Programmierer, die in diesen neuen und einzigartigen BASIC-Dialekt einsteigen möchten.

Ausführlich

Zu jedem Befehl wird eine genaue Beschreibung der Syntax und der Parameter

gegeben. Die detaillierten, aber kompakten Erklärungen lassen keine Unklarheiten bestehen. Zusätzliche Informationen und Hinweise machen die Lektüre komplett und das Verstehen einfach. Die Programmbeispiele zeigen den richtigen Umgang mit dieser mächtigen Programmiersprache, für den BASIC-Einsteiger wie für den Amiga-

Insider. Das Buch für jeden GFA-BASIC-Programmierer.

Alphabetisch

Wenn Sie mit den fast 400 Befehlen und Funktionen von GFA-BASIC arbeiten und die Syntax oder Bedeutung einzelner Befehle nachschlagen möchten, dann werden Sie diese Vorzüge, die Ihnen eine alphabetische Sortierung bietet, nicht missen wollen.

Befehle, Funktionen, Operatoren und Systemvariablen sind deshalb streng alphabetisch geordnet und vereinfachen das Auffinden der gewünschten Informationen erheblich, denn Sie benötigen kein Inhaltsverzeichnis und keinen Index.

DM 39,90



Bestellcoupon MAXON Computer GmbH Industriestraße 26 6236 Eschborn Tel.: 06196/481811

Name: _____
Vorname: _____
Straße: _____
Ort: _____
Unterschrift: _____

Hiermit bestelle ich:

- Exemplar(e) von "AMIGA BASIC Profibuch".
Mit Diskette für DM 59,00
- Exemplar(e) "AMIGA GFA-BASIC Fibel"
DM 39,00

Versandkosten: DM 7,50
Nachnahme zuzgl. DM 4,00 Nachnahmegebühr.

- Vorkasse
 Nachnahme

SHERLOCK V1.5

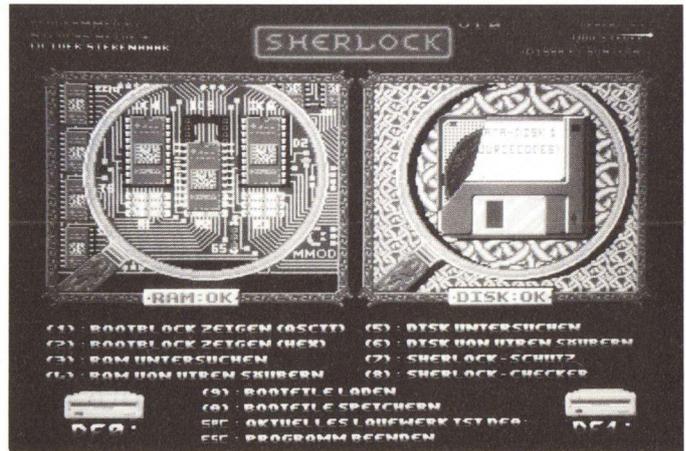
Das Antivirus-Programm SHERLOCK ist ab sofort in einer neuen Version erhältlich. Die Version 1.5 kennt über 35 Virenarten. Ist ein Bootblock nicht zu identifizieren, wird automatisch eine Warnmeldung ausgegeben. Ein SHERLOCK-CHECKER erkennt sofort ungewöhnliche Bootblöcke, ein SHERLOCK-Schutz kann Ihre

Diskette vor unbefugtem Zugriff schützen und etliches mehr. Die Programmführung ist komplett in deutscher Sprache, ebenso die Anleitung. Einem ständigen Update-Service bietet die Firma SUNTRONIC ebenfalls an.

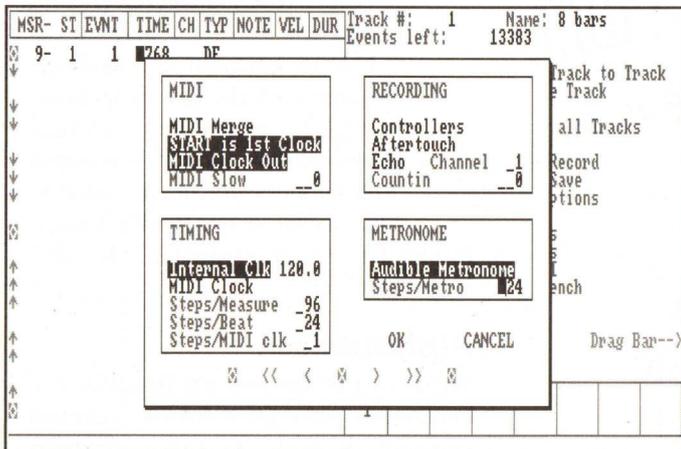
Anbieter:

SUNTRONIC
Am Schäfergraben 6
6231 Sulzbach/Ts.

Preis: 39,95 DM



Die SHERLOCK-Version 1.5 ist kompatibel zur Kickstart V1.3.



Die Aufmachung von Dr. T's MIDI Studio wirkt zwar etwas lieblos, ist aber sehr leistungsfähig.

Dr. T's...

...MUSIC SOFTWARE Inc. hat eine ganze Serie von Soundprogrammen auf dem Markt, die wohl den globalen Musikeralltag erleichtern und abdecken. Z.B. THE COPYIST, mit dem Sie Songs auf Notenbasis schreiben und ausdrucken können, wobei die Druckqualität ihresgleichen sucht, z.Z. werden der Epson FX,

HP Deskjet, HP Inkjet und der HP Laserjet Plus emuliert. Mit dem Keyboard Controlled Sequenzer können Sounds auf vielfältige Art und Weise manipuliert werden. Des weiteren ist Dr. T's MIDI und Recording Studio hauptsächlich für den Einsatz mit einem midifähigen Synthesizer gedacht.

PROFISAMPLER PLUS

Ein neuer Soundsampler wurde von OMEGA Datentechnik entwickelt. Der Anschluß erfolgt am Parallelport mit einem 40 cm langen Kabel, so daß der Sampler gut plaziert werden kann. Der Eingang paßt sich automatisch dem gelieferten Signal an, deshalb kann sowohl ein Mikrofon, ein NF-Ausgang der Stereoanlage als auch ein Lautsprecherausgang als Quelle verwendet werden. Ein Regler auf der Gehäuseoberseite dient der Lautstärkeanpassung an die verwendete Software. Der Sampler arbeitet mit jedem derzeit auf dem Markt befindlichen Sample-Programm zusammen. Durch den Einsatz eines neuen Chips ist es zudem möglich, mit einer Rate von bis zu 56 kHz zu

sampeln. Diese extrem hohe Sample-Rate wird von AUDIOMASTER II unterstützt und verspricht eine hervorragende Tonqualität. Folgende Sampler sind für A500 und A2000 lieferbar (Preise für die Plus-Version mit 56 kHz in Klammern):

- SoundSampler (mono) DM 74,- (89,-)
- SoundSampler (stereo) DM 129,- (145,-)
- ProfiSampler (mono) DM 115,- (129,-)

Hersteller:

OMEGA Datentechnik
Quellenweg 20
2900 Oldenburg
Tel.: 0441 - 7 11 09

PROFIBU & PROFAKT

Das Finanzbuchhaltungsprogramm PROFIBU ist jetzt in einer AMIGA-Version erhältlich. Es wird bereits seit mehreren Jahren auf dem PC von kleinen und mittleren Betrieben im Handel und Handwerk benutzt, gleiches gilt auch für PROFAKT. PROFIBU ist auf schnelle Dialogverarbeitung bei gleichzeitig hoher Kapazität ausgelegt, verfügt aber auch über eine Schnittstelle zur Stapelverarbeitung, so daß Daten aus anderen Programmen (z.B. aus der Fakturierung PROFAKT) eingelesen und automatisch gebucht werden können.

Das Fakturierungsprogramm

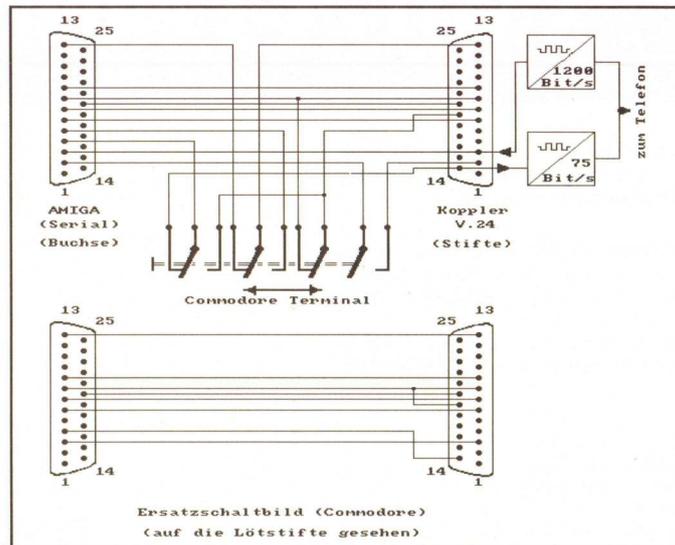
PROFAKT ist auf schnelle Dialogerstellung einer Rechnung anhand gespeicherter Adressen und Artikel ausgelegt, verfügt aber auch über die Möglichkeit, nicht gespeicherte Adressen oder Artikel bei einer Fakturierung einzugeben. PROFAKT wird in mehreren Versionen ausgeliefert, wobei eine Abhängigkeit von der Hardware besteht.

Hersteller:

MICTOTEC Sven Tietjen
Rigaweg 1
3300 Braunschweig

Umschaltkabel BTX

Es hat sich herausgestellt, daß das Commodore BTX-Programm in der endgültigen Version 2.9 nicht mehr auf der genormten Leitung Nr. 2 sendet, sondern vielmehr auf der Handshakeleitung Nr. 4. Deshalb ist das Schaltbild aus dem Heft 6, Seite 36, um zwei Drähte zu erweitern. Beachten Sie unbedingt die in die Stecker eingeprägte Ziffern. Die AMIGA-Buchse war im letzten Heft dargestellt als Ansicht von außen. Dagegen sind in diesem Heft die Lötstifte dargestellt. Beide Sichtweisen sind üblich, können einen Laien aber verwirren. Betrachten Sie deshalb die Stecker genau, damit Sie erkennen können, welche Darstellung der Verfasser jeweils gewählt hat.



Das Schaltbild des BTX-Umschaltkabels in einer erweiterten Version.

Zusätzlich wurde das Kabel um

zwei zweipolige Umschalter erweitert. In der gezeigten Schalterstellung kann das Commodore BTX-Programm problemlos gefahren werden. Im Koppler ist der Geschwindigkeitswandler abgeschaltet, da der AMIGA mit 75 Bit/s senden kann. Liegt der Schalter andersherum, können alle üblichen Terminal-Programme genutzt werden. Über den Stift 9 wird der Koppler mit Strom versorgt. Diese Leitung darf auf keinem Fall mit einer anderen Kontakt haben, da dies zur Beschädigung des AMIGA führen kann. Material: Modem, beispielsweise DATAPHON S21-23d, 150 cm zehnrädriges Kabel, 25pol. Canonstecker, 25pol. Canonbuchse, zwei zweipolige Umschalter, evtl. eine Sicherung.

A-MAX, DER MACINTOSH-EMULATOR

In letzter Minute erreichte uns ein Exemplar des neuen MacIntosh-Emulators, der soeben erschienen ist. Vertrieben wird der Emulator von ReadySoft Inc. in Kanada, die bereits mit Ihrem C64-Emulator großen Erfolg hatten. Der Emulator läuft auf allen AMIGA-Modellen und unterstützt Maus, Tastatur, Laufwerke, serielle und parallele Schnittstellen. Es können verschieden Auflösungen (640x512) und der komplette RAM-Speicher des AMIGAs genutzt werden. Außerdem

existiert eine Apple Imagewriter-Emulation für Epson-kompatible Drucker. Geliefert wird eine kleine Platine, in die die original MacIntosh-ROMs gesteckt werden müssen. Diese können aus rechtlichen Gründen nicht mitgeliefert werden, man kann sie jedoch im allgemeinen über den Ersatzteilhandel beziehen. Weiterhin gehört zum Lieferumfang ein recht ausführliches englisches Handbuch und zwei Disketten. Auf der ersten ist das Startpro-

gramm und ein Transferprogramm, mit dem man Dateien von MacIntosh- auf AMIGA-Disketten übertragen kann. Um Programme zu übertragen, benötigt man entweder ein MacIntosh-Laufwerk (kann direkt angeschlossen werden) oder einen MacIntosh. Die zweite Diskette ist im MacIntosh-Format und enthält einige Utilities. Unter anderem ein Programm zum Erstellen einer einfachen Sytemdiskette, um den Emulator zu starten und ein Programm zum Übertragen von Programmen auf das A-MAX-Format, das dann mit AMIGA-Laufwerken gelesen werden kann. Wir werden diesen interessanten Modulator für die nächste Ausgabe testen und ihn dabei auch mit seinem Original vergleichen.

Bezugsquelle:

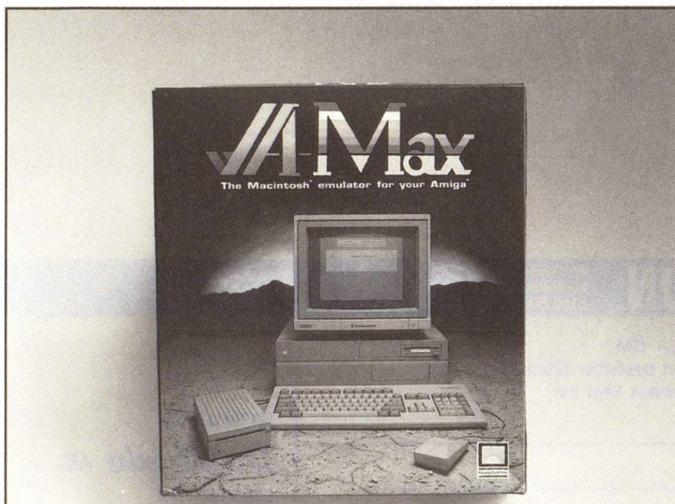
AMIGAOBERLAND
A. Koppisch
Hohenwaldstr. 26
6374 Steinbach
Tel.: 06171-7 48 05
Preis: DM 329.-

LOGISTIX PROFESSIONAL

Die Firma Precision Software hat das integrierte Softwareprodukt LOGISTIX weiterentwickelt und stellt die neue Version LOGISTIX PROFESSIONAL vor. Sie nutzt die Bedienerfreundlichkeit des AMIGA voll aus und ist kompatibel zur Tabellenkalkulation SUPERBASE. Neben der Erweiterung der Startoptionen und der Makrobefehle wurde der /Utilities-Befehl in der Weise geändert, daß direkt in LOGISTIX PROFESSIONAL AMIGADOS-Befehle ausgeführt werden können. Außerdem unterstützt die Software jetzt ARexx. Ein Update der LOGISTIX-Versionen 1.1/1.2 schlägt mit 99.- DM bzw. 49.- DM zu Buche. Der Verkaufspreis von LOGISTIX PROFESSIONAL liegt bei 449.- DM.

Hersteller:

Precision Software GmbH
Am Marktplatz 10
8033 Planegg/München
Tel. 089/8573014
Preis: 449.- DM



Jetzt für AMIGA



Dieses Siegerprogramm des Programmierwettbewerbes GOLDENE DISKETTE 87 ist die Grundlage für den neuen Vokabeltrainer aus dem Heim-Verlag:

AMIGA-Learn

Vers. 3.5

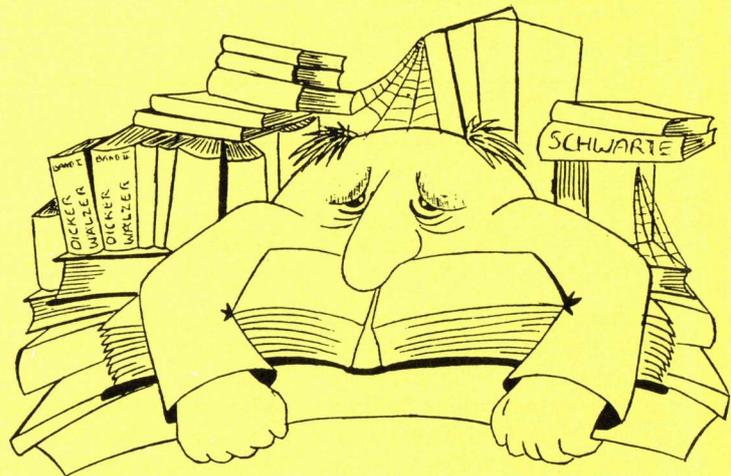
Das Urteil der Jury, bevor das Programm völlig überarbeitet und nochmals verbessert wurde:

„VOKABULA ist eines der wenigen Vokabelprogramme, das wirklich neue Eigenschaften für diese Kategorie von Lernprogrammen aufweist... Es wurde in C geschrieben und benutzt geschickt und effizient die grafische Benutzeroberfläche... Dirk Owerfeldt hat mit VOKABULA das bisher professionellste Vokabel-Programm geschrieben, das beim Wettbewerb um die GOLDENE DISKETTE eingereicht wurde.“

Und das sind die Leistungsdaten:

- ▶ bietet Spaß beim Lernen durch Grafik und Sound
- ▶ Unterstützt den vollen europäischen Zeichensatz (bequemer Zugriff durch die Maus)
- ▶ Fehlerhäufigkeit der Vokabel wird berücksichtigt
- ▶ Mehrere Bedeutungen eines Wortes werden berücksichtigt
- ▶ „Intelligente“ Auswertung der Benutzereingaben u. a. spezielle Berücksichtigung unregelmäßiger Verben (bei Eingabe von 'to go' erfolgt keine Fehlermeldung, sondern es werden die anderen Formen nachgefragt)
- ▶ Bei offensichtlicher Ähnlichkeit der Wörter wird wahlweise ein zweiter Versuch zugelassen
- ▶ Vielfältige Möglichkeiten des Lernens und der Abfrage (Deutsch-Fremdsprache, Fremdsprache-Deutsch, Multiple-Choice, lernen durch optische Rückkopplung)
- ▶ Sortieren der Vokabeln nach mehreren Parametern
- ▶ konsequente Realisierung verschiedener Lerntechniken:
 - u. a. – Karteikarten-Konzept
 - Lernen in fester Reihenfolge
 - zufällige Stichproben
 - Abfragen, bis alle Vokabeln gekannt werden
- ▶ Jederzeit Bewertung möglich, die den Lernerfolg anzeigt und mit einem Kommentar motiviert
- ▶ integriertes Lernspiel HANGMAN
- ▶ Wörterbuchfunktion, sucht Übersetzung für eine Vokabel
- ▶ Ausgabe aller Vokabeln bzw. aller falschen oder richtigen Vokabeln auf Bildschirm oder Drucker
- ▶ Trotz Einordnung der Vokabeln nach Lektionen oder Wissensgebieten ständig schneller Zugriff auf alle Vokabeln (nur durch Größe des Speichermediums begrenzt)

Im Lieferumfang befinden sich über 1600 englische Grundwortschatz-vokabeln in zwei Schwierigkeitsstufen, sowie mehrere Dateien mit wichtigen Vokabeln, etwa Wendungen und Struktur- oder Ordnungswörter.



DM 69,—

* alle Preise sind unverbindlich empfohlene Verkaufspreise

BESTELL-COUPON

an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Ich bestelle: ____ St. AMIGA-Learn á 69,— DM
zzgl. DM 5,- Versandkosten (unabhängig von bestellter Stückzahl)
 per Nachnahme Verrechnungsscheck liegt bei

Name, Vorname _____

Straße, Hausnr. _____

PLZ, Ort _____

Benutzen Sie auch die in KICKSTART vorhandene Bestellkarte.

Heim Verlag

Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 06151-56057

SCHWEIZ
DataTrade AG
Langstr. 94
CH-8021 Zürich

WELTNEUHEIT!!!

Bootfähige Eprombank für Amiga 500/1000 und A 2000

Die Hardware:

Unsere Eprombank besteht aus einem Grundmodul für 1 MB Aufnahmekapazität in 16 x 64K-Eproms. Mit einem Erweiterungsmodul kann die Eprombank auf 2 MB erweitert werden. Die A 2000 Version vereinigt beide Teile auf einer Karte.

Die Eprombank ist in einem formschönen Gehäuse untergebracht. Der Expansionsport des Rechners ist durchgeschleift, damit Sie Ihre anderen Module weiterhin benutzen können. Der Adressbereich der Eprombank ist für die Speicherbereiche hex. 200000, 400000, 600000 einstellbar, damit Sie keine Konflikte mit anderen Speichererweiterungen bekommen. Die Eprombank kann selbstverständlich auch komplett ausgeschaltet werden. Zwei Steckplätze sind für die Aufnahme von Static-RAM's ausgelegt, die über eine Batterie gepuffert werden können.

Die Anwendungsmöglichkeiten:

Auf der Eprombank können Sie Ihre meistbenutzten Anwenderprogramme abspeichern. Die mitgelieferte Steuerungssoftware erlaubt es, die Eprombank in mehrere Partitionen zu unterteilen. Jede der Partitionen ist durch die Steuerungssoftware beim Einschalten bootfähig. Sie müssen nur auswählen, welche Partition gebootet werden soll. So können Sie z.B. Ihre Festplatte bootfähig machen, Textprogramme direkt starten und vieles mehr.

Für Amiga 1000-Besitzer besteht die Möglichkeit, direkt auch Kickstarts von der Eprombank zu starten. Selbstverständlich kann auch eine voll funktionsfähige Workbench auf der Eprombank verwaltet werden.

Das Erstellen der Epromdaten:

Die auf die Eprombank zu erstellenden Daten können mit der mitgelieferten Treibersoftware brennfertig gemacht werden.

Das Booten der Eprombank:

Die Eprombank ist voll bootfähig. Das bedeutet, das Programme oder auch eine auf Eprom gespeicherte Workbench sofort nach dem Einschalten gestartet werden.

- A 500 Eprombank Basiskarte für 1MB
Bestellnummer: 6480 **248,-**
- A 500 Erweiterungskarte auf 2 MB
Bestellnummer: 6481 **139,-**
- A 2000 Eprombank 2 MB
Bestellnummer: 6620 **398,-**

Workbenchsteckmodul für Amiga 500/1000 und A 2000

Für den universellen Gebrauch haben wir eine Amiga-Eprombank mit einer kompletten Workbench 1.3 für Sie fertig gemacht. Das Modul braucht nur aufgesteckt zu werden. Die Workbench ist so nun ständig im System vorhanden und braucht nicht immer geladen zu werden. Für Kickstart 1.3-Besitzer steht diese direkt nach dem Einschalten des Amigas zur Verfügung.

- Workbenchmodul für Amiga 500
Bestellnummer: 6489 **598,-**
- Workbenchmodul A 2000
Bestellnummer: 6692 **748,-**



Amiga Grafikkarte Leonardo

- kein Flimmern im Interlacedmodus mehr
- Superbild durch Doublescandmodus
- 16 Farben
- höhere Zeilen- und Bildfrequenz
- 732 x 568 Bildpunkte
- Anschluß nur für Multi-Sync siehe Test "Amiga 4/89"

- Bildschirmspeicherkarte A 2000 ohne RAM's **398,-**
- Bildschirmspeicherkarten A 2000 komplett mit RAM's **Tagespreis**
- Bildschirmspeicherkarte 500/1000 incl. Gehäuse u. Netzteil aber o. RAM's **498,-**
- Bildschirmspeicherkarte 500/1000 incl. Gehäuse, Netzteil und RAM's **Tagespreis**

ausgereifte Ingenieurleistung • 14 Tage

- Umtauschrecht • fast
- alle IC's gesockelt • nur
- professionelle Leiterplatten • Bauteile
- namhafter Hersteller •
- mit Bedienungsanleitung



kostenlos Info anfordern!

Bestellung und Versand

ALCOMP GmbH
Glescher Weg 22
5012 Bedburg

Telefon
0 22 72/20 93

Alcomp PD-Serie Alcomp PD-Serie
Alcomp PD-Serie Alcomp PD-Serie
Alcomp PD-Serie Alcomp PD-Serie

ALCOMP PD-Serie

- mit Utilities, Demos, Animation, Sounds, Intros usw.
- 4 Blöcke a 10 Disketten je Block **60,-**
- komplett **220,-**

Weitere Alcomp PD-Serien in Vorbereitung

Keine Chance für Viren mehr!!!

Mit unserem Virenschutzmodul ist es fast unmöglich, daß Computerviren Zugang zu Ihren Disketten haben. Das Modul wird einfach als Zwischenstecker auf den Laufwerkbus gesteckt und schon ist es mit der Verbreitung von Virusprogrammen vorbei. Der Schutz wirkt auch für das interne Laufwerk.

Vollschutz mit optischer Anzeige:

In dieser Betriebsart wird grundsätzlich verhindert, daß der Bootblock beschrieben werden kann. So kann sich kein Virus auf Ihre Diskette aufkopieren. Ein Versuch, den Bootblock zu beschreiben, wird sofort optisch angezeigt.

Viruskiller

Der Viruskiller ist ein Programm, daß zum Aufspüren und zum Vernichten von Virusprogrammen geschrieben wurde. Nach der Vernichtung eines Virus kann die Diskette gegen neue Viren geschützt werden.

Damit ist das Viruskillerprogramm die optimale Ergänzung zum Viruskillermodul. Nachdem das Modul auf optische Weise einen Virus signalisiert, können Sie diesen mit dem Viruskillerprogramm auch auf der Originaldiskette unschädlich machen.

- Viruskillermodul **39,-**
Bestellnummer: 6165
- Viruskillerprogramm **35,-**
Bestellnummer: 6166
- Beide als Paket **65,-**
Bestellnummer: 6167

IDEE-SOFT-Programme
- Exzellent in ihrer Struktur - alle Programme in Deutsch -

TIPIST **AMIGA**
Der AMIGA als elektronische Schreibmaschine mit zeilenweisem Ausdruck und 15-zeiligem Bildschirm-Display. Je nach Drucker bis zu 30 verschiedene Schriften File auf Disk, Kopie-Ausgabe DM 88,-

GELD **AMIGA**
30 Routinen für den Umgang mit Geld - Anlage - Vermögensbildung - Rentensparen Rendite - Lasten - Zinsen - Kredit - Hypothek - Laufzeit - Amortisation - Raten Gleitklausel - Nominal-/Effektiv-Zinsen - Akonto bei Verzinsung - Diskontierung - Konvertierung - Komplette Tilgungspläne für alle Modi - Bild-/Druck DM 98,-

ASTROL. KOSMOGRAMM **AMIGA**
Nach Eingabe von Namen, Geburtsort (geographischer Lage) und -datum, werden errechnet: Sternzeit, Aszendent, Medium Coeli, Zodiakradianten, Positionen der Objekte im Tierkreis, Häuser nach Koch/Schaeck, Aspekte, Ausgabe einer allgemeinen Persönlichkeitsanalyse mit Partnerschaftsskala. Bild-/Druckerausgabe auf 3 DIN A4-Seiten, Horoskop-Diagramm - Erfassung aller Planeten mit Sonne, Mond und Mondknoten, Minutengenaue Ausgabe - Koordinaten-Tafel DM 78,-

BIOKURVEN **AMIGA**
Zur wissenschaftlichen Trendbestimmung der biologischen Rhythmik und des seelisch-geistig-körperlichen Gleichgewichtes - Bildschirmausgabe monatlich vor- und zurückschreitend, auf Drucker beliebig lang mit täglicher Analyse und Kennzeichnung kritischer Tage - Absolut- und Mittelwerte - Ideal für Partnervergleich Beschreibung wissenschaftl. Grundlagen - Texteditor für Zusatzdaten etc. DM 58,-

KALORIEN-POLIZEI **AMIGA**
Nach Eingabe von Körpergröße, Gewicht, Geschlecht und Arbeitsleistung erfolgen Bedarfsrechnung und Vergleich mit eingegebener tatsächlicher Ernährung nach Eiweiß, Fett und Kohlenhydraten - Ideal-, Über-/Untergewichtsbestimmung - Ausgabe der Vitalstoffe und der Gehalte - Liste der Aktivitäten und Verbräuche - Lebensmittel-Kalorientabelle - Bildschirm und/oder Drucker auf einigen DIN A4 DM 58,-

GESCHÄFT **AMIGA**
Ein Editor erstellt Formular-, Adressen-Artikel- und Dienstleistungsdateien. Im Hauptmenü wählt die Maus unter Angebot/Kostenvoranschlag, Auftrag/Bestellung, Auftragsbest., Rechnung, Lieferschein, Mahnung, Eingabe-Optionen Hand/Datei - Erfassung von je 20 Positionen/DIN A4, die über Menge, Preis, Aufschlag/Rabatt, Umsatzsteuer, Skonto durchgerechnet werden. Texteditor für Zusätze - Kein Verbund zu Lager und Buchhaltung - Schnell, übersichtlich, bedienfreundlich DM 198,-

Provisionsabrechnung **AMIGA**
Editor für Vertreter- und Kundenadressen sowie Formularitäten. Eingabe Hand/Datei, 25 Positionen/DIN A4, Provision 0,01 - 99,99%, Storno und Spesungsschriften - Durchrechnung bis zum Endbetrag mit/ohne Umsatzsteuer, schnell und genau DM 118,-

Inventur, Fibu-gerecht **AMIGA**
Kontinuierliche Bestandsverwaltung mit laufender Auswertung, Neuerfassung durch Streichen, Ändern, Hinzufügen, gruppenweise Selektion nach Code, max. 1000 Pos. /Liste, Blätter-Addition DM 118,-

Dateiverwaltungen **AMIGA**
Datenfelder von 8 Zeilen a 33 Zeichen je Datei max. 1000 - Suchcode von max. 33 Zch selektiert ab 1 Zeichen, mit jedem mehr die Zielgruppe weiter einengend - Optionen: Code, Nummer, alle, Blatt vor/zurück, Ändern (zeilenw.), Streichen, Hinzufügen Druck: 80-Zeichen-/Block-Liste, Seiten-vorschub, Etiketten, Datenfeldmaske - Gezielte Aufgaben, schnell am Ziel - Übersichtlich, schnell, bedienfreundlich -
ADRESSEN 68,- **GALERIE** 118,-
BIBLIOTHEK 118,- **LAGER** 118,-
BRIEFMARKEN 118,- **PERSONAL** 118,-
DISKOTHEK 78,- **STANNEBAUM** 118,-
EXPONATE 118,- **VIDEOTHEK** 78,-
DEFINDATA zum Selbstdefinieren der Inhalte DM 148,-

In Computer Shops oder bei uns per NN + DM 5,- oder Vorkasse + DM 3,-
Preise unverbindlich. Liste gegen adressierten Freiumschlag DIN A5
Am Schneiderhaus 7
Tel. 0 29 32 / 3 29 47 D-5760 ARNSBERG 1

GOMF 3.0 UND GOMF BUTTON



Den Guru umgangen

Seit einiger Zeit gibt es Gomf in der Version 3.0 und dazu einen ca. 75.-DM teuren Hardwarezusatz mit dem Namen "Gomf Button". Lohnt sich der Kauf von Gomf? Auf welcher Grundlage arbeitet dieses Programm? Alles Fragen, denen wir heute nachgehen.

Als ich das Ding bestellte, wurde ich auf eine Wartezeit von 14 Tagen vertröstet. So war ich sehr erstaunt, als schon nach 40 Stunden UPS mit der Lieferung vor der Tür stand. 158.- DM kostete der Gomf 3.0 mit der dazugehörigen Hardware.

Aufgaben & Ziele

Gomf, die Abkürzung für Guru - "Get Outa My Face", soll dafür sorgen, daß bei Programmabstürzen das System nicht ganz "hängen" bleibt, sondern nur der abgestürzte Task aus dem System

entfernt wird. Weiterhin sollen Windows und Screens, die zu dem abgestürzten Task gehören, geschlossen und, wenn möglich, weitere Ressourcen freigegeben werden. Diese Aufgabe wird von der Software erfüllt. Die Hardware soll es ermöglichen, über Errors "hinwegzuspringen", d.h. die Ausführung des Programmes nach dem fehlerhaften Befehl fortzusetzen und bei totalen Abstürzen ins Multitasking zurückzukommen.

Die Lösung

Die Software besteht im Prinzip aus drei Programmen, und die wären Gomf3.0, Recall und Nuke. Gomf3.0 wird, am besten in der Startup-Sequence und da möglichst früh, gestartet. Die Optionen "A" und "&" ermöglichen ein sogenanntes Low-Memory-Checking, was bedeutet, daß überprüft wird, ob die Systemvektoren des MC68000 verändert wurden, und eine Unterstützung der 680x0- Prozessoren. Gomf3.0 ist übrigens knapp 25 kB groß, paßt also auch noch bei kleineren AMIGA-Systemen ins RAM. Nuke ermöglicht es, bestimmte Tasks aus dem System zu entfernen. Die Anwendung von Nuke ist denkbar einfach: starten, den entsprechenden Task anklicken, "NUKE" anwählen, und schon ist der Task aus dem System entfernt. Recall bietet die Möglichkeit, Windows oder Screens, die von abgestürzten Programmen übriggeblieben

sind, zu entfernen. Alle Programme lassen sich sowohl vom CLI als auch vom der Workbench bedienen.

Wie arbeitet Gomf 3.0?

Gomf lenkt die Exec-Alert-, die OpenWindow-, die OpenScreen-Routine usw. auf eigene Prozeduren um. Auf diese Art und Weise ist ihm bekannt, welches Window oder welcher Screen zu welchem Programm gehören. Ein abstürzendes Programm ruft zuerst Exec auf den Plan, und, bevor die Reset-Routine angesprochen werden kann, greift Gomf 3.0 wieder ein und ermöglicht es, einen abgestürzten Task aus dem System zu entfernen. Falls man den Task entfernen läßt, kann Gomf jetzt die entsprechenden Aktionen durchführen und z.B. die zu dem Task gehörenden Windows schließen. Diese Informationen wurden aus dem disassemblierten Programm gewonnen. Weitere Informationen über den genauen Ablauf kann man beim Autor dieses Artikels erhalten.

Wie arbeitet die Hardware?

Die Hardware besteht aus einem Sockel für den Custom-Chip PAULA. In dem Sockel ist eine kleine Platine mit einem NE555 (Timer), einem LS74368 (Hex-

Buffer) und etwas Zusatzbeschaltung untergebracht. Von der Platine gehen noch zwei Leitungen weg, die zu einem Taster führen, den man irgendwo im Gehäuse unterbringen sollte. Mit etwas Nachdenken war es recht einfach, herauszufinden, was die Hardware tut. Wenn man auf den Taster drückt, wird ein NMI-Interrupt erzeugt. Dies hat zur Folge, daß eine bestimmte Routine von GOMF3.0 angesprungen wird. Diese Routine soll im Falle eines Absturzes dafür sorgen, daß man wieder in das normale Multitasking zurückkommt.

Der Einbau...

...wird in allen Einzelheiten im über 150 DIN-A5 Seiten starken Handbuch beschrieben und ist leicht durchzuführen. Es gibt je ein Kapitel, mit Fotos, für AMIGA 500, 1000 und 2000. Dieses Handbuch ist zwar ordentlich gearbeitet, aber nicht besonders stabil, denn der Ordner löste sich nach kurzem Gebrauch in zwei Teile auf. Die mitgelieferten Demos ermöglichen es, die Funktion der Hard- und Software zu testen. Bei den mitgelieferten Demos funktioniert die Hardware bestens. Leider hatte ich in den drei Monaten, in denen ich den GOMF Button im Einsatz hatte, keinen Totalabsturz, in dem der Taster von Nutzen gewesen wäre. Das gleiche Ergebnis hatten auch zwei Bekannte, die GOMF 3.0 & Hardware ebenfalls im Einsatz hatten. Einwandfrei dagegen arbeitet die Hardware in Verbindung mit der SKIP-Funktion. Diese ermöglicht es dem Anwender, über einen Fehler hinwegzuspringen. Da ein Fehler immer einem bestimmten Befehl zugeordnet werden kann, gibt es auch einen Folgebefehl. Dieser kann angesprungen werden, falls man nach dem Erscheinen des GOMF-Windows auf den Taster drückt und so einen NMI-Interrupt auslöst.

Wie arbeitet die Software?

Seit langem hatte ich die Version 1.0 im Einsatz und war sehr zufrieden mit ihr, bis - ja bis ich die Version 3.0 bekam. Denn diese erwies sich als wesentlich zuverlässiger. Da ich viel in Assembler programmiere und entsprechend viele Abstürze habe, ist der GOMF mehr als nützlich. Die Software hat sich bei mir schon lange bezahlt gemacht.

Grundlagen zu Interrupts

Der MC68000 kennt 8 verschiedene Interruptzustände (Unterbrechungszustände). Allerdings kann man davon Ebene 0, kein Interrupt, abziehen, so daß effektiv 7 verschiedene Interruptebenen übrigbleiben. Allerdings werden am AMIGA nur 6 davon benutzt.

Grundsätzlich ist es so, daß ein Interrupt mit einem höheren Level den mit einem niedrigeren Level unterbrechen kann. Mit Hilfe verschiedener Register (sowohl im MC68000 als auch in PAULA) kann man

softwaremäßig verhindern, daß der eine oder andere Interrupt ausgelöst wird. Dies gilt für alle, mit Ausnahme des sogenannten NMI-Interrupts. NMI = "Nicht maskierbarer Interrupt", auch Level 7-Interrupt genannt. Dieser unterbricht alle (!) gerade laufenden Aktionen und springt das zu ihm gehörende Programm an. Zu jedem Interrupt gehört ein Programm.

Die Start- bzw. Anfangsadressen dieser Programme sind in einer Tabelle abgelegt. Diese liegt im Speicher von \$64 bis \$7C.

Adresse	Ebene	Damit verbundene Programme
\$64	1	Software, Diskblock, serielle Ausgabe
\$68	2	CIA 8520A, Expansion-Port
\$6C	3	Copper, vertikale Austastlücke, Blitter fertig
\$70	4	Audiodaten, Kanal 0 bis 3
\$74	5	Puffer serieller Port voll, Disk Sync. erkannt
\$78	6	CIA 8520B, Expansion-Port
\$7C	7	NMI-Interrupt (von GOMF 3.0 genutzt)

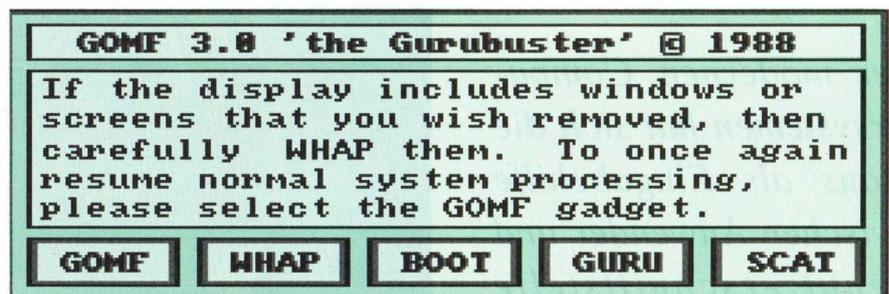


Bild: GOMF bietet viele Möglichkeiten beim Aufheben eines Gurus

Für wen?

Die Frage, für wen sich GOMF 3.0 eignet, ist einfach zu beantworten: hauptsächlich für Programmierer. Er spart bei der Programmentwicklung eine Menge Zeit, die sonst durch das laufende Hochfahren des Systems vertan wäre. Der "normale" Anwender wird für GOMF 3.0 nicht viel übrig haben, denn bei ihm sollten nicht so viele Gurus auftreten wie bei einem Entwickler. GOMF 3.0 ist nicht für jeden nützlich. Das Laufzeitsystem meines Modula-Compilers ersetzt GOMF fast völlig. Für C- oder Assembler-Programmierer ist er allerdings sehr nützlich.

Schnuppern möglich

Wer die Fähigkeiten von GOMF mal beschnuppern möchte, findet auf der

KICKSTART PD 57 die Version 1.0. GOMF 1.0 wurde in den USA einmal als eines der besten PD-Programme bezeichnet. Dies zu wissen, sollte ausreichen, um sich einmal die PD-Version zu besorgen.

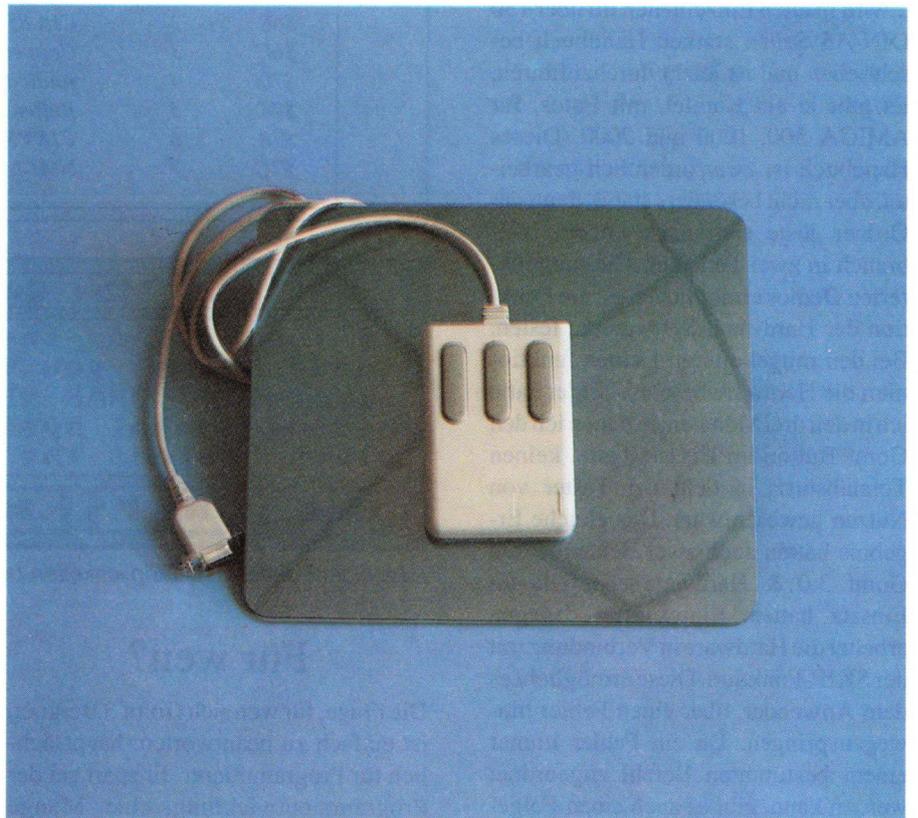
Alles in allem

Die Software ist ausgereift und für den richtigen Personenkreis empfehlenswert. Die Hardware entspricht dem leider nicht so ganz, bis auf die SKIP-Funktion erwies sie sich als kaum nützlich. Da hatte ich für knapp 80.- Mark doch mehr erwartet. Einen Preis von 70.- für die Software, die es auch ohne die Hardware zu kaufen gibt, kann man durchaus als gerechtfertigt ansehen, vor allem wenn man bedenkt, wie hoch der Prozentsatz ist, den der Programmierer bekommt.

Optische Maus

Eine Maus, die sehen kann

Bei modernen Computersystemen hat sich die Maus als Eingabehilfe zwischen Anwender und Benutzerschnittstelle eingebürgert. Mit der beim AMIGA mitgelieferten Maus werden alle Arbeiten, die auf der Workbench zu verrichten sind, erledigt.



So lassen sich durch Mauspositionierung und Klick auf eine der beiden Maustasten Disketten formatieren, Fenster verschieben, Dateien auswählen, löschen, Programme starten uvm. Das Ganze geht sehr einfach und wäre, besäße man keine Maus, eine leidige Tastaturquälerei.

Die Übertragung der Mausposition beim Bewegen der Maus geschieht mit Hilfe einer Gummikugel, die, über den Tisch gerollt, zwei Wälzchen antreibt, an deren Ende Lochräder sitzen und eine

Lichtschranke unterbrechen. Die so gewonnenen Impulse werden an eine Elektronik weitergeleitet, die die jeweilige Mausposition an den Computer sendet.

Während der vielen Einsatzkilometer wird jedes Staub- und Dreckteilchen von der Gummikugel aufgenommen und auf die innere Mechanik (Wälzchen und Lager) weitergeleitet. Mit der Zeit sammelt sich so eine beachtliche Menge Schmutz im Inneren an, der mühevoll entfernt werden muß, um die Funktion des Tierchens zu gewährleisten.

Für den AMIGA gibt es nun seit kurzem eine optische Maus, die auf den Namen Boing hört. Im Gegensatz zur original AMIGA-Maus besitzt sie keine mechanischen Teile, die zur Positionierung des Mauszeigers in Bewegung gesetzt werden müssen.

Auf ihrer Unterseite sind zwei Löcher im Gehäuse. Aus einem der beiden wird ein Lichtstrahl gesendet. Hinter dem zweiten sitzt eine Fozelle, welche eintreffende Lichtstrahlen registriert. Damit das Ganze funktioniert, muß die

R.C.T.

Requester und Menü-
Construction Tool für
den AMIGA

Wer wollte nicht schon immer sein Programm mit einer professionellen Benutzerführung versehen? Mit dem R.C.T. ist das ab sofort kein Problem mehr.

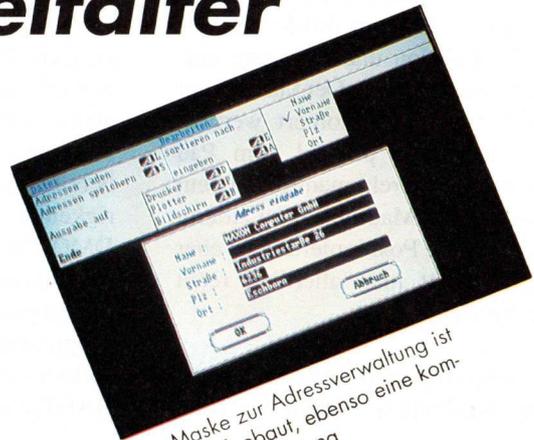
Das R.C.T. ermöglicht innerhalb kurzer Zeit die Benutzerführung und Benutzeroberfläche eigener Programme zu generieren. Die Gestaltung der Requester und Menüleisten geschieht direkt am Bildschirm. Der Anwender wird nicht mit den komplexen Gadget- oder Menüstrukturen belastet, sondern kann Sie leicht mit der Maus am Bildschirm entwerfen. Schluß mit unnötigem Ausprobieren und komplizierten Funktionsaufrufen - das R.C.T. übernimmt die Arbeit. Schluß mit langwierigem und zeitaufwendigem Programmieren - das R.C.T. erzeugt fertigen Programmcode.

Die herausragenden Eigenschaften des R.C.T.

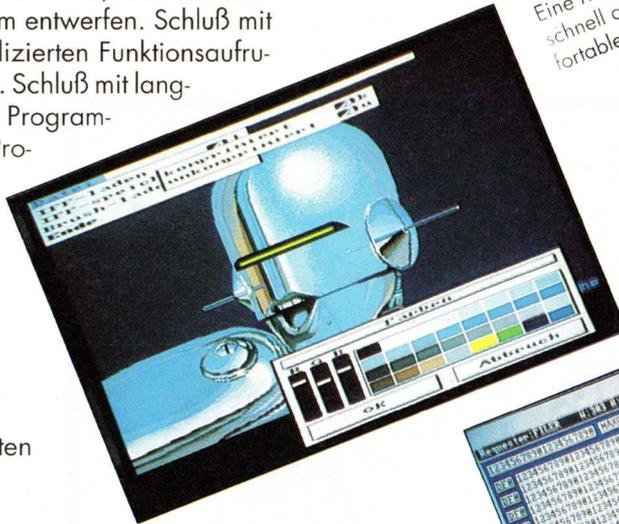
- einfaches und komfortables Konstruieren von Requestern
- Gadgets und Menüleisten
- Zahlreiche Gestaltungsmöglichkeiten wie z.B.:
 - verschiedene Zeichensätze
 - Gadget- und Requesterumrandungseditor
 - Füllmuster in Gadgets
 - Einfügen von IFF-Bildern und Konvertieren zu Auswahlknöpfen
- Arbeiten in allen Auflösungen
- Umfangreiche Funktionsbibliothek
- direktes Einbinden der Library-Funktionen
- Assembler-, C- und GFA-BASIC-Codegenerator

Mit ausführlichem, 75-seitigem Handbuch

Der Einstieg in ein neues Programmier- zeitalter



Eine Maske zur Adressverwaltung ist schnell aufgebaut, ebenso eine komfortable Menüführung



Beispiel zur Programmgestaltung: Menüs und komplette Farbeinstellbox wurden mit dem R.C.T. erzeugt.



Selbst komplexe Requesterstrukturen lassen sich mit dem R.C.T. leicht gestalten. Hier eine komfortable Filerequesterbox, die nichts zu wünschen übrig läßt



Unverb. Preisempfehlung **DM 129.-**

Bestellcoupon

Bitte senden Sie mir:

R.C.T. DM 129.-

zuzüglich Versandkosten:
Inland DM 7.50
Ausland DM 10.-

Vorkasse
 Nachnahme

Name _____

Straße _____

Ort _____

Datum _____ Unterschrift _____

Auslandsbestellungen nur gegen Vorkasse
Bei Nachnahme zuzüglich DM 4.00 Nachnahmegebühr

MAXON-Computer / Industriestr. 26 / 6236 Eschborn

Maus auf einer speziellen Unterlage bewegt werden, die natürlich im Lieferumfang enthalten ist.

Das Maus-Pad hat eine glatte, spiegelnde Oberfläche und ist mit einem dunklen, feinen Punktraster überzogen. Bewegt man die Maus über das Pad, wird der Lichtstrahl, den die Maus aussendet, unterschiedlich reflektiert, wodurch die Position und Richtung entschlüsselt werden kann. Der Mauszeiger kann sehr exakt positioniert werden, solange die Maus parallel zum Pad bewegt wird. Verdreht man sie auf dem Pad, beginnt der Mauszeiger zu zittern, und eine genaue Positionierung ist eher Zufall. Das hat man allerdings recht schnell im Griff.

Auf der mitgelieferten Diskette befindet sich ein "angepaßtes" DMOUSE-Tool,

das wohl jedem bekannt sein dürfte, da dieses Tool Public Domain ist. Es wurde hierbei allerdings lediglich der dritten Maustaste zu einer Funktion verholphen. Drückt man den mittleren Maustaster, werden die Screens geblättert, was beim herkömmlichen DMOUSE mit gleichzeitigem Drücken der linken und rechten Maustaste ebenfalls erreicht werden kann. Leider kann die mittlere Maustaste nicht über IDCMP-Messages erfragt werden, dazu muß man sich auf Systemebene einen eigenen Eventhandler schreiben. Findigen Programmierern bereitet dies aber sicher keine Schwierigkeiten, da der Source-Code zum DMOUSE-Programm ebenfalls mitgeliefert wird. Anhand dessen kann man sich abgucken, wie der Event-Handler programmiert werden muß. Am einfachsten ist es mit GFA-BASIC. Mit der BASIC-Funktion MOUSEK wird der

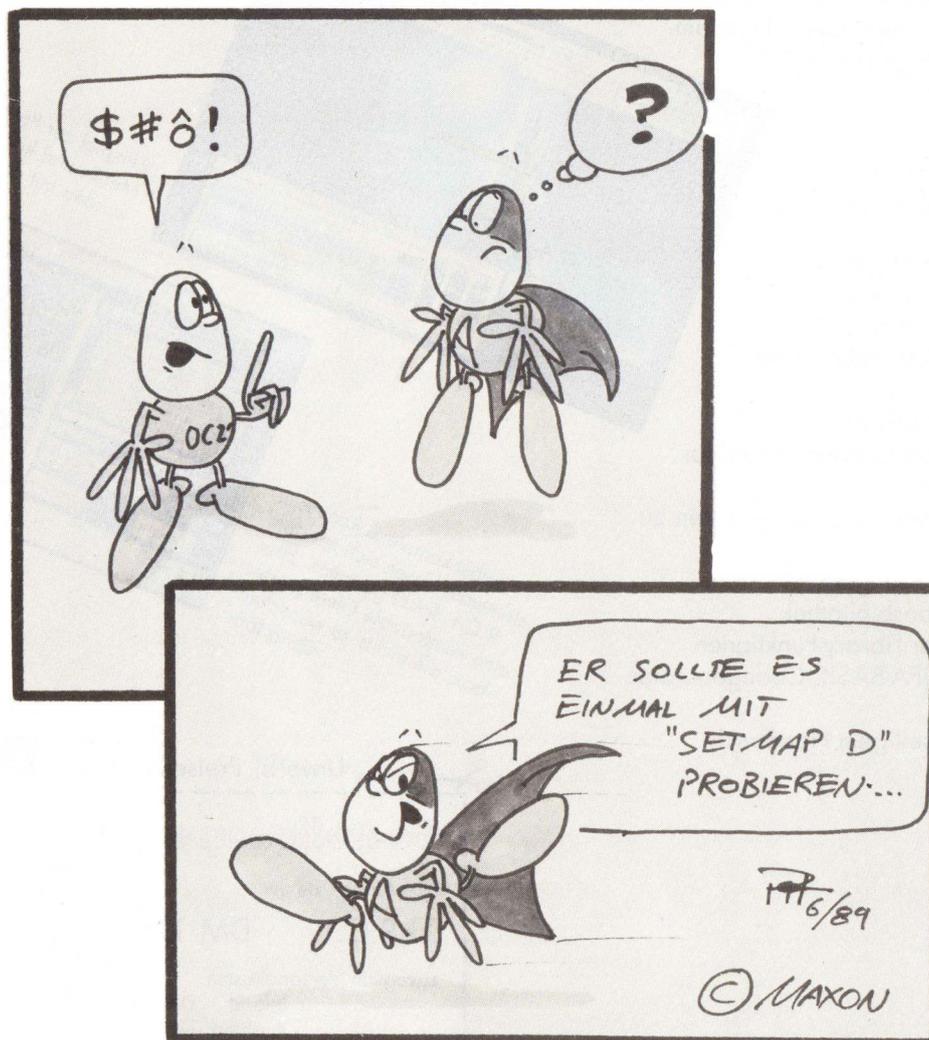
Status der Maustasten ermittelt und in einem Bitfeld mitgeteilt.

- 0 = keine Maustaste gedrückt
- 1 = linke Maustaste
- 2 = rechte Maustaste
- 4 = mittlere Maustaste

Im großen und ganzen ist das Arbeiten mit dieser Maus angenehmer als mit der AMIGA-Maus. Allerdings schreckt der Preis ein wenig ab. Er beträgt immerhin 249.- DM. ■

Anbieter:
CIT - Wiese
Maassenstr. 10
4235 Schermbeck
Tel. 02853/4099

Preis: 249.- DM



LASS' BATCHDATEIEN FÜR DICH ARBEITEN

Teil 2: Fortgeschrittene Anwendungen

Wie auch beim letzten Mal beschränken wir uns nicht nur auf Batchdateien, sondern vermitteln auch einige Grundzüge des CLIs.

Um den letzten Teil aufzufrischen, hier noch einmal eine hilfreiche Batchdatei. Ein übliches Problem ist, daß Files von einem Directory in ein anderes Directory/Device bewegt werden müssen. Zwar kann man den RENAME-Befehl zweckentfremden, solange sich die Sache auf einer Disk abspielt, doch das Wahre ist das nicht. Hier ist die richtige Lösung ein kleines Batchfile, welches die Files erst kopiert und dann löscht. Man kopiert das Batchfile nach s: und setzt das SCRIPT/SHELL-Flag; auf diese Art kann man das Batchfile wie einen normalen Befehl aufrufen.

z. B.
move df0:system/format c: OC=CLONE

Wobei OC=CLONE bedeutet, daß beim Kopieren die Option "CLONE" gesetzt wird.

```
.KEY Quelle/a,Ziel/a,OC/k,OD/k
; Batch zum "moven" von Files
; Aufruf: move df0:system/setmap c:
; Der SETMAP-Befehl wird nach c:
; kopiert und i System gelöscht
echo "Das File <Quelle> wird nach
  <Ziel> bewegt"
copy <Quelle> <Ziel> <OC> Quiet
delete <Quelle> <OD> Quiet
```

Das Abbrechen von Batchfiles; mit diesem Thema und dem Versprechen, es näher zu erläutern, haben wir den ersten Teil beendet.

Nachdem wir im Teil 1 die Grundlagen zu Batchdateien gelegt haben, wollen wir uns heute mit der mehr oder weniger fortgeschrittenen Nutzung dieses Werkzeuges beschäftigen.

Begriffserklärungen

<i>Returncode</i>	der Rückgabewert eines Programmes an das CLI. Üblicherweise Null, nur bei Fehlern wird ein Wert zurückgegeben, der ungleich Null ist. Je schwerer der Fehler, desto höher der Returncode.
<i>CTRL-?</i>	bedeutet, daß man die Taste CTRL und anschließend die nach dem "-" angegebene Taste drückt, wobei man CTRL gedrückt hält. Mit CTRL-Anweisungen werden meist Steueranweisungen an Programme übermittelt.
<i>Warnings Errors Fails</i>	Der RETURNCODE wird nach Größe in verschiedene Bereiche eingeteilt. 0 bis 9 Warnung, nichts Schwerwiegendes passiert 10 bis 19 Errors, normaler Fehler, z.B. Fehleingaben 20 bis 4294967295 schwerwiegende Fehler, z.B. Openlibrary
<i>[xxxx]</i>	bedeutet, daß man eine Option eines Befehles angeben kann, aber nicht muß. Zum Beispiel: "dir [Pfad]" bedeutet, daß man "DIR" schreiben kann aber auch "dir df0:". Beide Eingaben sind erlaubt.
<i>Resetfest</i>	meint ein Programm oder eine RAM-Disk (-ein im Arbeitsspeicher simuliertes Laufwerk, die einen Absturz des Systems genauso "überlebt" wie einen Reset über Tastatur.
<i>Bootfähig</i>	meint, daß man von diesem Device (Laufwerk) die Workbench laden kann.

ABBRECHEN & UNTERBRECHEN

Wie schon gesagt, lassen sich Stapeldateien mit CTRL-D abbrechen. Die Tastenkombination CTRL-C bricht den gerade laufenden Befehl ab, falls dieser

das zuläßt. Ein Abbruch hat zur Folge, daß der Befehl mit einer Fehlermeldung, dem sogenannten RETURNCODE, beendet wird. Falls dieser Wert größer bzw. gleich 10 ist, wird die Abarbeitung der Batchdatei abgebrochen.

Werte, die kleiner als 10 sind, werden als

Warnungen (WARNINGS) betrachtet und führen nicht zu einem Abbruch der Sequence. Ein CTRL-D sorgt dafür, daß der aktuelle Befehl noch abgearbeitet wird, und nach diesem Befehl wird die Ausführung der Datei beendet.

WAS KOMMT ZURÜCK?

Das eben erwähnte "RETURNCODE-LIMIT" oder auch Abbruchlevel kann man mit dem FAILAT-Befehl verändern. Die Anwendung dieses Befehls ist denkbar einfach:

FAILAT [Wert]

```
z.B. FAILAT 21 Returncode Limit
      auf 21
      FAILAT   ausgeben, wie
              hoch das Limit ist
```

Mit einem "FAILAT 21" kann man den Wert des Levels so setzen, daß bei allen Returncodes die kleiner als 21 sind, nicht abgebrochen wird. Wofür das nötig ist? Eine einfache Anwendung ist zum Beispiel, daß bei Tippfehlern oder bei unbekanntem Kommandos eine Batchdatei nicht abgebrochen wird. In Batch_6 ist ein Tippfehler eingebaut. Unter der Voraussetzung, daß man Batch_6 mit Fehler eingibt, wird man feststellen, daß diese Datei nicht fertig ausgeführt wird. Dies passiert aus dem Grund, daß ein unbekannter Befehl, und das ist "ECHI" im Endeffekt, einen Fehlercode von 20 verursacht. Normalerweise wird bei einem Rückgabewert von 20 der Ablauf einer Stapeldatei abgebrochen. Mit dem FAILAT-Befehl kann man verhindern, daß ein Abbruch stattfindet. Wie das geht, sieht man in Batch_7. Einfach das Fehlerlevel auf 21 setzen, und schon wird die Batchdatei bis zum bitteren Ende ausgeführt. Man muß sich das Fehler- und das Abbruchlevel als zwei Variablen vorstellen. Die eine dieser Variablen kann man selbst mit Werten beschreiben (FAILAT-Befehl), und die andere Variable wird von Befehlen bzw. Programmen mit Werten beschrieben. Das CLI (oder eine SHELL) vergleicht nun die beiden Variablen. Wenn die von einem Befehl beschriebene Variable (RETURNCODE, Rückgabewert) größer als die mit dem FAILAT-Befehl beschriebene ist, nun klar, dann wird das Batch-File abgebrochen. Auf die Returncodes und was man noch alles damit machen kann, kommen wir noch zurück

```
FAILAT           ; Ausgeben des aktuellen Abbruchlevels
ECHO "Noch richtig" ; Zeile wird ausgeführt
ECHI "Zeile mit Fehler" ; ! Tippfehler, Zeile wird nicht ausgeführt
                       ; bitte mit Fehler eingeben
ECHO "Wird nie ausgegeben" ; Zeile wird nie erreicht
```

Batch_6: Ein Schreibfehler führt zum Abbruch

```
FAILAT 21           ; Fehlerlevel auf 21 setzen
ECHO "Noch richtig" ; Zeile wird ausgeführt
ECHI "Zeile mit Fehler" ; ! Tippfehler, Zeile wird nicht ausgeführt
                       ; bitte mit Fehler eingeben
ECHO "Wird jetzt ausgegeben" ; Zeile wird jetzt ausgegeben
```

Batch_7: Nach Erhöhung der Fehlergrenze wird der Batch ausgeführt

WENN, FALLS UND ÜBERHAUPT SCHON GAR NICHT

Bisher haben wir zwei Möglichkeiten kennengelernt, um den Ablauf von Batchdateien zu beeinflussen. Zum einen die Übergabe von Parametern und zum anderen den Abbruch der Batchdatei durch die Controlsequenzen. Eine weitere Möglichkeit, auf äußere Zustände und Eingaben zu reagieren, bietet der IF-Befehl. Er macht es möglich, Eingaben abzufragen, auf Fehler zu reagieren oder übergebene Parameter auszuwerten. Im Teil 1 dieser Serie hatten wir bereits eine Startup-Sequence vorgestellt und besprochen. Diese Startup-Sequence wollen wir hier und heute erweitern. In Zukunft wird der Benutzer von seiner Startup-Sequence gefragt, ob eine RAM-Disk eingerichtet werden soll.

Der ASK-Befehl (ASK = fragen) ermöglicht es der Startup-Sequence, sich nach der Entscheidung des Benutzers zu richten. Der ASK-Befehl bietet die Möglichkeit, die Tastatur abzufragen

```
FAILAT           ; normales Fehlerlevel ausgeben
FAILAT 1         ; Verändern des Fehlerlevels auf 1
FAILAT           ; das neue Fehlerlevel ausgeben
ASK "Hallo, bitte y oder n eingeben"
ECHO "und das File wird abgearbeitet"
ECHO "Sie haben also?????? eingegeben"
```

Batch_8: Experimente mit FAILAT

und einen Text auszugeben. Falls man die Taste "y" (y steht für ya ...) betätigt, wird das Fehlerlevel auf 5 gesetzt. Der IF-Befehl des CLIs ermöglicht es nun, auf dieses Fehlerlevel zu reagieren. Bevor wir darauf eingehen, erst noch etwas zum ASK-Befehl. Dieser wird wie folgt angewendet:

ASK [Text]

```
z.B.:
ASK "Soll eine RAM-Disk eingerichtet werden? (y/n)"
```

Falls man "y" oder "Y" eingibt, wird das Fehlerlevel auf 5 gesetzt. Bei der Eingabe von "n", "N" oder RETURN erhält das Fehlerlevel den Wert 0.

Der ASK-Befehl setzt ja das Fehlerlevel auf 5, wenn man "y" bzw. "Y" eingibt. Wenn man nun das Abbruchlevel auf 1 setzt, was passiert dann? Nun, probieren Sie Batch_8 doch mal aus....

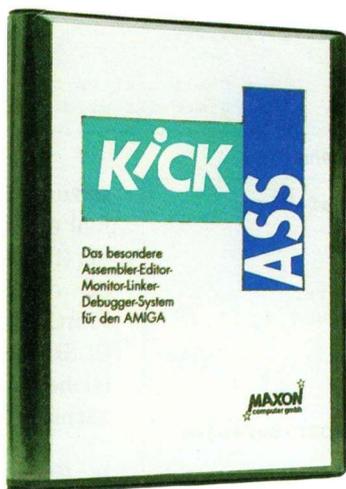
Ach ja, dabei noch eine Anmerkung zum Teil 1 dieser Serie: Ein Bekannter von mir probierte, die Batchdateien mit seinem geliebten Textverarbeitungsprogramm einzugeben, die Sache ging natürlich in die Hose.

Textverarbeitungen legen die mit ihnen erstellten Texte in einem eigenen Format ab, wie man selbst leicht mittels des TYPE-Befehls überprüfen kann. (TYPE Name_der_Datei). Das CLI erwartet aber ASCII-Dateien (ASCII = Amerikanisches Standardformat für Informationsübertragung). Das im Teil 1 erwähnte Notepad hält sich zwar auch nicht in allen Dingen an das ASCII-Format, doch sind die Abweichungen für Batchdateien unbedeutend.

Nun, was ist passiert, als Sie "Y" eingegeben haben? Ja die Stapeldatei wurde abgebrochen. Warum, das ist sicherlich auch klar: Wenn man "y" eingibt, wird die Variable, die das Fehlerlevel enthält, auf 5 gesetzt. Durch den FAILAT-Befehl haben wir festgelegt, daß bei einem Wert von größer gleich 1 (eins) abgebrochen wird.

KICK ME, AMIGA

Die Software mit dem ge- wissen KICK



KICK-ASS

Der besondere Assembler

KICK-ASS ist im Prinzip mehr als ein Assembler. Durch seinen integrierten EDITOR, MONITOR, DEBUGGER und LINKER ist KICK-ASS ein leistungsstarkes Entwicklungssystem, das die Entwicklung von Assembler-Programmen einfach, bequem und schnell macht. KICK-ASS wurde für all die entwickelt, die viel Wert auf Geschwindigkeit und Komfort legen.

Von den unzähligen Features, die KICK-ASS bietet, möchten wir nur einige nennen:

- Full-Screen-Editor mit vielen komfortablen Befehlen
- blitzschnelle Assemblierung schon während der Eingabe
- Der Quelltext wird stark komprimiert, so daß er sehr wenig Speicherplatz beansprucht.
- automatisches Einrücken der Zeilen, was gerade bei längeren Programmen eine hohe Lesbarkeit bewirkt
- automatische Syntaxüberprüfung schon bei der Eingabe
- Funktionen sind wahlweise durch Pull-Down-Menüs oder über Tastatur aufrufbar.
- Der integrierte LINKER fügt mehrere Programmmodule zusammen. Sogar die Verbindung von Assembler mit Kompilaten von höheren Programmiersprachen ist möglich.
- Der DEBUGGER dient der Programmanalyse und Fehlersuche und ist eine unentbehrliche Hilfe beim Hineinschauen in fremde Programme.

KICK-ASS DM 89.-



KICK-ED

Der universelle Editor

KICK-ED können wir allen, die gern programmieren, sei es in 'C', Assembler, Pascal oder Modula-2, wärmstens empfehlen. Aber auch, 'um 'mal schnell' einen Brief zu schreiben und auszudrucken eignet sich dieser Editor.

KICK-ED ist ein professionelles Werkzeug für Programmierer. Die Bemühung um eine möglichst einleuchtende und angenehme Benutzerführung läßt sich an vielen Details aufzeigen:

- gleichzeitige Verwaltung mehrerer Texte, die mit Hilfe der Funktionstasten angewählt werden können
- Ein Hilfs-Menü, das jederzeit aufgerufen werden kann, gibt vollständige Information über die verschiedenen Befehle.
- bequemes Laden, Speichern und Anhängen von Texten über die integrierte Datei-Auswahl-Box
- Volle Multitasking-Unterstützung des AMIGA.
- superschnelles Softscrolling

KICK-ED DM 49.-

Bestellcoupon

Bitte senden Sie mir:

KICK-ASS DM 89.-

KICK-ED DM 49.-

zuzüglich Versandkosten:

Inland DM 7.50

Ausland DM 10.-

Name _____

Straße _____

Ort _____

Datum _____

Unterschrift _____

Auslandsbestellungen nur gegen Vorkasse
Bei Nachnahme zuzüglich DM 3.80 Nachnahmegebühr

MAXON-Computer / Industriestr. 26 / 6236 Eschborn

VARIATIONEN IN IF

In Startup-Sequence 3 sieht man die Anwendung des IF-Befehles im Zusammenhang mit dem ASK-Befehl. Im Prinzip sind die Startup-Sequence und das Einrichten einer RAM-Disk aus Teil 1 bekannt. Nur die Abfrage mit dem ASK- und der folgende IF-Befehl sind neu. Doch erklären wir alles Stück für Stück und damit "back to the roots". Wir werden nun die Möglichkeiten des IF-Befehles durchsprechen und an Beispielen erläutern.

Beginnen wir mit dem in der Startup-Sequence 3 schon benutzten WARN-Parameter. Wie man aus der IF-Befehl-Tabelle entnehmen kann, reagiert der IF-Befehl in Verbindung mit dem WARN-Parameter auf den Returncode des letzten Befehls. Falls der Returncode (Rückgabewert) gleich oder größer als 5 ist, ist das Ergebnis des IF-Befehls TRUE (wahr). Es werden folglich die Befehle ausgeführt, die in den Zeilen direkt nach dem IF-Befehl stehen.

Falls die Bedingung nicht erfüllt, also FALSE (falsch, unwahr) ist, wird die Abarbeitung nach dem ENDIF-Befehl fortgesetzt. Häufig tritt aber der Wunsch auf, in einem solchem Falle eine andere Folge von Befehlen ausführen zu lassen. Dies kann man mit Hilfe des ELSE-Befehles erreichen. Unter der Voraus-

```
ECHO "Das ist meine neue Startup-Sequence 3"
PATH ram: c: s::system:utilities add ; Pfad für CLI festlegen
LOADWB ; Workbench starten
SETMAP d ; deutsche Tastaturbelegung
RESIDENT CLI L:Shell-Seg SYSTEM PURE ADD ; Shell aktivieren
MOUNT newcon: ; Starten des Consoledevices

; ***** Neuer Teil ANFANG *****
ASK "Soll eine RAM-Disk eingerichtet werden? (y/n)"
IF WARN
    ECHO " Es wird eine RAM-Disk eingerichtet"
    COPY c:dir|cd|copy|list|delete|makedir ram:c
    PATH ram:c add
ELSE
    ; Hier weiter, falls "n"
    ; eingegeben wurde
    ECHO " Es wird keine RAM-Disk eingerichtet"
ENDIF
; ***** Neuer Teil ENDE *****

NEWSHELL ; Shell starten
ENDCLI >NIL: ; Start-CLI wird geschlossen
```

Startup-Sequence 3: Anwendung des IF-Befehls

Allgemeine Syntax des IF-ENDIF-Befehles:

```
IF <Bedingung>
    Befehl_1
    Befehl_1          Werden ausgeführt,
    .                wenn die Bedingung
    .                TRUE (wahr) ist.
ENDIF
```

Allgemeine Syntax des IF-ELSE-ENDIF-Befehles:

```
IF <Bedingung>
    Befehl_1
    Befehl_1          Werden ausgeführt,
    .                wenn die Bedingung
    .                TRUE (wahr) ist.
ELSE
    Befehl_2
    Befehl_2          Werden ausgeführt
    .                wenn die Bedingung
    .                FALSE (falsch) ist.
ENDIF
```

setzung, daß der IF-Befehl das Ergebnis FALSE (falsch) liefert, werden die Befehle ausgeführt, die nach dem ELSE-Befehl kommen (siehe Listing auf der nächsten Seite).

In Batch_9 sieht man noch einmal genau das Zusammenspiel zwischen dem ASK- und dem IF-WARN-Befehl.

Bisher wurde der IF-WARN- im Zusammenhang mit dem ASK-Befehl angewandt. Nun wenden wir ihn zusammen mit dem ASSIGN-Befehl an.

Mit dieser Kombination kann man feststellen, ob ein Device (Laufwerk oder logisches Verzeichnis) dem System bekannt ist.

```
ASSIGN Devicename EXISTS
```

Falls das angegebene Laufwerk nicht vorhanden ist, wird das Fehlerlevel auf 5 (WARN) gesetzt und eine Fehlermeldung ausgegeben. Mit "IF WARN - ENDIF" kann man nun abfragen, ob das Laufwerk vorhanden ist.

z.B.

```
ASSIGN RAD: EXISTS
ASSIGN DH0: EXISTS
ASSIGN JH0: EXISTS
ASSIGN DF1: EXISTS
```

Da die Textausgabe des ASSIGN-Befehles lästig ist (einfach mal ASSIGN Blub: EXISTS eingeben), lassen wir diese im Nichts verschwinden. Die Entwickler des Betriebssystems haben schon ein NICHTS im System vorgese-

Parameter

Bedeutung

NOT	negiert das "Wahrheitsergebnis" der Bedingung, dh. es wird aus "ja" "nein" und aus "nein" "ja" gemacht.
WARN	Das Ergebnis ist "WAHR", wenn der Returncode des letzten Kommandos (siehe oben) größer oder gleich 5 ist.
ERROR	Das Ergebnis ist "WAHR", wenn der Returncode des letzten Kommandos (siehe oben) größer oder gleich 10 ist.
FAIL	Das Ergebnis ist "WAHR", wenn der Returncode des letzten Kommandos (siehe oben) größer oder gleich 20 ist.
EXISTS	dient zum Überprüfen des Vorhandenseins von Dateien und Directories. Devices (Laufwerke) können damit nicht überprüft werden.
EQ	Falls die beiden zu vergleichenden Texte gleich sind, ist das Ergebnis "WAHR".
GE	bedeutet "GREATER OR EQUAL", was zu deutsch größer oder gleich heißt. Hiermit kann festgestellt werden, ob ein Text größer oder gleich einem anderen ist. Hierbei wird nicht auf die Länge der Texte geachtet, sondern auf die ASCII-Codes der einzelnen Buchstaben.
GT	bedeutet "GREATER", zu deutsch größer. Hiermit kann überprüft werden, ob ein Text größer ist.
VAL	In Verbindung mit EQ, GE, GT bietet VAL die Möglichkeit zu numerischen Vergleichen.

hen. Dieses Nichts nennt sich NIL (Not In List) und ist als ein Device ausgelegt. Deswegen spricht man es als NIL: an. Um eine Ausgabe "verschwinden" zu lassen, lenkt man diese um (siehe ECHO-Befehl Teil 1), und zwar auf das NIL:-Device. Das sieht wie folgt aus (siehe Listing rechts).

Hiermit kann man die Ausgaben von praktisch jedem Befehl bzw. Programm ins NICHTS umleiten...(siehe Beispiel rechts).

An dieser Stelle noch einige Anmerkungen zur WShell (siehe letzte Ausgabe). Falls in einer Batchdatei eine reine Kommentarzeile auftritt, wird an dieser Stelle das Fehlerlevel auf 0 gesetzt. Beim CLI passiert das nicht. Weiterhin darf der ELSE-Befehl mit Kommentar nicht länger als 41 Zeichen sein.

RAD IM RAM

Im Laufe der Beispiele tauchte einigemale der Begriff RAD: auf. Diese Bezeichnung steht unter der Kickstart 1.3 und Workbench 1.3 für eine resetfeste und bootfähige RAM-Disk. Unter der Kickstart 1.2 kann man mit RAD: nur eine resetfeste RAM-Disk einrichten. Wir wollen hier beschreiben, wie man eine bootfähige und auch resetfeste RAM-Disk einrichtet. Beginnen wir erstmal mit einer resetfesten RAM-Disk unter der Kickstart 1.2. Hierzu müssen sich in dem Device DEVS: das File "ramdrive.device" und im Device C: der MOUNT-Befehl befinden. Im File "Mountlist", welches sich auch im Device DEVS: befindet, müssen die folgenden Einträge gemacht worden sein:

```
RAD: Device = ramdrive.device
      Unit = 0
      Flags = 0
      Surfaces = 2
      BlocksPerTrack = 11
      Reserved = 2
      Interleave = 0
      LowCyl = 0 ; HighCyl = 5
      Buffers = 5
      BufMemType = 1
#
```

Sinnvolle Änderungen kann der Benutzer bei den Einträgen RAD, HighCyl und Buffers vornehmen (siehe Tabelle).

```
ASK "Die Eingabe von (y/n) hat zur Folge, daß die IF-Bedingung ..."
      ; Textausgabe und Tastaturabfrage
IF WARN      ; Fehlerlevel auf 5?
  echo "... erfüllt ..." ; Ausgabe vom Text, falls "y"
ELSE        ; Hier weiter, falls "n"
  echo "... nicht erfüllt ..." ; Ausgabe vom Text
ENDIF      ; Abschluß der IF-Bedingung
echo "... ist..." ; Ausgabe vom Text
```

Batch_9: Zusammenspiel zwischen dem ASK- und dem IF-WARN-Befehl

```
ASSIGN >NIL: RAD: EXISTS
|      |      |      |- Option des ASSIGN-Befehls
|      |      |      |- Name des Devices
|      |      |      |- Name des Umleitungsdevices
|- aufgerufener Befehl
```

```
.KEY device      ; Parameterübergabe
.DEFAULT device DF0: ; falls kein Parameter, dann DF0:
FAILAT 21      ; Fehlerlevel auf 21T
ASSIGN >NIL: <device> EXISTS ; feststellen, ob RAD: bekannt ist
IF FAIL      ; falls Fehlercode 20
  ECHO "<device> ist da" ; Textausgabe
ELSE        ; Hier weiter, falls Fehlerlevel >20
  ECHO "<device> ist nicht da" ; Textausgabe
ENDIF      ; Ende der IF-Bedingung
```

RAD	Name der resetfesten RAM-Disk.	Hier kann ein noch nicht benutzter Device-Name eingetragen werden.
HighCyl	Max-Anzahl der Zylinder	Zahlen von 1 bis n je nach verfügbarem Speicher.
Buffers	Anzahl der 512 Byte-Blöcke, die zum Puffern verwendet werden.	Wird als PufferSpeicher verwendet, damit der Zugriff schneller geht. Sinnvoll zu ändern bei mechanischen Laufwerken.

Um RAD: zu aktivieren, muß man "MOUNT RAD:" eingeben. Falls man sich nun mit INFO-Befehl die Laufwerkliste ansieht, stellt man fest, daß RAD: dort nicht eingetragen ist, denn der Treiber für die RAM-Disk ist noch nicht geladen worden (gleiches gilt für die Festplatten). Das Laden des Treibers geschieht erst, wenn der erste Zugriff (Lock) auf das Device stattfindet. Natürlich kann man das Mounten in der Start-up-Sequence durchführen. Die Größe in Kilo-Byte des reservierten Speichers kann man nach folgender Formel ausrechnen:

$$RAD_MEM = HighCyl * BlocksPerTrack * Sectorgröße * Surfaces / 1024$$

Beim AMIGA beträgt die Sectorgröße üblicherweise 512 Bytes. Da ein Kilo-Byte 1024 Bytes hat, muß man eben durch diese 1024 teilen.

Ein RAD WOFÜR?

Als erstes kann man in RAD: ein C-Directory erzeugen. Anschließend kopiert man alle oft benutzten CLI-Befehle

nach RAD:c (z.B. mit der Batchdatei, die eine RAM-Disk einrichtet, wobei man RAM: durch RAD: ersetzt). Mit Hilfe des PATH-Befehls kann man RAD:c zu der bestehenden Pathliste hinzufügen (siehe Teil 1). Dadurch erreicht man, daß die CLI-Befehle erst in RAD:c gesucht werden. Dies erspart Zugriffszeiten und schon die Nerven. Außer dem C-Directory kann man im RAD: noch andere Directories erzeugen und ähnlich wie mit dem RAD:c verfahren. Oft benutzte Utilities finden im RAD: auch ihren Platz und stehen in "Null Komma nichts" zur Verfügung. Leute, die mit Compilern oder Assemblern arbeiten, können sich die oft benutzten Zusatzfiles (Module, Includes, Linker, u.s.w.) nach RAD: kopieren. Das gleiche ist aber auch mit der "normalen" RAM-Disk zu realisieren. Der Vorteil vom RAD: tritt erst nach einem Absturz oder Reset zutage. Denn zweitens: der Inhalt vom RAD: überlebt einen Absturz. Das heißt, man muß die Daten nur einmal kopieren und hat sie bis zum Ausschalten des Rechners zur Verfügung. Falls man der "glückliche"

Besitzer einer Kickstart 1.3 ist, bootet das System automatisch von der RAD-Disk unter der Voraussetzung, daß sich in DF0: keine Disk befindet.

Zum vernünftigen Hochfahren muß sich in RAD: eine Startup-Sequence befinden, ja, genau wie auf einer normalen Disk. Zu einer praktischen Lösung kommen wir später. Benutzer, die mit der Kickstart 1.2 arbeiten, müssen nach einem Reset oder Systemabsturz RAD: mittels des MOUNT-Befehls dem System neu anmelden.

Die Batch_11 und Batch_12 erzeugen jeweils eine resetfeste RAM-Disk.

Batch_11 ist für Kick/Work 1.3 gedacht, und Batch_12 sollte zusammen mit Kickstart 1.2 und Workbench 1.3 benutzt werden.

Batch_11 ist von der Startup-Sequence aufzurufen. Als Option kann man den Namen des Devices übergeben, in das die Dateien kopiert werden. Die vorgeschlagene Auswahl von Files kann natürlich jederzeit ergänzt werden. Nur sollten keine der Files weggelassen werden. Zum Starten einer Hard-Disk mittels einer bootfähigen RAM-Disk sollte man noch das FastFilesystem, die Mountliste, den MOUNT-Befehl und den Harddisktreiber nach RAD: kopieren. Weiterhin sollte man eine kleine Startup-Sequence nach RAD: kopieren, in der die Hard-Disk angemeldet wird. Da hier aber ausstattungsspezifische Gesichtspunkte eine Rolle spielen, gehen wir darauf nicht näher ein.

Batch_12 ist, wie schon gesagt, für Kickstart 1.2 gedacht und sollte in der Startup-Sequence mit dem EXECUTE-Befehl aufgerufen werden. Die Sequence sorgt dann dafür, daß RAD: gemountet wird, und wenn RAD: leer ist, was ja beim ersten Hochfahren der Fall ist, werden die entsprechenden Dateien nach RAD: kopiert.

WAS IST DA?

Wer Batch_12 aufmerksam gelesen hat, wird feststellen, daß eine noch nicht erklärte Eigenschaft des IF-Befehls benutzt wurde: die EXISTS-Funktion. Mittels dieser Option des IF-Befehls kann man das Vorhandensein von Dateien oder Directories überprüfen. Die Anwendung ist denkbar einfach. Man schreibt:

```
.KEY device ; Parameterübergabe
.DEFAULT device RAD: ; falls kein Parameter, dann DF0:
FAILAT 21 ; Fehlerlevel auf 21
ASSIGN >NIL: <device> EXISTS ; feststellen, ob Device bekannt ist
IF FAIL ; falls Fehlercode 20

    ECHO "<device> ist gemountet"; Textausgabe

ELSE ; Hier weiter, falls Fehlerlevel >20

    ECHO "<device> wird erzeugt" ; Textausgabe
    MOUNT <device> ; Device anmelden
    COPY c:dir|cd|copy|list|delete|makedir|assign <device>c quiet
    ; Befehle kopieren
    PATH <device>c add ; Path auf die kop. Befehle
    MAKEDIR <device>devs ; DEVS-Directory erzeugen
    COPY devs:system-configuration <device>devs
    ; Voreinstellung kopieren
    COPY l:disk-validator|ram-handler <device>l quiet
    ; wichtige Systemfiles kopieren
ENDIF ; Ende der IF-Bedingung
```

Batch_11: Resetfeste RAM-Disk unter Kick/Work 1.3

```
.KEY device ; Parameterübergabe
.DEFAULT device RAD: ; falls kein Parameter, dann DF0:
FAILAT 21 ; Fehlerlevel auf 21
MOUNT <device> ; Device anmelden
; mit Ausgabeumlenkung

IF EXISTS <device>c/dir

    ECHO "<device> ist gemountet" ; Textausgabe

ELSE ; Hier weiter, falls Fehlerlevel >20

    ECHO "<device> wird erzeugt" ; Textausgabe
    COPY c:dir|cd|copy|list|delete|makedir|assign <device>c quiet
    PATH <device>c add ; Path auf die kop. Befehle

ENDIF ; Ende der IF-Bedingung
```

Batch_12: Resetfeste RAM-Disk unter Kick 1.2/Work 1.3

```
IF EXISTS [Device:][[Directory]]
(FileName|Directory)
```

Damit die Syntax niemand versteht, haben wir sie in EBNF (EBNF = Erweiterter Backus Naur-Formalismus) abgefaßt. Aber die Beispiele werden es bestimmt klarer machen.

```
z.B.
IF EXISTS df0:Utilities/Notepad
; Notepad da?
<Befehle, die ausgeführt werden,>
<wenn das Notepad da ist>
ENDIF
```

Klar? Wenn noch irgendwelche Unklarheiten bestehen, am besten ausprobieren.

Jetzt noch ein Beispiel zum EQ- und

```
KEY Name ; Parameter einlesen

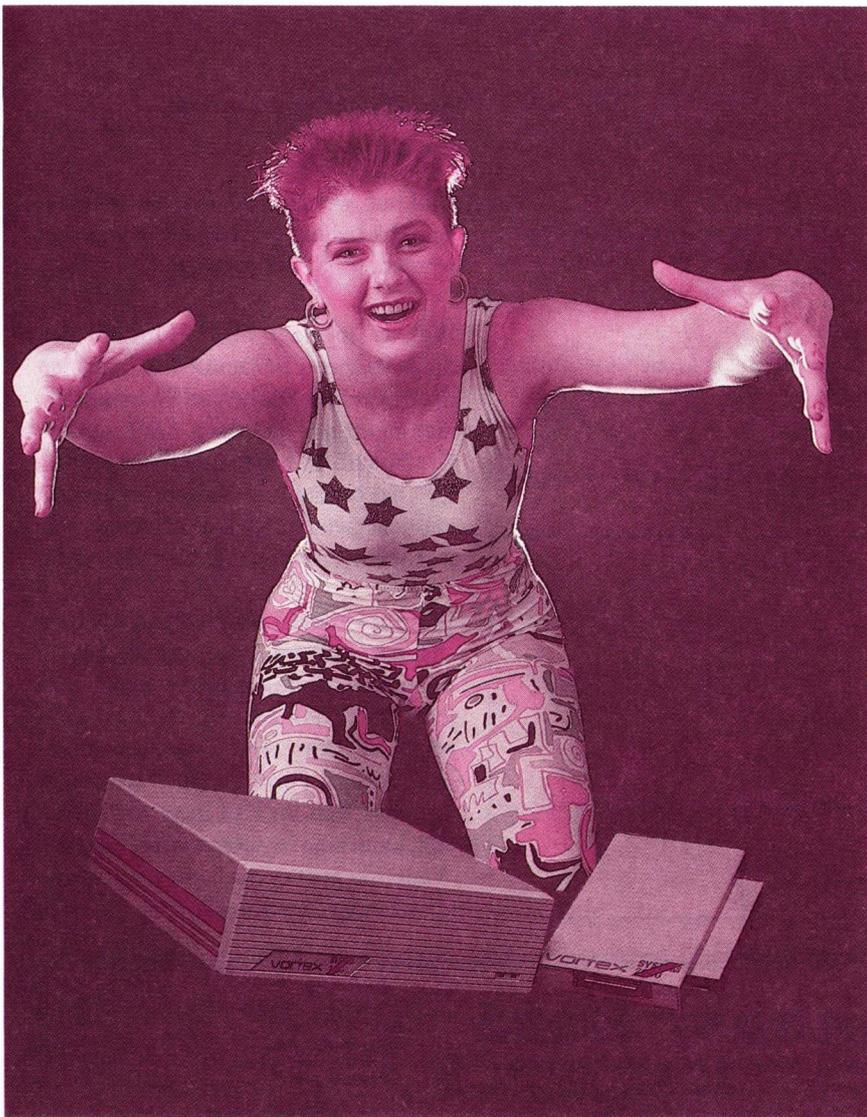
IF <Name> NOT EQ Mein_Paßwort ; Parameter vergleichen
    ECHO "Paßwort ist falsch" ; FALSCHER Parameter
    QUIT ; Ende des Batchfiles!
ELSE
    ECHO "Paßwort ist richtig" ; RICHTIGER Parameter
ENDIF ; Fertig ..
```

Batch_13: Paßwortabfrage mittels EQ und QUIT

NOT-Befehl. Mit Batch_13 ist es möglich, einen übergebenen Text mit einem festeingegebenen zu vergleichen. In Batch_13 wird auch noch ein neuer Befehl eingeführt, QUIT. Mit ihm kann man aus einem Batchfile herausspringen (siehe nächste Seite).

DAS T-DEVICE

Bei Batchdateien spielt das T-Device eine wichtige Rolle, dort werden Daten zwischengespeichert. Da auf die dortigen Dateien im Verlauf bestimmter Batchdateien häufig zugegriffen wird, die Dateien aber recht klein sind, sollte man sein t:-Device im RAM einrichten. Dies geschieht mittels des ASSIGN-



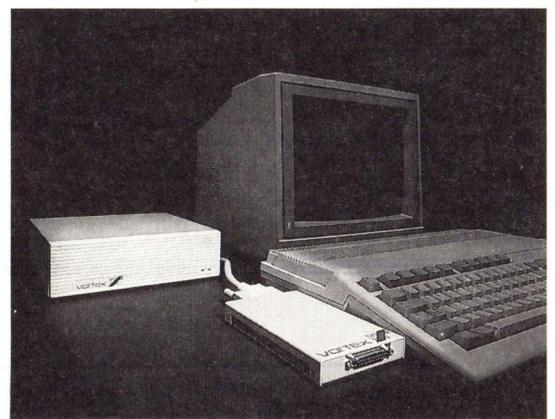
...UND
 PLÖTZLICH
 AMIGA
 GIBT'S
 FÜR DICH
 EINE
 AMIGANTISCHE
 FEST-
 PLATTE

Die neue vortex-Festplatte System 2000 für AMIGA 500 und AMIGA 1000. Serienmäßig autobootfähig ab KICKSTART 1.2. Zum sensationellen Preis von DM 998,-.*

*unverbindliche Preisempfehlung für 20 MB

Bei diesen vortex-Fachhändlern gibt's ab sofort die amigantische Festplatte:

- 1000:** Mükra, Berlin 42; Schlichting, Berlin 61; MiniSoft's DFÜ Shop, Berlin 62; Karstadt, Berlin 65.
2000: Brinkmann, 2000 Hamburg; Lavorenz, 2082 Uetersen; Ihlow & Kruse, 2100 Hamburg 90; Andreas Voss, 2280 Westerland/Sylt; MCC, 2300 Kiel; Brinkmann, 2800 Bremen; Dodenhof, 2802 Ottersberg-Posthausen.
3000: Com Data, 3000 Hannover; Ludwig Haupt, 3100 Celle; Computerstudio Frank Ueckert, 3180 Wolfsburg 11; Witte Bürotechnik, 3250 Hameln.
4000: Data Becker, 4000 Düsseldorf; Horster Computertechnik, 4300 Essen 11; Microcomputer Weber, 4400 Münster; OCB, 4422 Ahaus; Delo Computer, 4600 Dortmund; Microtec, 4800 Bielefeld; MC Byte, 4830 Gütersloh; Knicker & Wortmann, 4950 Minden.
5000: AB Computer, 5000 Köln; Büromaschinen Braun, 5000 Köln; H & G, 5300 Bonn; Kaurisch, 5500 Trier; Megabyte Computer Vertrieb, 5600 Wuppertal; Wrede Bürotechnik, 5778 Meschede.
6000: Computer Transparent, 6070 Langen; Herbig, 6120 Erbach; Computer Corner, 6330 Wetzlar; Shop 64, 6680 Neunkirchen/Saar; Phora, 6800 Mannheim-Neckarau.
7000: Schreiber Computer, 3 x in 7000 Stuttgart, 7032 Sindelfingen, 7100 Heilbronn, 7140 Ludwigsburg, 7250 Leonberg, 7530 Pforzheim; Rothfuß KG, 7142 Marbach; Wiebelt, 7730 Villingen-Schwenningen; Hettler-Data, 7890 Waldshut-Tiengen.
8000: Seemüller, 8000 München 2; Media Markt, 8000 München 45, 8070 Ingolstadt, 8300 Landshut, 8400 Regensburg, 8580 Bayreuth und 8605 Bamberg; BSC, 8000 München; Elektro Egger, 8000 München 60; COM, 8000 München 80; Promarkt, 8032 Gräfelfing; Horten, 8500 Nürnberg; Tevi-Markt, 8500 Nürnberg 80; Elektro Stender, 8640 Kronach; Top 3 Markt, 8700 Würzburg; Draga, 8730 Bad Kissingen; Bissinger, 8833 Gundelfingen.



Ja, mein lieber Freund, AMIGA, die neue vortex-Festplatte SYSTEM 2000 für AMIGA 500/1000 wird dir Beine machen: Kapazitätsmäßig habe ich jetzt ganz locker 20, 30, 40 oder 60 MB zur Verfügung. Und die mittlere Zugriffszeit, Junge, Junge: Ganze 30 ms bei der 60 MB-Version.

Der absolute Überhammer: Modernste Chip-Technologie (vortex DMA-Gate-Array) ermöglicht eine Übertragungsrate von 1 MB/sec.!!! Whowww!

vortex
 COMPUTERSYSTEME

...UND PLÖTZLICH LEISTET IHR COMPUTER MEHR

```

KEY ab

IF EXISTS t:command-00-t<$$>
  COPY t:command-00-t<$$> to t:0<ab>
ELSE
  ECHO >t:0<ab> "Test.bat command-00-t<$$> zur Zeit noch nicht vorhanden"
ENDIF

IF EXISTS t:command-01-t<$$>
  COPY t:command-01-t<$$> to t:1<ab>
ELSE
  ECHO >t:1<ab> "Test.bat command-01-t<$$> zur Zeit noch nicht vorhanden"
ENDIF

ECHO "Hier ist File 1, Tasknummer = <$$>, Param: <ab>"
EXECUTE File2 <ab>

```

```

KEY ab

IF EXISTS t:command-00-t<$$>
  COPY t:command-00-t<$$> to t:2<ab>
ELSE
  ECHO >t:2<ab> "Test1.bat command-00-t<$$> zur Zeit noch nicht vorhanden"
ENDIF

IF EXISTS t:command-01-t<$$>
  COPY t:command-01-t<$$> to t:3<ab>
ELSE
  ECHO >t:3<ab> "Test1.bat command-01-t<$$> zur Zeit noch nicht vorhanden"
ENDIF

ECHO "Hier ist File 2, Tasknummer = <$$>, Param: <ab>"
EXECUTE File1 <ab>1

```

Befehls und zwar wie folgt:

```
ASSIGN t: ram:
```

Und nun werden Dateien, die während des Abarbeitens von Batchfiles angelegt werden, im RAM angelegt.

Was da abgelegt wird? Nun, mit den Stapeldateien *File1* und *File2* kann man das feststellen. Die beiden Dateien sollte man eingeben und dann *File1* starten. Die Batchfiles haben keinen "Abschalter", das bedeutet, sie laufen, bis man sie abbricht (wie, mit ... klar doch) oder bis im System etwas an seine Grenzen stößt.

Was für Grenzen das sind und wie man sich das erklären kann, was die beiden Batchfiles so alles erzeugen; nun, wie in jeder guten Fortsetzung: NÄCHSTEN MONAT. Dort werden wir auch anfangen, ARexx einzusetzen. Verschiedenes, was sich im CLI nicht realisieren läßt, kann man in ARexx leicht machen. Weiterhin geht es noch um automatisch (!) erzeugte Batchfiles. ■

LAUFWERKE 3 1/2" Amiga Extern Formsch. Metallgehäuse helle Front, 880 KB, durchgef. Port, mit Schraubverr. abschaltbar 249,-		 <h1 style="margin: 0;">Rainbow Data</h1> <p>NEU » Wir finanzieren Ihre Anschaffung « NEU Profitieren auch Sie von unseren Finanzierungsmodellen. Nähere Angaben auf Anfrage.</p>		MONITORE Commodore 1084 S-D 598,- Atari ST SM 124 398,-	
3 1/2" Amiga Intern Komplet mit Einbausatz und Anleitung 179,-				Neu im Angebot Kickstarterumschaltung 159,- Highscorekiller f. alle 68000 3Stufenschalter und stufenlosem Geschwindigkeitsregler 59,- Akkustischer Viruswarner anzustecken an einen Laufwerksport 49,-	
5 1/4" Amiga Extern Formsch. Metallgehäuse helle Front, 40/ 80 Spur, durchgef. Port mit Schraubverr. abschaltbar 298,-		COMPUTER Amiga 500 949,- Amiga 2000 m. 2. LW.		COMPUTERLEITUNGEN Druckerkabel Amiga 500/ 1000/ 2000/ 23,00 Monitorkabel Amiga/ Scart 25,00 Emulatorkabel C 64- Amiga 19,90 Bootselector DF 0/ DF 1 oder 2-3 19,00 Mouse - Pad antistatisch, rutschfest 12,50	
3 1/2" Atari ST Extern wie oben, 720 KB, 2 x 80 Spur, eig. Netzteil 269,-		PC-XT-Karte u. 1084S 3250,-		DISKETTEN 3 1/2" No Name 2 DD ab 21,00 3 1/2" Seika 2001 2 DD 24,90 3 1/2" TDK 2 DD 28,50 5 1/4" No Name 48 TPI 7,50 5 1/4" No Name 96 TPI 12,50 5 1/4" TDK 48 TPI 16,50	
5 1/4 Atari ST Extern wie oben, 720 KB, 40/ 80 Spur, eig. Netzteil 329,-		Profex Amiga Festplatte 32 MB Autobooten 1279,- Amiga - Filecard 25 ms, 31 MB Übertragungsgeschw. ca. 460 K 1198,-		Public Domain Wir führen ca. 800 PD f. Amiga auch für Atari und IBM komp. Wir kopieren auf 2 DD Disk.	
SPEICHERERWEITERUNGEN 512 KB Ram f. Amiga 500 1.8 KB Ram f. Amiga 500 2 MB Box Extern z. Zt. auch teilbestückt mit 512 K und 1 MB für Amiga 500 u. 1000 a. A.		Weitere Angebote auf Anfrage Preisänderungen vorbehalten Erfragen Sie unsere aktuellen Tages- und Staffelpreise. Versand per Nachnahme. Rainbow Data, Am Kalkofen 32, 5603 Wülfrath, Tel.: 02058/1366		5 1/4" ab 4,- 10 ab 3,50 3 1/2" ab 5,- 10 ab 4,50	

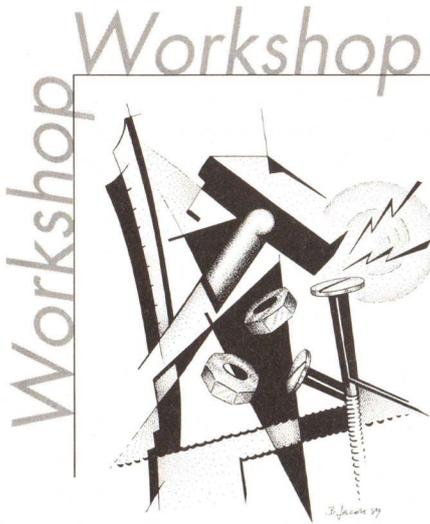
2 MB-Rambox A1000 mit Bus 2 MB bestückt	DM 949,00
Amstrad LQ 3500 Letter-Quality-24-Nad-Drucker	DM 599,00
Aztec C Dev. V3.6 mit Source Level Debugger	DM 499,00
Aztec C Professional System V3.6	DM 279,00
Digi View Gold V3.0 für A500/2000	DM 279,00
Golem 2 MB-Rambox A1000/A500	DM 999,00
Golem 3.5-Zoll-Laufwerk	DM 249,00
Vortex 20 MB-Festplatte A500/1000	DM 899,00
Vortex 60 MB-Festplatte A500/1000	DM 1499,00
MCC-Makroassembler	DM 79,00
Cambridge Lisp/Metacomco	DM 199,00
TDI-Modula Developers Version V3.01	DM 199,00
Silent Service/Thexder/Barabarian (Psyg.)	DM 49,95
Guild of Thieves, Obliterator, Balance of P.	DM 49,95

Kostenlose Prospekte auch für ST und IBM von

C W T G Joachim Tiede
 Bergstraße 13 · 71099 Reigheim
 Tel/BTX 062 98/30 98 von 17-19 Uhr

OMEGA Datentechnik	
AMIGA 2000, Monitor 1084 S	2498,- DM
AMIGA 2000, 1084 S, PC XT Karte	3199,- DM
A 2058 2MB RAM bis 8MB	1398,- DM
A 2620 Turbo 2MB 32BitRAM	3198,- DM
AMIGA 2000 AT-Karte	2295,- DM
AMIGA 500 512kByte mit Uhr	369,- DM
Animate-Turbo-Board I	ab 998,- DM
Animate-Turbo-Board III	ab 1498,- DM
3.5" Diskdrive, durchg. Bus	279,- DM
3.5" Diskdrive, intern	215,- DM
5.25" Diskdrive, durchg. Bus	349,- DM
Drucker Star LC 10 mit Anschlußka	566,- DM
Soundsampler mono & stereo	ab 74,- DM
Profisampler 40cm Anschlußkabel	115,- DM
Midi-Interface 1xIn 2xOut 1xThru	79,- DM
ALF Festplatten(A 2000) 32MB,	999,- DM
ALF Festplatten(A 2000) 65MB,	1399,- DM
3D Grafik+Animations Workstation	ab 4598,- DM
DTP Workstation	ab 3259,- DM
SW Kamera für Video-Digitizer	595,- DM
DigiView GOLD	369,- DM
Sculpt Animate 4D	1098,- DM
Audio Master II	188,- DM
2MB RAM-Chips 16 St. für A2058	998,- DM
Flicker-Fixer non Interlace Adapt.	1298,- DM
VideoSpeeder 50% besseres Bild	59,- DM

OMEGA Datentechnik Tel.: 0441/771109
 Quellenweg 20 Mo. bis Fr. ab 16 Uhr
 2900 Oldenburg Sa. von 9 Uhr bis 13 Uhr



ICONS

Erstellen und Manipulieren

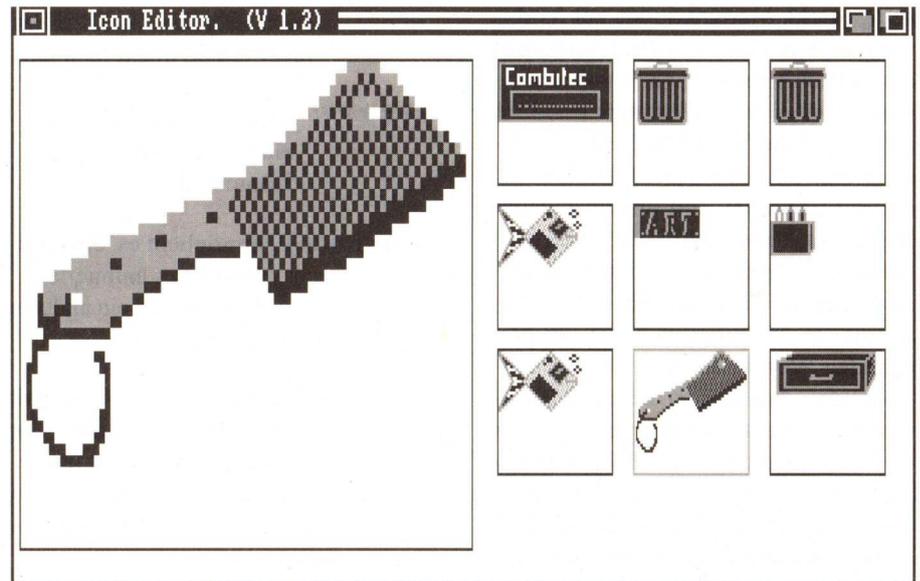
ICONED und ICONMERGE

Eine der auffallendsten Eigenschaften der AMIGA-Workbench sind ihre Icons, also die kleinen oder auch größeren Bildchen, die Disketten, Ordner und Dateien symbolisieren. Anders als beim ATARI ST, der nur Standardicons verwendet, kann das Aussehen der einzelnen Icons von jedermann geändert werden. Die bekannteste Möglichkeit ist dabei sicherlich die Verwendung des Iconeditors ICONED, der auf der Workbench- bzw. der EXTRAS-Diskette enthalten ist.

Der ICONED ist zwar mausgesteuert und hat auch einige Menüpunkte anzubieten, doch seine Bedienung und der Komfort müssen als recht mager eingestuft werden. Ein Nachteil ist auch seine Beschränkung der Icongröße und die fehlenden Optionen, Brushes bzw. IFF-Bilder in Icons umzuwandeln oder zwei Icons zusammenzufügen ("animierte" Icons). Für diese Aufgabe wird jedoch von der Workbench das Programm ICONMERGE angeboten, das sowohl Icons zusammenfügen (merge) als auch wieder trennen (split) kann.

ANIMIERTE ICONS

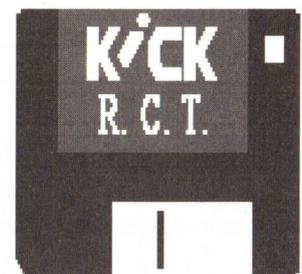
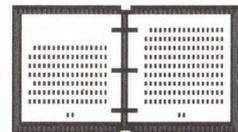
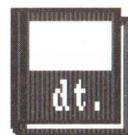
Animierte Icons, also Icons, die aus zwei Bildern bestehen, und sich deshalb beim Anklicken verändern, sind eine weitere



Der Iconeditor ICONED der Workbench

besondere Eigenschaft der AMIGA-Workbench und suchen auf anderen Computersystemen ihresgleichen. Sicherlich mag der eine oder andere sagen, daß man dies auch nicht unbedingt benötigt, aber ich finde es eine interessante

und oft auch amüsante Sache, wenn sich ein Buch oder eine Schublade beim Anklicken öffnen, oder aus einer Box ein Männchen hüpf (z.B. beim Anklicken von ICONMERGE).



Animierte Icons mit Ihren zwei Bildern

DPAINT UND ANDERE ...

Kommen wir zurück zum eigentlichen Kernproblem: woher die Bilder, sprich die Icons, nehmen? Eine der einfachsten Möglichkeiten ist, sie mit einem Malprogramm, z.B. DeluxePaint, selbst zu erstellen. Ein Malprogramm hat gegenüber dem Iconeditor ICONED den Vorteil, daß hier sehr vielfältige und leistungsstarke Malfunktionen zur Verfügung stehen. Aus dem Bild kann man dann einen Ausschnitt als Brush abspeichern, wobei man allerdings beachten sollte, daß die Workbench eine Auflösung von 640x200 bzw. 640x256 und nur 4 Farben hat. Ein umgewandeltes Bild kann deshalb von den Seitenverhältnissen und den Farben her recht seltsam wirken. Es empfiehlt sich daher, bereits im Malprogramm die entsprechenden Workbench-Verhältnisse herzustellen, damit man keine Überraschungen erlebt. Die Bearbeitung einer Grafik kann auch gut mit Hilfsprogrammen wie BUTCHER oder PIXMATE vorgenommen werden, die die verschiedenen Auflösungen umrechnen und auch eine gute Farbanpassung bzw. Farbreduzierung durchführen können. So, spätestens jetzt sollte man den gewünschten Bildausschnitt in Form eines Brushes vorliegen haben. So weit, so gut, doch wie überredet man einen Brush, sich in ein Icon zu verwandeln? Kein Problem, wenn man eines der nachfolgenden Programme zur Hand hat.

DPICON, SNAP UND SONSTIGE

Es gibt bereits eine Vielzahl von Hilfsprogrammen, die sich mit der Erstellung und Bearbeitung von Icons beschäftigen. Die meisten davon sind in unserer PD-Sammlung (KICK PD 65) zu finden, so z.B.:

DPICON: wandelt einen mit DeluxePaint erstellten Brush beliebiger Größe in ein Icon um.

ALTICON: fügt zwei Icon-Bilder zu einem "animierten" Icon zusammen.

ICONIMAGE: ermöglicht es, ein Icon-Bild eines animierten Icons gegen ein anderes auszutauschen.

SNAP: Ein beliebiger Bildschirminhalt wird als Icon abgespeichert. Die Größe des entstehenden Icons ist einstellbar, bei mehrfarbigen Screens wird eine Farbreduzierung vorgenommen.

Mit diesen Programmen lassen sich schon die wichtigsten Funktionen zur Erzeugung von Icons ausführen. Um unser Problem vom vorhergehenden Abschnitt zu lösen, einen Brush in ein Icon zu verwandeln, benötigen wir das Programm DPICON. Dieses Programm wird vom CLI aus aufgerufen und benö-

tigt als Parameter den Namen des Brushes und den des zu erzeugenden Icons. Gibt man als Pfad die RAM-Disk an, dann kann man sich das fertige Icon sogleich ansehen.

HERMES' ICONLAB

Eines der leistungsfähigsten Hilfsprogramme ist das ICONLAB von A.G. Kartsatos alias HERMES. Das Programm ist Shareware und befindet sich auf der KICK PD 159. Es vereint in sich die meisten Funktionen der anderen Programme zur Iconbearbeitung und zeichnet sich durch einen besonders hohen Bedienungskomfort aus.

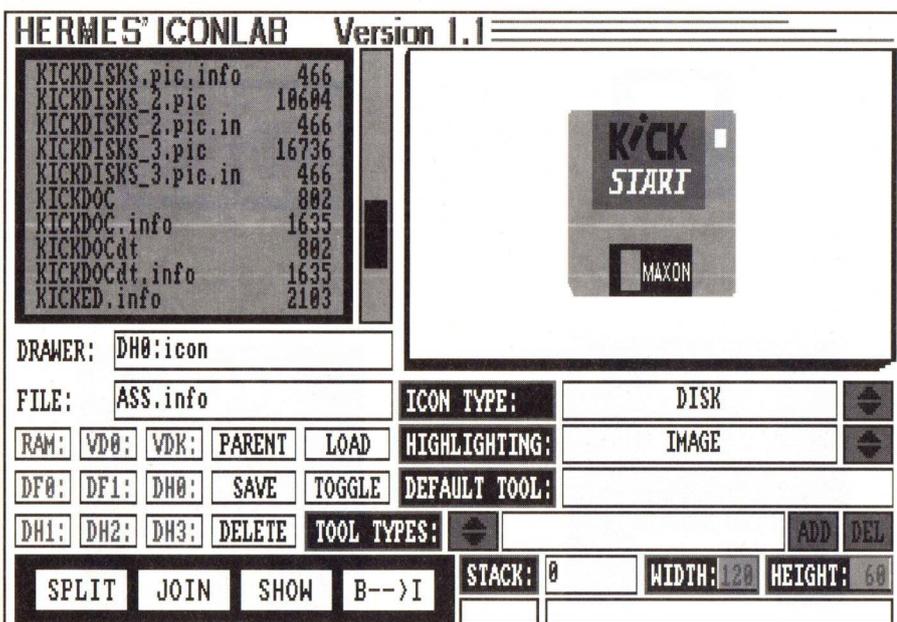
Das Programm erlaubt es, beliebige Icons und auch IFF-Bilder und Ausschnitte (z.B. Brushes) einzuladen und anzuzeigen. Außerdem können alle Iconparameter wie z.B. ICON TYPE, HIGHLIGHTING, DEFAULT TOOL, TOOL TYPES und STACK eingestellt werden.

Weitere Möglichkeiten der Version 1.1 sind:

- Icons trennen
- Icons verbinden (animieren)
- Brushes als Icons anzeigen (so wie sie aussehen, wenn man sie in den Farben der Workbench und der entsprechenden Auflösung darstellt)
- Brushes als IFF-Bilder anzeigen
- Brush als Icon speichern
- erzeugt animierte Icons
- verarbeitet Icons und Brushes in jeder Größe
- Trennen eines animierten Icons
- Einstellen aller Icon-Optionen

Dateiauswahl

Die Bedienung von ICONLAB ist sehr einfach, denn alle Aktionen werden über Knöpfe angewählt. Dies beginnt schon beim Laden einer Datei, wozu neun Laufwerksnamen vorbelegt sind. Nach Anklicken eines Laufwerks wird das Inhaltsverzeichnis im darüberliegenden Kasten angezeigt. Klickt man dort auf ein Unterverzeichnis, so wird auch dies angezeigt. Wenn man endlich beim gewünschten Dateinamen angelangt ist, wird dieser angewählt und mit "LOAD" geladen. Entsprechend einfach ist auch



Das Arbeitsfenster von ICONLAB

FROHE BOTSCHAFT VON AMIGAOBERLAND: DIESE PREISE HAT UNSER HERR KOPPISCH ABGESEGNET!

ANIMATION: NEU			
Multiplane Effects	198.00	Stand Flipper	98.00
Apprentice Disney 3-D	98.00	Forgotten World	298.00
Aegis Animate	298.00	Turbo Silver mit dem roten Handbuch	198.00
Lights, Camera, Action Como 512 KB A 500	198.00		98.00
	109.00		98.00
	349.00		49.00
			375.00
			169.00
			128.00

Unsere Hits des Monats:

Flicker Fixerplus X-Cad
CMI Prozessor Accelerator
Deluxe Paint III

1500.00
498.00
269.00

Datendisketten für Turbosilver
Digi View Gold PAL Elite

89.00

Engl. Lernprogramme
Lords of the Rising Sun
Photon Cel Animator
Photon Paint PAL 2.0

a. A. 79.00
79.00
269.00
249.00

Populous
The Director
Reflection
Blood Money

75.00
128.00
98.00
69.00

Unser absoluter SUPER-HIT:

Unsere Topangebote:

ACTIONWARE

Capone
P.O.W.
Lichtpistole

ANIMATION

3-Demon
Animation TV-Show
Caligari

Deluxe Productions
Deluxe Video II
Fanlvision

Lights, Camera, Action
Sculpt Animate 4D
Silver V.1.1

Turbo Silver V.3.0
TV-Text 3D
Video Effects 3D

Video Effects 3D
VideoScape 3D PAL 2.0
Videotitler V.1.1

Movie Setter
Comic Setter

DATENBANK

Superbase II
Superbase Professional

DIVERSES

Aegis Diga
C-64 Emulator II

FESTPLATTEN

20 MB A500
20 MB Filecard A2000
GVP Festplatten

GRAFIK

Aegis Draw 2000
Calligrapher
Deluxe Paint II PAL +

Deluxe Paint III
Deluxe Print II
Printmaster Plus

Deluxe Photo Lab
Pagerender 3-D
Digi Paint PAL

Forms in Flight II
Intro CAD
Modeler 3D

Photon Paint PAL V2.0
PixMate
Professional Draw

HURRICANE:

A-2000 Board
incl. Proz. (68020 + 68881)

ohne Proz. 2295.00
2 MByte 32 Bit 100Ns (best.) 1395.00
0 MByte 2795.00
68030 Board incl. Proz. 1395.00
ohne Proz. 2195.00
A-1000 Board 1749.00
1998.00

Evil Garden

ohne Proz.
CMI Prozessor Accelerator
mit Co Prozessor

KALKULATION

Logistix
Math-A-Mation
Maxiplan Plus

Maxiplan 500
Maxiplan Plus

MUSIK

Aegis Audiomaster II
CMI MIDI Interface
Deluxe Musik Constr. Set

ECE MIDI Interface
Future Sound II
Perfect Sound

Pro Midi Studio V.1.4
Sonix
Soundscape Sampler Mimitics

SIMULATION

Flugsimulator II
F-16 Falcon
Galileo Planetarium V.2.0

Jet
Original Jet Anleitung
Scenery Disk #7

Scenery Disk #9
Scenery Disk #11
Scenery Japan

Scenery Europa
Surgeon Operations Sim.
Universal Military Simulator

UMS Data Disk Vietnam
UMS Data Disk Civil War

SPEICHER

512 KByte A-500
8 MByte, 2 MByte Bestck

SPIELE

Andromeda Mission
Balance of Power
Balistics

Barbarian (Psygnosis)
Bards Tale
Bards Tale II

Battle Chess
Biochallenge
Blitzkrieg at the Ardennes

Blood Money
California Games
Carrier Command

Crazy Cars II
Defender of the Crown
Dragons Lair

Dungeon Master (1MB)
Elite
Empire

Evil Garden
Faery Tale Adventure

Falcon

Ferrari Formula One
Fugger
Gauntlet II

Gettysburgh
Holiday Maker
Hollywood Poker Pro

Hostages
Impossible Mission II
Interceptor

Jeanne d'Arc
Kampfgruppe
Katakis

Kings Quest I + II + III
Kristall
Leaderboard Golf + Tournam.

Leisure Suite Larry
Lords of the Rising Sun
Marble Madness

Menace (Psygnosis)
Operation Neptun
Out Run

Pac Mania
Pioneer Plaque
Populous

Ports of Call
Powerdrome
Return to Atlantis

Rocket Ranger
Roger Rabbit
R-type

Sargon III
Sentinel
Sex Vixens from Space

Shadogate
Shanghai
Sherlock

Speedball
Spherical
Summer Olympiad

Starglider II
Starry Defender
Starwars

Sub Battle Simulator
Sword of Sodan
Telewars II

Test Drive II
Terrorpods
TV Football (Cinemaware)

Wall Street Wizzard
W. Gretzky Hockey
Winter Challenge

Winter Games
Winter Olympics 88
World Games

Zak Mcracken
Zoom
Zork Trilogy

4x4 Off Road Racing
SPRACHEN
AC Basic Compiler

AC Fortran
AREXX
Aztec Source Level Debugger

85.00

72.00
59.00
58.00
89.00
85.00
55.00
75.00
69.00
65.00
55.00
50.00
95.00
79.00
72.00
65.00
79.00
60.00
54.00
70.00
59.00
62.00
65.00
75.00
85.00
79.00
69.00
65.00
75.00
67.00
60.00
69.00
79.00
65.00
55.00
73.00
69.00
45.00
79.00
73.00
78.00
85.00
79.00
55.00
69.00
69.00
85.00
49.00
59.00
59.00
69.00
39.00
115.00
60.00

89.00

439.00
319.00
339.00
189.00
189.00
530.00
398.00
59.00

419.00
199.00
159.00
589.00
339.00
225.00
495.00
145.00

89.00
89.00
185.00
119.00
90.00
59.00
79.00
50.00
349.00
55.00
149.00
75.00
138.00
49.00
95.00
129.00
97.00
79.00
65.00

298.00
598.00
349.00
1098.00
598.00
2.25
89.00
259.00
2448.00

45.00
49.00
49.00

Vergleicht die Preise, Freunde, und freut Euch mit uns!

Wir setzen Zeichen!

- ↘ = im Preis gesenkt
- ↙ = SUPERBILLIG
- ⊗ = in deutsch
- ⊛ = völlig neu

Preisänderungen und Irrtümer vorbehalten.

AMIGAOBERLAND liefert

- innerhalb von 10 Tagen (garantiert!) ((meistens))
- bei einem Mindestbestellwert von DM 50,-
- plus DM 6,- Versandkosten (sorry!)
- gegen Vorkasse oder per Nachnahme
- ins Ausland bitte nur Vorkasse (Scheck o.ä.)

**Unsere
Bestellservice-Hotline:**

0 61 71 / 7 18 46 (day & night)

Fax: 0 61 71 / 7 48 05

Unsere Hochburg:

**AMIGAOBERLAND
A. Koppisch
Hohenwaldstr. 26
D6374 Steinbach**

AMIGAOBERLAND.

Soft- und Hardware vom Feinsten. Preise vom Kleinsten.

Achtung: Bitte Liste aufbewahren. Gültigkeit 2 Monate, Neuheiten auf Anfrage!

das Speichern ("SAVE") eines geänderten oder neu erstellten Icons.

Das Icon wird im rechten oberen Kasten angezeigt, sofern es nicht zu groß ist und ein eigenes Fenster benötigt. Wenn es sich um ein animiertes Icon handelt, das sein Aussehen beim Anklicken verändert, dann kann man mit "TOGGLE" zwischen den beiden Bildern (IMAGE) umschalten.

Trennen und Zusammenfügen von Icons

Von einem solchen Icon könnte man nun das zweite Bild mit "SPLIT" abtrennen. Der umgekehrte Vorgang ist in den meisten Fällen natürlich interessanter. Dazu lädt man das erste Icon ein, klickt auf "JOIN" (Verbinden) und gibt in dem erscheinenden Feld den Namen des zweiten Icons ein. Nach Eingabe eines weiteren Namens wird das neue Icon abgespeichert.

Anzeigen von Icons und IFF-Bildern

Mit ICONLAB kann man nicht nur Icons, sondern auch jedes IFF-Bild bzw. Ausschnitte (Brush) davon anzeigen. Lädt man eine solche Datei, so erscheint eine Auswahlbox mit zwei Möglichkeiten der Darstellung.

BRUSH WINDOW	Width = 640		
	Height = 256		
SHOW BRUSH AS:	Planes = 2		
	Res. = MED		
ICON	IFF	PANIC	CANCEL

Laden und Darstellen von IFF-Bildern

1. Sie können es als ein Icon anzeigen lassen. Dies bedeutet, daß der Brush oder das IFF-Bild in den Farben und der Auflösung der Workbench dargestellt werden. Dabei ist es egal, welche Auflösung oder Größe der Brush oder das Bild haben, man sieht es so, als wäre es als Icon umgewandelt worden.

2. Sie können es als IFF-Datei ansehen. Dabei sehen Sie den Brush bzw. das IFF-Bild in der Auflösung und den Farben, mit denen sie ursprünglich erstellt wurden. Sollte der Brush eine unvollständige IFF-Datei sein, dann wird er nur in der MEDIUM-Auflösung dargestellt. Ansonsten können die IFF-Dateien in jeder Auflösung, auch Overscan, vorliegen.

Die SHOW-Funktion ist besonders dann hilfreich, wenn Sie einen Ausschnitt aus einem z.B. mit DeluxePaint erstellten Bild als Brush abspeichern und vorab sehen wollen, wie dieser als Icon in der entsprechenden Auflösung und den Farben der Workbench aussehen würde. Außerdem können sie hiermit ein geeignetes Stück aus dem Bild für Ihr späteres Icon aussuchen.

IFF-Brush in ein Icon wandeln

Ein Brush ist im allgemeinen der Ausschnitt eines IFF-Bildes und somit von den Ausmaßen her kleiner. Es können jedoch alle Größen von Brushes in Icons umgewandelt werden, auch von Overscan-Bildern, allerdings sollte man sich der Konsequenzen (nur 4 Farben, MEDRES, hoher Verbrauch an CHIPMEM !!!) bewußt sein.

Einstellen der Icon-Optionen

Die AMIGA-Workbench kennt mehrere Arten von Icons. Dabei können die Icon-Bilder gleich sein, denn es kommt allein auf die Informationen der info-Datei an, wie sich ein Icon beim Anklicken verhält und welche Aktionen ausgeführt werden. Im folgenden werden die einzelnen Optionen erklärt.

Nun folgen noch einige kurze Erläuterungen zu den Optionen der Icons, die in der englischen Originalanleitung nicht erwähnt werden. Die einzelnen, im rechten unteren Bildschirmviertel befindlichen Felder haben folgende Bedeutung:

ICON TYPE

Der ICON TYPE bestimmt die Art des Icons. Dabei werden folgende Typen unterschieden:

KICK	für Kickstartdiskette
GARBAGE	Mülleimer
PROJECT	Icon, das noch ein Programm startet, das in DEFAULT TOOL angegeben wird.
TOOL	Icon eines ausführbaren Programms
DRAWER	Schubladen/Ordner
DISK	Diskettenicon (disk.info)

Die Wahl des Icontyps ist sehr wichtig für die spätere Verwendung des Icons (s.o.).

HIGHLIGHTING

Die Option HIGHLIGHTING bestimmt das Verhalten des Icons beim Anklicken.

IMAGE	für animierte Icons
BACKFILL	komplementiert die Farben, aber nicht den Hintergrund
COMPLEMENT	komplementiert die Farben

DEFAULT TOOL

Unter dem Punkt DEFAULT TOOL werden die Programme eingetragen, die beim Anklicken aufgerufen werden sollen.

TOOL TYPES

Die TOOL TYPES enthalten weitere Informationen (Parameter) für das durch DEFAULT TOOL aufgerufene Programme. Manchen Programmen kann man Parameter zur Einstellung übergeben. Diese werden hier eingegeben. Jede Parameterzeile muß mit ADD bestätigt werden, mit den Pfeiltasten kann man weitere Zeilen erzeugen oder anzeigen (um sie z.B. zu ändern). Mit DEL kann eine Zeile wieder gelöscht werden.

STACK

Der Parameter STACK bestimmt die Größe des Stacks. Manche Programme brauchen für ihre Abarbeitung einen größeren Stack. Dies kann hier eingegeben werden.

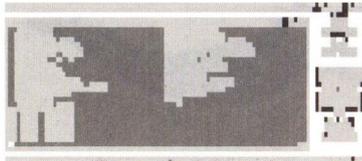
WIDTH & HEIGHT

Dieser Punkt ist bestimmt die Breite und Höhe des Icons. Diese Werte können nicht verändert werden. Sie dienen der Information über das aktuelle Icon.

Erstellung eines animierten Icons mit ICONLAB:

1. Malen eines Bildchens z.B. mit DeluxePaint III
2. Ausschneiden eines Bildausschnitts (Brush)
3. Abspeichern des Bildausschnitts
4. Laden von ICONLAB
5. Ansehen des Bildausschnitts mit SHOW
6. Wandeln des Bildausschnitts in ein Icon mit B->I

7. Wandeln eines zweiten Bildausschnitts in ein Icon
8. Erstes Icon laden
9. JOIN anwählen und Namen des zweiten Icons angeben
10. Icon-Optionen einstellen (je nach Verwendungszweck des Icons)
11. Icon abspeichern



Shareware

Das Programm ICONLAB ist Shareware, und wenn Sie es benutzen bittet Sie der Autor ("HERMES"), ihm 15 Dollar zu schicken.

Adresse: A. G. Kartsatos
8524 Caladesi Island Drive
Tampa, FL 33637, USA

Fehler

Das Programm ist nach meinen Erfahrungen sehr absturzsicher, allerdings kann es vorkommen, daß das Inhaltsverzeichnis nicht korrekt angezeigt wird. Das Programm reagiert dann meistens nicht mehr auf einen Verzeichniswechsel und zeigt immer die alte Liste. Hier hilft meines Erachtens nur ein Verlassen des Programms.

ANREGUNGEN

Sicherlich ist das Malen eines Bildes/ Bildchens nicht die einzige Möglichkeit, an ein geeignetes Icon für die heimische Workbench zu kommen. Man kann mit einem Trick auch auf schon vorhandene Icons zurückgreifen. Dazu benötigt man allerdings ein Programm, das den Bildschirm auf Diskette abspeichern kann wie z.B. GRABBIT, TURBOPRINT oder ähnliche. Dann kann man die Workbench mit dem Icon im normalen und angeklickten Zustand abspeichern und dieses Bild mit einem Malprogramm weiterbearbeiten.

Eine andere Möglichkeit bietet SNAP (KICK PD 65) an. Mit diesem Programm kann ein kompletter Bildschirm in ein Icon umgewandelt werden. Die Größe des zu erzeugenden Icons kann dabei beim Aufruf gewählt werden. Wenn der Bildschirm (Screen) mehr als 4 Farben hat, wird eine automatische Anpassung auf die Workbenchfarben durchgeführt. Damit kann man sehr schöne und vor allem aussagekräftige Icons erzeugen, deshalb kann ich diese Methode wirklich empfehlen. Sie macht zudem nur wenig Arbeit, denn alle Vorgänge werden von SNAP übernommen. Weiterbearbeiten kann man die Icons dann natürlich mit ICONLAB.

Ich hoffe, daß dieser Artikel Sie dazu ermutigt, eigene Icons zu erstellen. Mit selbstgestellten Icons kann man sich eine individuelle aussehende Workbench gestalten und den eigenen Programmen charakteristische Bilder geben, die gleich auf Ihren Zweck hinweisen.

Btx/Vtx-Manager

Btx/Vtx. Jetzt auf dem Amiga!

Die neue Welt der Telekommunikation läßt sich mit dem *Btx/Vtx Manager* komfortabel nun auch auf dem Amiga handhaben. Informationen über dieses „Fenster“ zur qualifizierten, maßgeschneiderten Information senden wir Ihnen auf Anfrage gerne zu.

Drews EDV + Btx GmbH
Bergheimerstraße 134 b
D-6900 Heidelberg
Telefon (0 62 21)
2 99 00 und 2 99 44
Btx-Nummer 0622129900
Btx-Leitseite * 2 99 00 #

d
Drews

CompiMate Der Computerladen
H & P Rodat GbR

Superpreise rund um den AMIGA :

VORTEX - Festplatten in allen Größen
(Autobootend auch mit KICK 1.2)

20 MB **DM 978,-**

30 MB **DM 1148,-**

40 MB **DM 1348,-**

AMIGA Zweitlaufwerk extern NEC 1037a
DM 259,-

AMIGA 2000 nur DM 1875,-

(neueste Version mit 1 MB CHIP - RAM)

Flickerfixer für A2000 (volle 768 x 556 Auflösung
ohne Interlaceflimmern)

Sonderpreis nur DM 1198,-

Der Hammer im Sommer :

HURRICANE Turboboard f. A2000

mit 68020 und 68881 16 MHz mit Anschluß für 32-Bit RAM Karte

Absoluter Superpreis DM 1798,-

**Weitere Superpreise in unserer AMIGA -
Liste, die wir gerne kostenlos übersenden.**
(Software, Hardware etc.)

Wir versenden per UPS - Nachnahme DM 10,-

CompiMate Computer, Sudbrackstr. 31, 4800 Bielefeld 1
Telefon 0521 - 133621 / FAX 0521 - 124333

UND SIE BEWEGEN SICH DOCH



Mit den Programmen *Cel Animator* und *Transport Controller* versucht das amerikanische Softwarehaus *Micro Illusions*, hierzulande bekannt durch *PHOTON Paint*, auch auf dem Animationssektor Fuß zu fassen. Der *Cel Animator* ist dabei das eigentliche Animationsprogramm, der *Transport Controller* dient dazu, einen Videorecorder zu steuern und so die Überspielung einer Animation auf Video zu erleichtern.

PHOTON VIDEO-Cel Animator und Transport-Controller

Beide Programme laufen auf allen AMIGA-Typen mit mindestens 512K Speicher, allerdings ist zu bemerken, daß damit beim *Cel Animator* kein ernsthaftes Arbeiten möglich ist. Wirklich interessant wird es erst ab 2Mb RAM und einer Festplatte.

Beide Programme warten mit je einem dicken Handbuch auf, die aber leider in Englisch sind. Inhaltlich wie optisch sind sie sehr gut gelungen und bieten sowohl dem Anfänger als auch dem Profi einen leichten Einstieg in das Programm. Beim Animationsprogramm gibt's sogar noch eine Quick-Reference-Card dazu. Der *CEL-Animator* verarbeitet alle gängigen und weniger gängigen Grafikmodi des AMIGA. Der gewünschte Modus wird beim Start des Programms ähnlich wie bei *DPaint II* eingestellt. Es scheint dem Programmierer allerdings entgangen zu sein, daß

es auf der Welt nicht nur NTSC-AMIGAs gibt. Ob eine Anpassung auf die europäische PAL-Norm vorgenommen wird, ließ sich nicht in Erfahrung bringen. Die Programme verrichten ihren Dienst aber auch auf PAL-AMIGAs. Was in dem Programm selbst steckt, beweisen die mitgelieferten Demos.

PHOTON VIDEO-Cel Animator

Das Programm *Cel Animator* wird komplett über die obere Menüleiste gesteuert. Dort sind unter anderem auch einige Malwerkzeuge "versteckt", mit denen man seinen Animationen noch schnell "den letzten Schliff" geben kann. Ansonsten finden sich dort die üblichen Funktionen, die ein Animationsprogramm ausmachen. Besondere Erwähnung sollte dabei die Möglichkeit



Bild 2: Beim Cel Animator muß zunächst die gewünschte Auflösung und Farbanzahl eingestellt werden, des weiteren läßt sich eine Animation geladen und die Anzahl der Frames angeben.

führen sind oder auf allzu gewagte Experimente des Testers, steht in den Sternen...

PHOTON VIDEO-Transport Controller

Der Transport Controller sorgt dafür, daß die fertigen Animationen relativ problemlos auf Video übertragen werden können. Er arbeitet mit verschiedenen sogenannten Video-Tape-Machine-controller-interfaces (VTMs) zusammen, die einen Video-Tape-Recorder (VTR) steuern. VTMs dienen dazu, Computer (in diesem Fall den AMIGA) in die Lage zu versetzen, einen Videorecorder direkt zu steuern, ähnlich wie bei der Datasette des C64. Der Transport Controller stellt dabei über ein Nullmodem die Verbindung zwischen dem VTM und dem AMIGA her und steuert das Gerät entsprechend den Anweisungen des Anwenders oder eines Animationsprogramms. Am besten arbeitet der Transport Controller mit dem Cel Animator zusammen (warum wohl!), aber auch VideoScape 3D gibt sich die Ehre.

Das Handbuch ist zwar nicht so umfangreich wie beim Cel Animator, es liest sich aber sehr gut und befriedigt dabei nicht nur das Informationsbedürfnis des Lesers, sondern sorgt nebenbei auch für eine gründliche Auffrischung des Schulenglischen. Im Anhang findet der passionierte Videofreak Beispielprogramme in C und Assembler, die veran-

finden, das Abspielen von IFF-Musikstücken exakt dem Ablauf der Animation anpassen zu können. Einige Menüpunkte sind aber sehr unbefriedigend gelöst. Beispielsweise gibt es einen nicht gerade vorbildlichen Requester zum Laden von kompletten Animationen; wenn man aber einfach nur ein Bild einlesen möchte, verlangt das Programm kühn die manuelle Eingabe des kompletten Pfad- und Bildnamens, ohne dem geplagten Benutzer die Möglichkeit zu geben, sich doch wenigstens das Inhaltsverzeichnis des aktuellen Laufwerks ausgeben zu lassen. Was das bei einer einigermaßen gefüllten Festplatte bedeuten kann, weiß wohl jeder Besitzer eines solchen Gerätes aus eigener, leidvoller Erfahrung. Das Fehlen der deutschen Tastaturbelegung fällt dabei schon gar nicht mehr ins Gewicht.

Offensichtlich war man bei dieser Softwarefirma sehr optimistisch, was den RAM-Speicher des Benutzers angeht. Beim Laden von fertigen Animationen wird nämlich einfach "vergessen", einmal nachzuschauen, ob für all die bunten Bilder überhaupt genug Platz ist. Bei längeren Animationen wird dies schnell störend, weil dann die letzten Bilder einfach fehlen.

Ein weiteres Problem beim Testen waren die sporadisch auftretenden Guru-Fehlermeldungen. Ob diese allerdings auf mangelnde Programmtests des Herstellers vor der Auslieferung zurückzu-

Wem nach dem Start des Programms einfällt, daß wegen zu wenig Speicher die Anzahl der möglichen Animationsstufen im 16farbigen HIRES OVERSCAN-Modus doch etwas gering ist und in ein weniger speicherintensives Bildformat umschalten möchte, sucht vergebens nach einem entsprechenden Menüpunkt. Auch die Tatsache, daß eine Festplatte erst extra angemeldet werden muß, stimmt ärgerlich, außerdem trägt diese NICHT in allen Fällen den Namen DH0:, aber das ist einigen Leuten bei Micro Illusions anscheinend nicht in den Sinn gekommen.

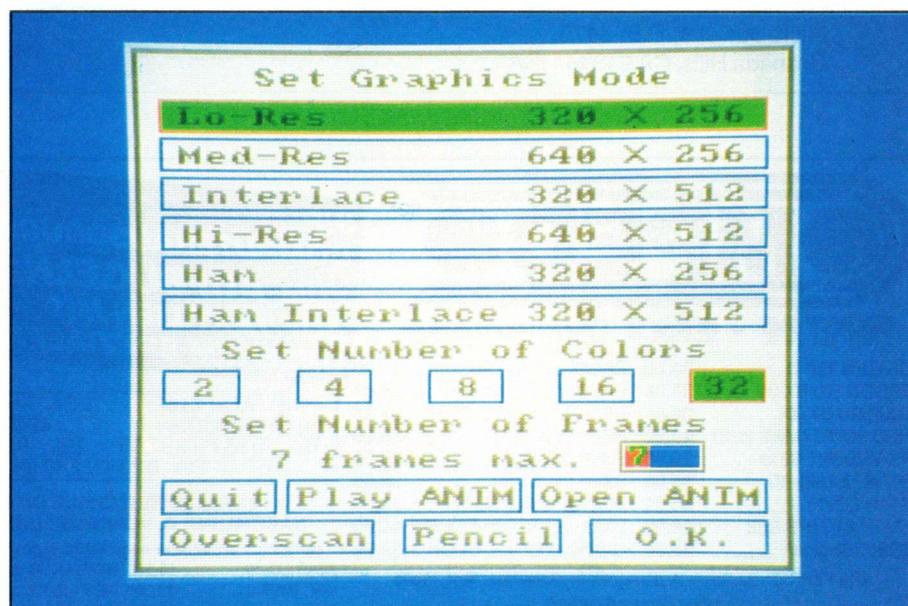


Bild 3: Die einzelnen Frames können vielfältig gestaltet werden.

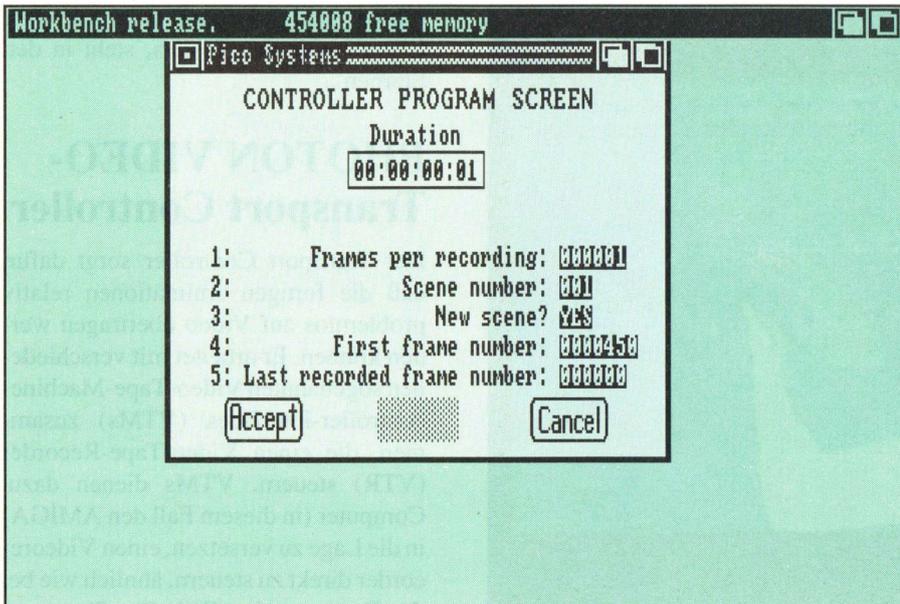


Bild 4: Der Transport Controller übergibt einzelne Sequenzen an den Videorecorder, jedoch benötigt man hierzu einen Hardware-Adapter.

schaulichen, wie man den Transport Controller von eigenen Programmen aus anspricht.

Da die Götter uns Menschen in ihrer unendlichen Weisheit die Möglichkeit

gaben, Fehler einzusehen und sie wieder gutzumachen, können wir hoffentlich bald mit der überarbeiteten, deutschen Version 1.1 des Programms rechnen. Cel Animator hat das Zeug zu einem einfachen, leicht beherrschbaren Ani-

mationsprogramm. Zusammen mit PHOTON Paint bildet Cel Animator ein durchaus leistungsfähiges Gespann, allerdings stellt DPaint III mit seinen Animationsoptionen einen ernsthaften Konkurrenten dar, dem der Cel Animator nur seine Fähigkeit, auch HAM-Formate zu verarbeiten, die Möglichkeit, die Animationen zu vertonen und das Interface zum Transport Controller entgegenzusetzen hat.

PHOTON VIDEO-CelAnimator

- + menügesteuert
- + Anpassung von IFF-Sounds möglich
- + Schnittstelle zum Transport Controller
- + unterstützt alle Bildformate
- + gut gemachtes Handbuch
- unterstützt keine PAL-Auflösung
- Handbuch & Programm in Englisch
- teilweise unkomfortabel
- nicht absturzsicher

Hersteller:

Microllusions, P.B. 3475, Granada Hills, CA 91344/USA.

Anbieter:

CSS
Auf der Warte 46
6367 Karben 1
Tel. 06039/5776

Preis:

PHOTON VIDEO Cell Animator
ca. 250,- DM

PHOTON VIDEO-Transport Controller

- + kompatibel zu vielen Animationsprogrammen
- + einfache Bedienung
- + gut gemachtes Handbuch

- Handbuch & Programm in Englisch

Hersteller:

Microllusions, P.B. 3475, Granada Hills, CA 91344/USA.

Anbieter:

CSS
Auf der Warte 46
6367 Karben 1
Tel. 06039/5776

Preis:

PHOTON VIDEO Transport-Modul auf Anfrage



CITIZEN 120 D	34,90	EPSON LX 80/90	31,90
EPSON FX/RX 80	33,50	EPSON LQ 500/800	35,90
FUJITSU DX/DL	38,90	NEC P2/P6 MPS 2000	37,50
NEC P3/P7/MPS 2010	40,90	NEC P6+/P7+	39,90
PRÄSIDENT 63xx	29,90	STAR SG-10	28,50
STAR NL/NG-10	35,90	STAR LC 24-10	36,80
STAR LC-10	33,90	NEC CP 6 4-COLOR	59,90
TALLY 81/MPS 802	36,90	STAR LC-10C 4-COLOR	46,90
SEIKOSHA SP	35,90	OKI ML 292 4-COLOR	59,90

Alle Farbbänder zum Aufbügeln in Schwarz, Rot, Gelb oder Blau erhältlich

HÄNDLERANFRAGEN ERWÜNSCHT

IHR COMPUTERAUSDRUCK ZUM AUFBÜGELN AUF TEXTILIEN MIT UNSEREM SPEZIALFARBAND

POSTFACH 13 52 POSTFACH 10 0105
5860 ISERLOHN 4630 BOCHUM
TEL.: 02371/29785 TEL.: 0234/12664
Fax: 02371/24099

VERSANDPAUSCHALE DM 6,-
NACHNAHME O. VORKASSE (AUSLAND)

Normale Farbbänder auch SUPER-preiswert! z.B.:

STAR LC-10	10,90	STAR LC 10 COLOR	17,90
NEC P2/P6	12,50	NEC P2/P6 COLOR	30,90
EPSON FX/RX 80	10,60	NEC P6+/P7+ COLOR	36,50
NEC P6+/P7+	14,70	OKI ML 292 COLOR	31,90

- Ausdruck auf Normalpapier
- Bügeln auf T-Shirts, Jacken, Regenschirme, Kissen etc.
- Fertig!!!
- waschecht – Ideal für Werbung
- Lebensdauer wie normales Farbband

FÜR AMIGA – ATARI – PC's

CAMERON HANDY-SCANNER 400 DPI
16 Graust., inkl. Texterkennung **877,-**
CAMERON HANDY-SCANNER 200 DPI
s/w inkl. Handypainter und Texterkennung **544,-**

FLACHBETTSCANNER

200 DPI, 16 Graustufen, inkl. Software **999,-**

DIAMOND VIDEO DIGITIZER (siehe H.C. 4-89, Seite 12) **277,-**

KONZEPTHALTER **23,-**
Weitere Preise auf Anfrage

KICK IT OUT!

Lange Zeit war es nur ein Gerücht, daß es sie auf dem AMIGA gäbe. Nun ist diese Vision zur schrecklichen Wirklichkeit geworden: Der erste Link-Virus auf dem AMIGA ist im Umlauf!

Aber sachte, sachte. So schlimm ist das ja nun auch wieder nicht, denn als Besitzer dieses KICK-START-Heftes werden Ihnen gleich drei Wünsche auf einmal erfüllt. Erstens werden Sie einmal gründlich aufgeklärt. Nein, natürlich über Link-Viren, was sie sind, wie Sie arbeiten, und wie man sie wieder wegbekommt. Zweitens erhalten sie eine Beschreibung des Programms ANTILINK, das dem Aufstöbern und Entfernen von Link-Viren dient. Drittens, und das ist nicht so ganz unwichtig, erhalten Sie das Programm ANTILINK selbst! Also schon mal drei Gründe, dieses Heft zu kaufen (Sie werden sogar noch mehr entdecken!). Doch nun zurück zum Thema.

WAS SIND LINK-VIREN?

Nun, gehen wir es systematisch an. Was Viren sind, dürfte wohl jedem bekannt sein. Für die, die es vergessen haben, will ich es noch einmal erklären. Viren sind kleine Programme, die sich bisher

ANTILINK, MEDIZIN GEGEN LINK-VIRUS

auf dem Bootblock von Disketten befanden, und so beim Booten der infizierten Diskette gestartet wurden. Sie reservierten sich dann einen Speicherbereich, wo sie sich hinkopierten und ihre oft zerstörerischen Werke vollbrachten. Auf jeden Fall versuchten sie jedoch immer, sich auf andere, nicht infizierte Disketten zu kopieren, und sich auf diese Weise zu vermehren. Nun zum zweiten Teil des Wortes: "Link" kommt aus dem Englischen und bedeutet soviel wie "anhängen". Dabei wird die Arbeitsweise von Link-Viren deutlich: Sie versuchen, sich an ausführbare Programme (z.B. CLI-Befehle wie "setmap" oder "dir") "anzuhängen". Das Virusprogramm wird also in das eigentliche Programm eingebunden. Wird ein infiziertes Programm gestartet, so wird zuerst in die Virusroutine verzweigt, bevor das eigentliche Programm gestartet wird. In dieser Virusroutine passiert nun ähnliches wie bei den Bootblockviren: Speicher reservieren, bestimmte Zeiger auf eigene Routinen zeigen lassen, resetfest machen usw. Doch kommen wir zu einem konkreten Beispiel, dem Link-

Virus des IRQ-Teams, der zur Zeit der einzige seiner Art ist. Wird ein von ihm verseuchtes Programm gestartet, setzt er sich resetfest im Speicher fest. Er verbiegt dann den Zeiger der Exec-Funktion "OldOpenLibrary", die auch von vielen CLI-Befehlen angesprochen wird, auf seine eigene, in der er zufallsmäßig (von der Position des Raststrahls abhängig) versucht, sich zu verbreiten. Wird also von irgendeinem Programm die Funktion "OldOpenLibrary" angesprochen, meldet sich der Virus (auch zufallsbedingt) mit einem Text in der Windowleiste. Danach holt er sich den Namen des ersten Programms, das in der Startup-Sequence steht, und versucht, es zu lesen. Findet er es nicht, sucht er es im C-Directory. Ist es größer als 100 000 Bytes, oder handelt es sich um ein nicht ausführbares Programm, geschieht nichts. Es werden zunächst alle Protection-Bits auf Null gesetzt, das heißt, es nützt nichts, ein Programm gegen Schreiben zu schützen, damit der Virus sich nicht reinhängt.

Beim AMIGA bestehen Programme aus Teilsegmenten, sogenannten Hunks, die beim Laden miteinander verkettet werden und so das gesamte Programm ergeben. Der Virus macht nun folgendes: Er erhöht die Anzahl der Hunks um eins und setzt sich selbst als der Hunk, der zuerst ausgeführt werden soll in die Hunktabelle. Er muß nun auch die Nummern der Hunks, auf die sich die zu korrigierenden Offsets beziehen, um eins erhöhen, da er sich ja eingeschoben hat. Hat er sich nun erfolgreich oder weniger erfolgreich verbreitet, springt er in die eigentlich angesprungene "Old OpenLibrary"-Routine, damit niemand Verdacht schöpft. Der IRQ Link-Virus hat keinerlei zerstörerische Wirkung, wenn man davon absieht, daß er jedes

befallene Programm um genau 1096 Byte verlängert und so den Diskettenplatz schrumpfen läßt. Außerdem laufen manche Programme nicht, wenn der Virus aktiv ist, und stürzen einfach ab.

Doch kommen wir zur Problematik, die sich ergibt, wenn man nach einer Strategie sucht, gegen den Link-Virus vorzugehen. Da er ja immer nur versucht, sich an Programme zu linkern, die in der Startup-Sequence stehen, läge es nahe, nur solche zu untersuchen. Was ist jedoch wenn man die Startup-Sequence ändert? Ein eventuell befallenes Programm bleibt weiterhin vorhanden, sein Name kommt jedoch nicht mehr in der Startup-Sequence vor. Ein Killerprogramm würde das befallene File dann nicht mehr erkennen. Eine andere Möglichkeit, die auch schon realisiert wurde, ist, einfach den Namen des zu untersuchenden Programms mit Pfadnamen einzugeben und nur diesen zu untersuchen. Aber vergleichen Sie doch mal die Längen aller ausführbaren Files Ihrer Workbenchdiskette mit denen auf dem Backup. Und tippen Sie doch mal all die verdächtigen Programme mit ihren Pfadnamen ein. Viel Spaß! Die dritte Möglichkeit, die auch mein Programm ANTILINK beinhaltet, ist, alle ausführbaren Programme, die nicht länger als 100 000 Bytes sind, auf Virenbefall zu untersuchen. Dies ist zwar zeitaufwendiger als die anderen beiden Möglichkeiten, jedoch wesentlich effizienter. Sie müssen ja nicht jeden Tag Ihre

komplette Diskettensammlung auf Link-Viren hin untersuchen. Der Virus selbst gibt ja noch früh genug Alarm, wenn er sich gelegentlich mit einem Text, der übrigens per Zufallswert EOR verknüpft im Programm enthalten ist, in der Window-Leiste meldet.

Kommen wir nun also zum Programm ANTILINK (Applaus, Applaus). Das Programm wurde mit dem KICK-ASS-Assembler geschrieben, dürfte aber auch problemlos auf andere Assembler übertragen werden können. Das Programm ist nach dem Assemblieren als Objectfile unter dem Namen Antilink abzuspeichern. Es wird folgendermaßen aufgerufen: Antilink Drive (DFx:, RAM:, VD0: ..).

Es untersucht dann alle ausführbaren Files des angegebenen Drives. Wird der IRQ-Virus in einem Programm entdeckt, wird dessen Name ausgegeben, und man kann das Programm restaurieren oder weitersuchen.

NUN ZUR ERKLÄRUNG:

Nachdem erfolgreich Speicher belegt wurde, wird zuerst getestet, ob sich der Virus im Speicher befindet, das heißt, ob der Zeiger von "OldOpenLibrary" verbogen wurde. Ist dies der Fall, wird eine Meldung ausgegeben, und man hat die Möglichkeit, den Virus im Speicher unschädlich zu machen. Da der Virus sich selbst den alten "OldOpenLibrary"-Zeiger sichert, ist es einfach, den alten Zustand wieder herzustellen. Durch

Löschen des KickTagPtr wird der reset-feste Virus endgültig aus dem Speicher verbannt. Dadurch gehen leider auch alle anderen resetfesten Programme verloren. Das Programm holt sich nun die Namen aller ausführbaren Files, die nicht länger als 100 000 Bytes sind, aus allen Directories und speichert sie mit Pfadnamen in einem Puffer ab (N_Buffer). Danach folgt die Routine, die alle Programme, deren Namen in dieser Liste stehen, auf Virenbefall testet. Dabei kommt uns wieder der Virus zu Hilfe, der nach dem gleichen Verfahren testet, ob er sich schon in dem File befindet. Er erkennt es an dem Befehl "movem.l a6-a0/d7-d0,-(a7)". Hat das Programm einen Virus entdeckt, wird eine Alert-Meldung ausgegeben. Man kann nun entweder weitersuchen oder das befallene File restaurieren. Beim Restaurieren wird das Virusprogramm, das sich im ersten Hunk befindet, gelöscht und die Anzahl der Hunks um eins erniedrigt. Danach werden die Hunknummern, auf die sich die zu korrigierenden Offsets beziehen, ebenfalls um eins erniedrigt. Am Ende hat das befallene Programm die ursprüngliche Länge und zeigt keinerlei Unterschiede zum ursprünglichen Programm.

Das war's. Die von diesem Link-Virus befallenen Programme sind sauber und vollständig restauriert. Sie können aufatmen, der Virus ist von Ihrem Rechner verbannt. ■

```

1:
2:  Open = -30
3:  Close = -36
4:  Read = -42
5:  Write = -48
6:  SetFunction = -420
7:  AllocMem = -198
8:  FreeMem = -210
9:  OldOpenLibrary = -408
10: OpenLibrary = -552
11: CloseLibrary = -414
12: DisplayAlert = -90
13: Examine = -102
14: ExNext = -108
15: Lock = -84
16: Unlock = -90
17: ParentDir = -210
18:
19:  MODE_OLD = 1005
20:  MODE_NEW = 1006
21:
22:  ; Link Virus Killer V1.0
23:  ; Written by Oliver Siebenhaar in 5/89,
24:  ; using the Kick-Ass Assembler
25:  ; (c) MAXON Computer GmbH 1989
26:  ; KICKSTART 1989

```

```

27:
28:  lea Path(pc),a1          ; Drivepath kopieren
29:  subq.l #1,d0
30:  beq.s No_Paars
31:  moveq #3,d0
32:  C_Drive: cmp.b #$0A,(a0)  ; Linefeed?
33:  beq.s No_Paars
34:  move.b (a0)+,(a1)+
35:  dbra d0,C_Drive
36:
37:  No_Paars: cmp.b #' ':'',-(a1)
38:  bne Early_End
39:  move.l $4.s,a6
40:  move.l #260*4,d0        ; Speicher für Fileinfoblock
41:  move.l #$10002,d1      ; reservieren
42:  jsr AllocMem(a6)
43:  beq Early_End
44:  move.l d0,fileinfo
45:
46:  move.l #10000,d0        ; Speicher Namen-Ablage
47:  move.l #$10002,d1      ; reservieren
48:  jsr AllocMem(a6)
49:  beq Early_End
50:  move.l d0,N_Buffer
51:
52:  move.l #100000,d0      ; Speicher für Ablage des →

```

```

53:  move.l #$10002,d1 ; Files reservieren
54:  jsr AllocMem(a6)
55:  beq Early_End
56:  move.l d0,F_Buffer
57:
58:  lea intlub(pc),a1
59:  moveq #0,d0
60:  jsr OpenLibrary(a6) ; Intuition-Library öffnen
61:  move.l d0,intbase ; Basis retten
62:
63:  lea doslib(pc),a1
64:  moveq #0,d0
65:  jsr OpenLibrary(a6) ; Dos-Library öffnen
66:  move.l d0,dosbase ; Basis retten
67:
68:  ; Virus im Speicher ?
69:
70:  cmp.w #$00F0,OldOpenLibrary+2(a6) ; Zeiger
                                verändert?
71:  bpl.s Not_in_Memory ; nein, dann alles OK
72:  move.l intbase(pc),a6 ; Basis Adresse der
                                Intuition-Library
73:  moveq #0,d0 ; Alertnummer = 0
74:  moveq #50,d1 ; Höhe = 50
75:  lea Alert_Mem(pc),a0
76:  jsr DisplayAlert(a6) ; Alert anzeigen
77:  tst.l d0 ; Virus inaktivieren ?
78:  bne.s Not_in_Memory ; nein, dann weiter
79:
80:  ; Kick it out!
81:
82:  move.l OldOpenLibrary+2(a6),a0
83:  move.l 18(a0),d0 ; hier steht der alte Vektor
84:  move.l $04.s,a6
85:  move.l #OldOpenLibrary,a0
86:  move.l $04.s,a1
87:  jsr SetFunction(a6) ; alten Vektor setzen
88:  clr.l 550(a6) ; KickTagPtr löschen
89:
90:  Not_in_Memory: move.l dosbase(pc),a6
91:
92:  ; Diskette untersuchen
93:
94:  move.l N_Buffer(pc),a5 ; Anfang der Namenliste
95:  lea Stack-4(pc),a4 ; eigenen Stack
                                initialisieren
96:  bsr.s Get_Fnames ; Namen holen
97:  bsr Check_Files ; Files testen
98:  bra Ende
99:
100: Init_Path: move.l dosbase(pc),a6
101: move.l #Path,d1
102: moveq #-2,d2
103: jsr Lock(a6)
104: moveq #0,d6
105: tst.l d0
106: bne.s No_Open_Error
107: moveq #-1,d6
108: No_Open_Error: move.l d0,locksav
109: move.l d0,d1
110: move.l fileinfo(pc),d2
111: jsr Examine(a6)
112: clr.l (a4)
113: rts
114:
115: Get_Fnames: bsr.s Init_Path
116: tst.l d6 ; Fehler beim LOCKen?
117: bpl.s Check_on_Disk
118: rts ; wenn ja, dann Ende
119: Check_on_Disk: move.l dosbase(pc),a6
120: move.l locksav(pc),d1
121: move.l fileinfo(pc),d2
122: jsr ExNext(a6)
123: tst.l d0
124: bne.s Check_it
125: move.l Path_End(pc),a1
126: Seek_Begin: cmp.b #'/',-(a1)
127: beq.s Seek_End
128: cmp.b #':',(a1)
129: beq.s Seek_End
130: bra.s Seek_Begin
131: Seek_End: move.l a1,Path_End
132: addq.l #1,a1

```

```

133:  clr.b (a1)
134:
135:  move.l locksav(pc),d1
136:  move.l locksav(pc),d5
137:  jsr ParentDir(a6) ; ein Directory höher
138:  tst.l d0
139:  bne.s No_Root_Dir
140:  move.l locksav(pc),d1
141:  jsr Unlock(a6)
142:  rts
143:
144:  No_Root_Dir: move.l d0,locksav
145:  move.l d5,d1
146:  jsr Unlock(a6)
147:
148:  addq.l #4,a4
149:  move.l (a4),d5
150:  subq.l #1,d5
151:  bsr Init_Path
152:  Get_Position: move.l locksav(pc),d1
153:  move.l fileinfo(pc),d2
154:  jsr ExNext(a6)
155:  addq.l #1,(a4)
156:  dbra d5,Get_Position
157:  bra.s Check_on_Disk
158:
159:  Check_it: addq.l #1,(a4)
160:  move.l fileinfo(pc),a0
161:  tst.l 4(a0) ; Directory?
162:  bmi.s No_Dir
163:
164:  ; Directory gefunden
165:
166:  move.l locksav(pc),d1
167:  jsr Unlock(a6)
168:  lea Path(pc),a1 ; Neuen Pfad anfügen
169:  Seek_Path_End: tst.b (a1)+
170:  bne.s Seek_Path_End
171:  subq.l #1,a1
172:  move.l fileinfo(pc),a0
173:  lea 8(a0),a0
174:
175:  C_Dir: move.b (a0)+,(a1)+
176:  tst.b (a0)
177:  bne.s C_Dir
178:  move.b #'/',(a1)+
179:  clr.b (a1)
180:  subq.l #1,a1
181:  move.l a1,Path_End
182:  subq.l #4,a4
183:  bsr Init_Path
184:  bra Check_on_Disk
185:
186:  No_Dir: move.l dosbase(pc),a6 ; Basisadresse der
                                DOS-Library nach a6
187:  move.l a5,a3
188:  lea Path(pc),a0
189:  C_Path: move.b (a0)+,(a5)+
190:  tst.b (a0)
191:  bne.s C_Path
192:  move.l fileinfo(pc),a0
193:  lea 8(a0),a0
194:
195:  C_Name: move.b (a0)+,(a5)+ ; Namen kopieren
196:  tst.b (a0)
197:  bne.s C_Name
198:  clr.b (a5)+
199:  move.l a3,d1 ; Testen ob executable File
200:  move.l #MODE_OLD,d2
201:  jsr Open(a6)
202:  tst.l d0
203:  beq.s Non_Ex
204:  move.l fileinfo(pc),a0
205:  cmp.l #100000,124(a0) ; File zu lang?
206:  bgt.s Non_Ex
207:  move.l d0,handle
208:  move.l d0,d1
209:  move.l F_Buffer(pc),d2
210:  moveq #4,d3 ; 1 Langwort lesen
211:  jsr Read(a6)
212:  tst.l d0
213:  bmi.s Non_Ex
214:  move.l handle(pc),d1

```

```

215:   jsr Close(a6)
216:   move.l F_Buffer(pc),a0
217:   cmp.l #$000003F3,(a0); Testen ob ausführbar
218:   beq.s Executable
219: Non_Ex: clr.b -(a5)    ; fehlerhaften Filennamen
220:   cmp.l a5,a3        ; löschen
221:   bne.s Non_Ex
222: Executable:
223:   bra Check_on_Disk
224:
225:   ; Virustestroutine
226:
227: Check_Files: move.l N_Buffer(pc),a5
228:
229: Check_Loop: move.l dosbase(pc),a6
230:   tst.b (a5)
231:   beq Check_End
232:   move.l a5,a4
233: Filename_On: tst.b (a5)+
234:   bne.s Filename_On
235:   move.l F_Buffer(pc),a0
236:   move.l #500-1,d0
237: Clr_FBuffer: clr.b (a0)+
238:   dbra d0,Clr_FBuffer
239:   move.l a4,d1
240:   move.l #MODE_OLD,d2
241:   jsr Open(a6)      ; File suchen
242:   tst.l d0
243:   beq.s Check_Loop ; Error
244:   move.l d0,d5      ; Filehandle retten
245:   move.l d0,d1
246:   move.l F_Buffer(pc),d2 ;Puffer für File nach d2
247:   move.l #500,d3    ; Anzahl der zu lesenden Byte
248:   jsr Read(a6)     ; lesen
249:   move.l d5,d1
250:   jsr Close(a6)    ; File schließen
251:   move.l F_Buffer(pc),a0
252:   move.l 8(a0),d6
253:   lsl.l #2,d6      ; feststellen, ob sich der Virus
254:   cmp.l $FFFFE6100,30(a0,d6) ; im ersten Hunk
                        befindet
255:   beq.s Virus      ; wenn ja, dann Virus anzeigen
256:   bra.s Check_Loop
257:
258: Virus: lea File(pc),a0
259:   moveq #29,d0
260: Clr_Buffer: move.b #$20,(a0)+ ;Buffer für
                        Filennamen löschen
261:   dbra d0,Clr_Buffer
262:   move.l a4,a0      ; Filename
263:   lea File(pc),a1
264: cfile: move.b (a0)+,(a1)+ ; Filennamen in Alert
                        Meldung kopieren
265:   tst.b (a0)
266:   bne.s cfile
267:   move.l intbase(pc),a6 ; Basis Adresse der
                        Intuition-Library
268:   moveq #0,d0      ; Alertnummer = 0
269:   moveq #50,d1     ; Höhe = 50
270:   lea Alert(pc),a0
271:   jsr DisplayAlert(a6) ; Alert anzeigen
272:   tst.l d0         ; Weitersuchen?
273:   bne Check_Loop  ; ja, dann weiter
274:
275: Kill: movem.l d0-d7/a0-a6,-(sp) ;Register retten
276:   move.l dosbase(pc),a6
277:   move.l a4,d1
278:   moveq #-2,d2
279:   jsr Lock(a6)
280:   move.l d0,locksav
281:   move.l d0,d1
282:   move.l fileinfo(pc),d2
283:   jsr Examine(a6)
284:   tst.l d0
285:   beq Kill_End
286:   move.l fileinfo(pc),a0
287:   move.l 124(a0),d7 ; Filelänge nach d7
288:   move.l locksav(pc),d1
289:   jsr Unlock(a6)
290:   move.l a4,d1
291:   move.l #MODE_OLD,d2
292:   jsr Open(a6)
293:   tst.l d0

```

```

294:   beq Kill_End
295:   move.l d0,handle
296:   move.l d0,d1
297:   move.l F_Buffer(pc),d2
298:   move.l d7,d3
299:   jsr Read(a6)
300:   move.l handle(pc),d1
301:   jsr Close(a6)
302:   move.l a4,d1      ; Filename
303:   move.l #MODE_NEW,d2
304:   jsr Open(a6)     ; File neu öffnen
305:   tst.l d0
306:   beq.s Kill_End
307:   move.l d0,a5      ; Filehandle in a5 speichern
308:   move.l F_Buffer(pc),a4 ; Filebasis nach a4
309:   subq.l #1,8(a4)  ; Anzahl der Hunks-1
310:   subq.l #1,16(a4) ; letzter Hunk-1
311:   move.l a5,d1
312:   move.l a4,d2
313:   moveq #$14,d3    ; Erste 5 Langworte
314:   jsr Write(a6)   ; abspeichern
315:   lea $18(a4),a4  ; Länge des Virus und erste
316:   move.l a5,d1    ; 5 Langworte addieren
317:   move.l a4,d2
318:   move.l d6,d3    ; Länge der Hunk Tabelle
319:   jsr Write(a6)  ; abspeichern
320:   lea 0(a4,d6),a4 ; Länge der Hunk Tabelle
                        addieren
321:   lea $444(a4),a4 ; Länge des Virus-Hunks
                        addieren
322:   move.l a4,a0
323:   move.l a4,a1
324:   sub.l d6,d7     ; Länge der Hunk-Tabelle,
325:   sub.l #$14,d7  ; erste 5 Langworte,
326:   subq.l #4,d7   ; Viruslänge in Tabelle
327:   sub.l #$444,d7 ; und Länge des Virus von
328:                  ; ursprünglicher PRG Länge
                        abziehen
329:   lea 0(a1,d7),a1
330:
331: Seek_Hunk:
332:   cmp.l #$000003EC,(a0)+ ; in dieser Routine
333:   bne.s Seek_Hunk      ; werden die Hunknummern
334:                          ; der zu korrigierenden
335: S2: move.l (a0)+,d0    ; Offsets korrigiert
336:   beq.s Test_End
337:   subq.l #1,(a0)+
338:   lsl.l #2,d0
339:   lea 0(a0,d0),a0
340:   bra.s S2
341:
342: Test_End: cmp.l a0,a1
343:   bgt.s Seek_Hunk
344:   move.l a5,d1      ; ursprüngliches PRG
345:   move.l a4,d2
346:   move.l d7,d3     ; Länge des ursprünglichen
347:   jsr Write(a6)   ; Programms abspeichern
348:   move.l a5,d1
349:   jsr Close(a6)
350: Kill_End: movem.l (sp)+,d0-d7/a0-a6
351:   bra Check_Loop  ; weitersuchen
352: Check_End: rts
353:
354: Ende: move.l $4.s,a6
355:   move.l dosbase(pc),a1 ; Librarys schließen
356:   jsr CloseLibrary(a6)
357:   move.l intbase(pc),a1
358:   jsr CloseLibrary(a6)
359:   move.l fileinfo(pc),a1
360:   move.l #260*4,d0
361:   jsr FreeMem(a6)  ; Speicher freigeben
362:   move.l N_Buffer(pc),a1
363:   move.l #10000,d0
364:   jsr FreeMem(a6)
365:   move.l F_Buffer(pc),a1
366:   move.l #100000,d0
367:   jsr FreeMem(a6)
368:
369: Early_End: moveq #0,d0
370:   rts              ; Programm beenden
371:
372: dosbase: dc.l 0
373: intbase: dc.l 0

```

```

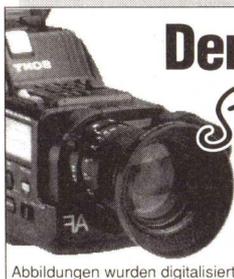
374: handle: dc.l 0
375: fileinfo: dc.l 0
376: N_Buffer: dc.l 0
377: F_Buffer: dc.l 0
378: Path_End: dc.l 0
379: locksav: dc.l 0
380: doslib: dc.b 'dos.library',0
381: intlhb: dc.b 'intuition.library',0
382: even
383: Path: dc.b 'df1:',0 ; Puffer für Pfad
384: blk.b 500
385:
386: even
387: blk.l 50,0 ; eigener Stack

```

```

388: Stack:
389: ; Alertmeldungen
390: even
391: Alert: dc.b 0,220,20,'Link Virus in file '
392: File: blk.b 30,$20
393: dc.b 0,1
394: dc.b 0,80,40,'Left mouse = continue '
395: dc.b ' Right mouse = restore file',0,0
396: Alert_Mem: dc.b 0,220,20,'IRQ Link Virus found
in memory '
397: dc.b 0,1
398: dc.b 0,80,40,'Left mouse = continue '
399: dc.b ' Right mouse = Kick it out!',0,0
400: ; LISTING ENDE

```



Abbildungen wurden digitalisiert

Der neue Echtzeit-Videodigitizer

SNAPSHOT!

1/50 Sekunde für ein professionelles Bild macht Schluß mit dem minutenlangen Warten.

Telefonische Fachberatung
ab 16 Uhr: (0431) 9 44 24
Händler-Anfragen erwünscht.

- Alle AMIGA-Grafikmodi werden unterstützt
- Auflösungen bis zu 704 x 552 Bildpunkten
- 16 / 32 / 46 Grautöne oder 4096 Farben
- 4 verschiedene Video-Eingänge

SNAPSHOT! Professional 845,-
RGB-Vorsatz für Farbbilder ab 125,-

SNAPSHOT! STUDIO 1970,-
(19 Zoll, Super-VHS, Farbe)
2 Demo-Disketten 10,-
Kostenlose Informationen anfordern!

Made in Germany

VIDEOTECHNIK DIEZEMANN

Dammstraße 42, 2300 Kiel 1

NEU !!! KICKS COLLECTION

KICKS Collection ist unsere neue Software-Serie, mit der wir Ihnen die besten AMIGA-KICKS präsentieren wollen. Die Programme sind sowohl

in ausführbarer Form, als auch im Sourcecode vorhanden, so daß Programmierer sie in eigene Programme übernehmen können.

ANTIVIRUS-Sammlung

Auf dieser Diskette haben wir wichtige Programme und Utilities zusammengestellt, mit denen Sie Viren bekämpfen und sich wirksam dagegen schützen können.

- **ANTILINK:** Programm zum Auffinden und Zerstören von IRQ-Linkviren. ANTILINK überprüft die Systemvektoren auf eingeklinkte Viren und entfernt sie aus dem Speicher. Weiterhin überprüft es komplette Disketten/Festplatten auf virenbefallene Programme und restauriert sie automatisch.
- **BOOTPREVENTOR:** Verhindert das Ausführen von Bootprogrammen. Ist das Programm einmal gestartet, wird der Bootblock nach einem Reset nicht mehr

ausgeführt. Ein eventuell dort vorhandener Virus kann somit nicht aktiviert werden.

- **BOOTBLINK:** Bootblockschutzprogramm. Zeigt optisch an, ob der Bootsektor in Ordnung ist.
- **VEKTORCHECK:** Überprüft alle kritischen Vektoren.

Und viele andere.

Neben den direkt ausführbaren Versionen befinden sich für Programmierer ebenfalls die Sourcecodes der Programme in verschiedenen Sprachen auf der Diskette.

ANTIVIRUS-Sammlung DM 19,90

- zuzüglich Versandkosten
Inland DM 5,-
Ausland DM 10,-
- nur Vorkasse möglich

MAXON Computer • Industriestr. 26
6236 Eschborn • Tel.: 06196 / 481811

Demnächst erscheinen:

KICKS Collection 'Assembler'
KICKS Collection 'C'
KICKS Collection 'BASIC'

ESPERANTOMAT

Übersetzungen per Computer

Viele AMIGA-Benutzer haben mit der englischen Sprache Schwierigkeiten. So manchem von ihnen bleiben viele nur in englisch beschriebene Anwenderprogramme und Spiele unklar. Gerade diesen Lesern will das Übersetzungsprogramm ESPERANTOMAT ein wertvoller Helfer sein. Der ESPERANTOMAT übersetzt über die Tastatur eingegebene fremdsprachliche Texte problemlos in einen verständlichen deutschen Text. Die Ausgabe des übersetzten Textes kann auf den Bildschirm, in eine Datei und zusätzlich auf einen Drucker erfolgen.

Eine übersichtliche Pull-Down-Menüsteuerung macht auch den ungeübten Benutzer schnell mit der Bedienung des Programms ESPERANTOMAT vertraut. Durch einen Programmtrick wird der Übersetzungsvorgang zudem recht schnell, d.h. die deutsche Bedeutung des Fremdwortes wird ohne merkliche zeitliche Verzögerung ausgegeben. Das Programm übersetzt alle Sprachen, die sich an den ASCII-Code halten, in die deutsche bzw. eine beliebige andere Sprache. Das Programm untersucht in beliebig langen Texten Wort für Wort, ob eine Übersetzung vorhanden ist. Ist das der Fall, wird das Fremdwort durch das entsprechende (deutsche) Wort ersetzt. Es ist so in der Lage, z.B. englische Anleitungen in eine verständliche Form zu bringen. Diese Möglichkeit möchte ich auch den KICKSTART-Lesern anbieten.

Vor etwa einem Jahr kaufte ich mir einen AMIGA. Nach und nach habe ich mir die wichtigsten Public Domain-Disketten bestellt. Die auf diesen Disketten befindlichen Utilities konnte ich aber nicht benutzen, da mir die Bedienung unbekannt war und auch in vielen Versuchen nicht ermittelt werden konnte. Nachdem ich endlich den "DIR OPT A"-Befehl anwenden konnte, habe ich auf der Diskette Textdateien entdeckt, die offensichtlich Anleitungen waren. Mit Hilfe eines Übersetzungsbuches versuchte ich dann, Wort für Wort zu übersetzen. Wie man sich denken kann, ist dieser Weg sehr mühselig. Es liegt nahe, diese Arbeit den AMIGA selbst erledigen zu lassen. Es mußte ein Programm her, das folgende Bedingungen zu erfüllen hat:

1. Wenn die Anleitung bzw. der Brief in einer fremden Sprache vorliegt, muß es möglich sein, ihn einzuladen und den übersetzten deutschen Text auf dem Bildschirm und auf einem Drucker ausgeben zu lassen.
2. Zudem sollte der übersetzte Text in einer Textdatei abspeicherbar sein, damit man ihn gegebenenfalls weiterbearbeiten kann.
3. Worte, die noch nicht bekannt sind, muß das Programm automatisch zur Übersetzung anbieten, damit man eine sinngemäßen Bedeutung ergänzen kann.
4. Die Übersetzung muß in einer annehmbaren Zeit erfolgen.
5. Das Programm darf nicht zu umfangreich sein und sollte leicht bedienbar bleiben.

Aus den Punkten entstand das vorliegende Programm ESPERANTOMAT, dessen einzelne Menüpunkte nun erläutert werden sollen.

Bedienungsanleitung für ESPERANTOMAT

Vokabeln laden

Die Vokabeldatei muß als erstes geladen werden. Der Name der Vokabeldatei lautet "vok.DAT". Zum Laden dieser Datei wird nur "vok" <Return> eingegeben. Danach wird die Anzahl der geladenen Vokabeln angezeigt. Während des Ladens werden die Vokabeln in 27

“Schubladen” einsortiert. Zur Kontrolle wird während des Ladevorgangs ein “Füllstandszeiger” gezeigt. Das Fremdwort muß stets kleingeschrieben sein. Erlaubt sind nur die Anfangsbuchstaben “a” bis “z”. Die Adresse jeder Schublade (“a” ... “z”) ist der Anfangsbuchstabe. Bei einem späteren Suchvorgang schaut das Programm nur in der jeweils richtigen Schublade nach. Der Suchvorgang wird dadurch angenehm kurz.

Vokabeln sichern

Bei Anwahl dieses Menüpunktes wird die Vokabeldatei auf Diskette gesichert. Der Hinweis “JA” bzw. “NEIN” zeigt an, ob noch nicht gesicherte Vokabeln im Speicher stehen. Zum Sichern muß der Dateiname “vok” und <Return> eingegeben werden. Um nun die erweiterte Vokabeldatei vollständig in den Speicher zu laden, und um außerdem den für neue Vokabeln reservierten Speicher wieder freizumachen, erfolgt ein automatischer Restart. Danach muß die Vokabeldatei neu geladen werden.

Verzeichnis wechseln

Das Verzeichnis kann leicht gewechselt werden, um, z.B. die Datendiskette wechseln zu können. Diese muß dann mit z. B. ‘DF0:’ dem Programm bekannt gemacht werden. Um in ein Unterverzeichnis zu gelangen, könnte man etwa eingeben: ‘DF0:daten’.

Directory listen

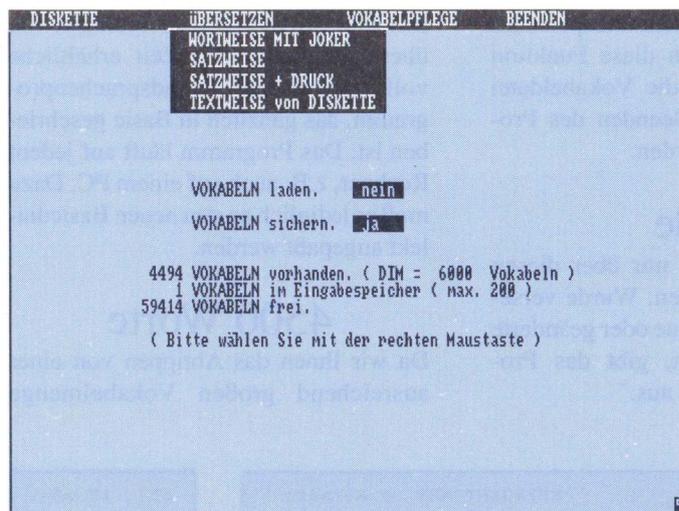
Hier kann man sich den Inhalt des angemeldeten Verzeichnisses ansehen. Das ist sehr zu empfehlen, wenn man das zu ladende Programm nicht finden kann.

Wortweise übersetzen

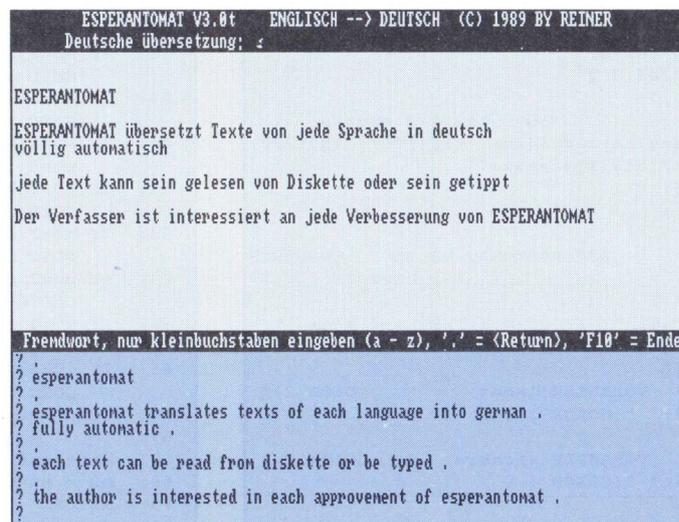
Unter diesem Punkt kann die deutsche Übersetzung einer einzelnen Vokabel abgefragt werden. Ist sie unbekannt, wird automatisch die Eingabe der deutschen Übersetzung angefordert. Nach deren Eingabe wird die Vokabel in den Eingabespeicher übernommen und das Menü berichtigt. Komfortabel ist das Suchen mit dem Joker (*). Gibt man z.B. “ab*” ein, werden alle Übersetzungen aufgelistet, die mit “ab” beginnen.

Satzweise übersetzen/ Satzweise + Druck

Unter diesen Punkten kann ein beliebiger langer Satz in der Fremdsprache einge-



Das Anfangsmenü des ESPERANTOMAT mit geladener Vokabeldatei



Beispiel einer Übersetzung mit ESPERANTOMAT

geben werden. Nach Abschluß des Satzes mit <Return> wird die deutsche Übersetzung ausgegeben. Nicht übersetzbare Vokabeln werden unverändert revers in den übersetzten Text übernommen. Nach Drücken der Taste “F10” kann die deutsche Bedeutung jeder zunächst nicht übersetzbaren Vokabel eingegeben werden. Bei Eingabe eines einzelnen Punktes ‘.’ <Retrun> wird die Zeile beendet und der Cursor an den Textanfang gestellt. Im Text ist der Punkt nicht erlaubt. Der deutsche Text kann zusätzlich auf den Drucker (epson-kompatible) ausgegeben werden. Dem Programm noch nicht bekannte Vokabeln sind mit einem Stern (*) gekennzeichnet.

Über den Menüpunkt “SATZWEISE + DRUCK” kann man sich so eine deutsche Bedienungsanleitung erstellen.

VOKABELN anzeigen

Der gesamte Vokabelvorrat läßt sich

unter diesem Punkt wahlweise auf den Bildschirm (seitenweise) oder auf dem Drucker ausgeben. Die Richtigkeit der Übersetzung kann so überprüft und gegebenenfalls unter dem Menüpunkt “Vokabeln ändern” nachgebessert werden. Durch Drücken der ESC-Taste wird die Anzeige abgebrochen.

VOKABELN ändern

Unter diesem Menüpunkt läßt sich die Bedeutung einer gespeicherten Vokabel ändern: Nachdem die Vokabel in der fremden Sprache eingegeben wurde, erscheint sie zusammen mit ihrer derzeit gültigen deutschen Übersetzung auf dem Bildschirm. Anschließend ist die Eingabe einer neuen Bedeutung möglich.

VOKABEL löschen

Nach Eingabe einer zu löschenden Vokabel wird diese im Speicher durch ein Löschkennzeichen ersetzt. Dadurch

wird sie beim nächsten Abspeichern nicht gesichert. Wenn diese Funktion benutzt wurde, muß die Vokabeldatei also unbedingt vor Beenden des Programms gesichert werden.

Ende

Das Programm kann nur über diesen Punkt verlassen werden. Wurde versehentlich vergessen, neue oder geänderte Vokabeln zu sichern, gibt das Programm eine Warnung aus.

Mit ESPERANTOMAT verfügen Sie über das einzige zur Zeit erhältliche vollautomatische Fremdsprachenprogramm, das gänzlich in Basic geschrieben ist. Das Programm läuft auf jedem Rechner, z.B. auch auf einem PC. Dazu muß es lediglich an den neuen Basicdiagnostik angepaßt werden.

4500 Worte

Da wir Ihnen das Abtippen von einer ausreichend großen Vokabelmenge

nicht zumuten können, haben wir eine umfangreiche Vokabeldatei zusammengestellt. Diese 4500 Wörter umfassende Datei (ENGLISCH-DEUTSCH) ist zum Preis von nur DM 19.90 samt ESPERANTOMAT auf einer KICKSTART SPEZIAL-Diskette enthalten. Das Programm ist kompiliert und dadurch um einiges schneller als die AmigaBASIC-Version.

```

1: start:          ' ESPERANTOMAT by Reiner
2:                ' (c) 1989 MAXON Computer
3: CLEAR
4: CLEAR , 25000
5: CLEAR , 180000& 'fuer 1/2 MBytes = 100000
6:
7: ON ERROR GOTO fehler
8: SCREEN 1,640,243,3,2
9: WIDTH "scrn:",75
10: an = 6000      '3000 bei 1/2 MBytes
11: DIM bu(27), en$(an), dt$(an), te$(200), td$(200)
12: en$(1)="zzzzz":dt$(1)="zzzzz"
13: ta = 1: bu(26) = 1
14:
15: menue:
16: WINDOW 1,"          ESPERANTOMAT V3.0t   ENGLISCH
          --> DEUTSCH",,1,1
17: WINDOW OUTPUT 1
18: PRINT : ac$="nein": IF ac = 1 THEN ac$ ="ja "
19: al$="ja " :IF al=1 OR ac=1 THEN al$="nein"
20: CLS: LOCATE 10
21: PRINT , "          VOKABELN laden. " ;: COLOR 2,3:
22: PRINT " ";al$;" " :COLOR 1,0
23: PRINT
24: PRINT , "          VOKABELN sichern. " ;: COLOR 2,3:
25: PRINT " ";ac$;" " :COLOR 1,0
26: PRINT
27: PRINT , USING"#####";va+ta-1 ;:PRINT " VOKABELN
          vorhanden. ( DIM = ";an;" Vokabeln )"
28: PRINT , USING"#####"; ta-1 ;:PRINT " VOKABELN im
          Eingabespeicher ( max. 200 )"
29: PRINT , USING"#####"; INT(FRE(1)/22);
          PRINT " VOKABELN frei. "
30: IF va+200 > an THEN PRINT :COLOR 0,1:
          PRINT , " 'an' sollte erhöht werden " :COLOR 1,0
31: PRINT :PRINT , " ( Bitte wählen Sie mit der
          rechten Maustaste )"
32:
33: leiste:
34: MOUSE ON : MENU ON
35: MENU 1,0,1," DISKETTE "
36: MENU 1,1,1," VOKABELN LADEN "
37: MENU 1,2,1," VOKABELN SICHERN "
38: MENU 1,3,1," VERZEICHNIS WECHSELN "
39: MENU 1,4,1," DIRECTORY LISTEN "
40: MENU 2,0,1," ÜBERSETZEN "
41: MENU 2,1,1," WORTWEISE MIT JOKER "
42: MENU 2,2,1," SATZWEISE "
43: MENU 2,3,1," SATZWEISE + DRUCKEN"
44: MENU 2,4,1," TEXTWEISE VON DISKETTE"
45: MENU 3,0,1," VOKABELPFLEGE "
46: MENU 3,1,1," ANZEIGEN ODER DRUCKEN"
47: MENU 3,2,1," ÄNDERN "
48: MENU 3,3,1," LÖSCHEN "
49: MENU 4,0,1," BEENDEN "
50: MENU 4,1,1," ENDE (SICHER?) "
51: ' m0 m1
52: ON MENU GOSUB Maushandler
53:
54: maus:
55: IF m0=1 AND m1=1 THEN GOSUB laden :m1=0:m0=0 :
          GOTO menue

```

```

55: IF m0=1 AND m1=2 THEN GOSUB sichern :m0=0:m1=0 :
          GOTO menue
56: IF m0=1 AND m1=4 THEN CLS: FILES :m0=0:m1=0:
          GOSUB warten: GOTO menue
57: IF m0=1 AND m1=3 THEN GOSUB verzeichnis: m0=0:
          m1=0: GOTO menue
58: IF m0=2 AND m1=1 THEN GOSUB wort :m1=0:m0=0:
          GOTO menue
59: IF m0=2 AND m1=2 THEN GOSUB satz :m0=0:m1=0:
          GOTO menue
60: IF m0=2 AND m1=3 THEN dr=1: GOSUB satz :m0=0:
          m1=0 : GOTO menue
61: IF m0=2 AND m1=4 THEN GOSUB text :m0=0:m1=0:
          GOTO menue
62: IF m0=3 AND m1=1 THEN GOSUB zeigen :m1=0:m0=0:
          GOTO menue
63: IF m0=3 AND m1=2 THEN GOSUB aendern :m0=0:m1=0:
          GOTO menue
64: IF m0=3 AND m1=3 THEN GOSUB loeschen :m1=0:m0=0:
          GOTO menue
65: IF m0=4 AND m1=1 THEN GOSUB ende :m0=0:m1=0:
          GOTO menue
66: GOTO maus
67:
68: Maushandler:
69: m0 = MENU(0) :m1 = MENU(1)
70: RETURN
71:
72: laden:
73: GOSUB menuoff
74: IF ac<>0 OR al=1 THEN GOSUB sicherheit:
          IF f=1 THEN RETURN
75: CLS: LOCATE 10,5
76: INPUT "          Dateiname z.B. 'df0:vok'
          ( Lesen ) : ";ni$
77: laden1:
78: FOR i=1 TO 27: bu(i)=0: NEXT i
79: OPEN ni$+".DAT" AS #1
80: IF LOF(1)=0 THEN
81: PRINT :COLOR 2,3: PRINT , "          Datei nicht
          vorhanden " ;: COLOR 1,0 :xx=1
82: ELSE
83: PRINT :COLOR 2,3: PRINT , "          Datei ";ni$;
          ".DAT wird geladen " ;: COLOR 1,0
84: END IF:
85: CLOSE #1 : IF xx=1 THEN xx=0: m0=0: m1=0:
          GOTO menue
86: OPEN "i",#1,ni$+".DAT"
87: GOSUB zeiger
88: FOR i=1 TO an
89: INPUT#1,en$(i)
90: q=q+1
91: IF q=50 THEN
92: PRINT ">";
93: q=0
94: END IF
95: IF i=3000 THEN LOCATE 24,6
96: IF EOF(1) THEN va=i : i=an: GOTO lad2
97: INPUT#1, dt$(i) :
98: IF EOF(1) THEN va=i : i=an: GOTO lad2
99: IF LEFT$(en$(i),1)<> LEFT$(en$(i-1),1) THEN
          bu(ASC(en$(i))-96)=i
100: lad2: NEXT i

```

```

101: GOSUB satzzeiger
102: ac=0: al=1: CLOSE :RETURN
103:
104: zeiger:
105: LOCATE 20,6
106: PRINT "0.....500.....1000.....1500.....
      2000.....2500.....VOKABELN"
107: LOCATE 23,6
108: PRINT "3000.....3500.....4000.....5000.....
      6000.....6500.....VOKABELN"
109: LOCATE 21,6: COLOR 1,3
110: RETURN
111:
112: sichern:
113: GOSUB menuoff
114: IF al<>1 THEN GOSUB sicherheit:
      IF f=1 THEN RETURN
115: CLS: LOCATE 10
116: PRINT "Nach dem Sichern muß die Datei neu
      geladen werden!":PRINT
117: PRINT ,: INPUT "Dateiname z.B. 'df0:vok'
      ( Schreiben ): "; no$
118: PRINT : COLOR 2,3: PRINT , "      Bitte etwas
      Geduld      ": COLOR 1,0
119: GOSUB zeiger
120: OPEN "o",#2,no$+".DAT"
121: tx=ta
122: IF ta>1 THEN GOSUB sort
123: FOR i=1 TO an
124:   q=q+1
125:   IF q=50 THEN
126:     PRINT ">";
127:     q=0
128:   END IF
129: IF i=3000 THEN LOCATE 24,6
130: IF ta<>0 THEN GOSUB pruefen
131: IF en$(i)=" " THEN sic4
132: IF en$(i)="#" THEN sic3
133: IF dt$(i)=" " THEN sic3
134: PRINT #2,en$(i)+CHR$(13)+dt$(i)
135: sic3: NEXT i
136: sic4: ac=0: al=2: ta=1: CLOSE
137: KILL no$+".DAT.info"
138: GOTO start
139:
140: verzeichnis:
141: CLS
142: LOCATE 15,5
143: INPUT "Welches Verzeichnis: (z.B. 'DF0:') "; vz$
144: CHDIR vz$
145: RETURN
146:
147: sicherheit:
148: CLS
149: LINE (90,58)-(500,175),3,bf ' oranges
      Rechteck zeichnen

150: LOCATE 11
151: GOSUB sic1
152: LINE (90,130)-(500,131),0,bf
153: LINE (299,130)-(302,175),0,bf
154: LOCATE 19,19: PRINT "Ja, in Ordnung"
155: LOCATE 19,41: PRINT "Nein, zurck ins Men"
156: titelbild:
157: w=0:WINDOW 1:aa=MOUSE(0):WHILE MOUSE(0)<>0:WEND:
      WHILE MOUSE(0)=0:WEND
158: x=MOUSE(1):y=MOUSE(2)
159: IF y>130 AND y<175 THEN
160:   IF x>90 AND x<500 THEN
161:     IF x>90 AND x<300 THEN w=1
162:     IF x>300 AND x<494 THEN w=2
163:   END IF
164: END IF
165: IF w=0 THEN titelbild
166: ON w GOTO ja,nein
167: ja:
168: CLS: f=0: RETURN
169: nein:
170: CLS: f=1: RETURN
171: sic1:
172: COLOR 2,3
173: IF al=0 THEN
174:   PRINT , "      W A R N U N G !!"
175:   PRINT , "      Es war keine Datei geladen."
176:   PRINT , "      Eine bereits bestehende Datei"

```

```

177:   PRINT , "      wird überschrieben"
178: END IF
179: IF al=1 THEN
180:   PRINT , "      W A R N U N G !!" :PRINT
181:   PRINT , "      Datei war bereits geladen !!"
182: END IF
183: IF al=2 THEN
184:   PRINT , "      W A R N U N G !!" :PRINT
185:   PRINT , "      Es war keine Datei geladen !!"
186: END IF
187: RETURN
188:
189: wort:
190: CLS: GOSUB menuoff: LOCATE 10
191: PRINT , "W O R T W E I S E übersetzen"
192: PRINT , "Gebe das Wort in Kleinbuchstaben ein.
      <Return> = Men."
193: PRINT , "Kürze das Wort beliebig mit * (Joker)
      ab"
194: ueb5: gf=0
195: ueb1: PRINT : INPUT " englisch: "; en$
196: ac$= LEFT$(en$,1)+" "
197: IF ac$=" " THEN RETURN
198: IF ac$<"a" OR ac$>"{" THEN PRINT "      bitte
      nur Kleinbuchstaben": GOTO ueb1
199: FOR i=bu( ASC (en$)-96) TO bu( ASC (en$)-95)
200:   IF en$(i)=" " THEN i=an: GOTO ueb2
201:   IF en$=en$(i) THEN PRINT "      ";en$(i);
      " = ";TAB(30)dt$(i): i=an: gf=2
202:   IF RIGHT$(en$,1)<>"*" THEN ueb2
203:   gf=1: IF LEFT$(en$, LEN(en$)-1)= LEFT$(
      en$(i),
      LEN(en$)-1) THEN PRINT "      ";en$(i);
      " = ";TAB(30)dt$(i)
204:   IF gf=1 THEN gf=2
205:   ueb2: NEXT i
206:   IF gf=2 THEN ueb4
207:   FOR i=1 TO ta
208:     IF en$=te$(i) THEN PRINT "      ";
      te$(i);
      " = ";TAB(30)td$(i): i=ta: gf=2
209:   IF RIGHT$(en$,1)<>"*" THEN ueb3
210:   gf=1: IF LEFT$(en$,LEN(en$)-1)= LEFT$(te$(i),
      LEN(en$)-1) THEN PRINT "      ";te$(i);
      " = ";td$(i)
211:   IF gf=1 THEN gf=2
212:   ueb3: NEXT i
213:   ueb4:
214:   IF gf=0 THEN GOSUB aufnahme
215:   GOTO ueb5
216:
217: warten:
218: MENU 1,0,0 : MENU 2,0,0 : MENU 3,0,0: MENU 4,0,0
219: PRINT
220: COLOR 1,2: PRINT " Bitte eine TASTE drücken ";:
      COLOR 1,0: PRINT
221: war1: a$=INKEY$:IF a$=" " THEN war1
222: RETURN
223:
224: menuoff:
225: MENU 1,0,0: MENU 2,0,0 : MENU 3,0,0: MENU 4,0,0
226: RETURN
227:
228: satz:
229: GOSUB menuoff
230: WINDOW 2,"      Deutsche Übersetzung :", (0,10)-
      (630,137),0,1
231: WINDOW 3," Fremdwort, nur kleinbuchstaben
      eingeben (a - z), '.' = <Return>, 'F10' =
      Ende", (0,150)-(630,229),0,1
232: tx=ta
233: sat1:
234: WINDOW OUTPUT 3
235: INPUT sa$:sa$=sa$+" "
236:   IF ASC(sa$)=138 THEN WINDOW OUTPUT 2:
      GOTO eingabel
237: en$="": FOR k=1 TO LEN(sa$):gf=0
238:   IF MID$(sa$,k,1)=" " THEN sat3
239:   en$=en$+MID$(sa$,k,1)
240:   sat2: NEXT k
241:   GOTO sat1
242:   sat3:
243:   ac$= LEFT$(en$,1)+" "
244:   IF dr=1 AND ac$=CHR$(46)+" " THEN

```

```

LPRINT CHR$(13)
245: IF ac$=CHR$(46)+" " THEN WINDOW OUTPUT 2:
      PRINT :GOTO sat2
246: IF ac$<"a" OR ac$>"{" THEN
247: WINDOW OUTPUT 2: PRINT "???" ;
248: IF dr=1 THEN LPRINT "???" ;
249: WINDOW OUTPUT 3
250: PRINT "      bitte nur Kleinbuchstaben":
      en$="": GOTO sat2
251: END IF
252: FOR i=bu(ASC(en$)-96) TO bu(ASC(en$)-95)
253: IF en$=en$(i) THEN
254: WINDOW OUTPUT 2
255: PRINT dt$(i);" ";
256: IF dr=1 THEN LPRINT dt$(i);" ";
257: en$="": i=an:gf=2
258: END IF
259: NEXT i
260: IF gf=2 THEN en$="": GOTO sat2
261: FOR i=1 TO ta
262: IF en$=te$(i) THEN
263: IF dr=1 THEN LPRINT td$(i);" ";
264: WINDOW OUTPUT 2
265: IF ASC(td$(i))=36 THEN
266: COLOR 0,1:
267: ll=LEN(td$(i))
268: dd$=RIGHT$(td$(i),ll-1)
269: PRINT dd$;: dd$=""
270: COLOR 1,0 :PRINT " "; :en$="": i=ta: gf=2
271: ELSE
272: PRINT td$(i); " "; :en$="": i=ta: gf=2
273: END IF
274: END IF
275: NEXT i
276: IF gf=0 THEN
277: WINDOW OUTPUT 2
278: IF dr=1 THEN LPRINT "*" ;en$; " ";
279: COLOR 0,1: PRINT en$;: COLOR 1,0 :PRINT " ";:
      te$(ta)=en$: td$(ta)="$"+en$: en$="": ta=ta+1
280: END IF:
281: en$="":GOTO sat2
282:
283: eingabel:
284: CLS: dr=0: GOSUB menuoff: LOCATE 15,5
285: WINDOW OUTPUT 3
286: IF ta>1 THEN ac=1
287: PRINT "      Folgende Vokabeln waren unbekannt":
      PRINT
288: PRINT "      Englisch      Deutsch ( Groß/Klein und
      Umlaute möglich )"
289: FOR i=tx TO ta-1
290: PRINT :PRINT "      ";te$(i);"      ";: td$(i)="
      : INPUT td$(i)
291: IF td$(i)=" THEN te$(i)="#" : td$(i)="#"
292: IF ta>199 THEN einl
293: NEXT i: RETURN
294: einl: PRINT "      Vor weiteren Zugängen muß die
      Datei gesichert werden"
295: GOSUB warten
296: RETURN
297:
298: satzzeiger:
299: bu(0)=0
300: FOR j=1 TO 27
301: IF bu(j)=0 THEN bu(j)=va
302: IF bu(j)=va THEN satz2
303: IF bu(j)>bu(0) THEN satz1
304: PRINT "      W A R N U N G die Sortierfolge der
      Vokabeln ist falsch !!":STOP
305: satz1: bu(0)=bu(j)
306: satz2: NEXT j
307: bu(27)=va
308: RETURN
309:
310: zeigen:
311: CLS: GOSUB menuoff
312: CLS: LOCATE 10,5: PRINT "Ausgabe Bildschirm oder
      Drucker ( b/d ? )"
313: PRINT :PRINT "      'ESC' = Men"
314: GOSUB warten
315: IF a$="b" OR a$="d" THEN b$=a$: GOTO ze12
316: RETURN
317: ze12: bl$="      " : cl$="-Zugänge-----"
      -----"

```

```

318: x=0: CLS
319: FOR i=1 TO bu(27) : x=x+1
320: IF en$(i)=" THEN ze13
321: IF b$="b" THEN PRINT i;" ";en$(i);TAB(25);" =
      ";dt$(i)
322: IF b$="d" THEN LPRINT bl$;en$(i);" = ";dt$(i)
323: IF x=25 THEN GOSUB warten: CLS: x=1 :
      IF a$=CHR$(27) THEN i=bu(27): RETURN
324: ze13: NEXT i
325: IF b$="b" THEN PRINT cl$
326: IF b$="d" THEN LPRINT cl$
327: FOR i=1 TO ta : x=x+1
328: IF te$(i)=" THEN ze14
329: IF b$="b" THEN PRINT te$(i);TAB(25);" = ";
      td$(i)
330: IF b$="d" THEN LPRINT bl$;te$(i);" = ";td$(i)
331: IF x=25 THEN GOSUB warten: CLS: x=1
332: ze14: NEXT i
333: GOSUB warten
334: RETURN
335:
336: fehler:
337: PRINT :PRINT "      Fehlernummer: ";ERR;" ";:
338: IF ERR = 52 THEN PRINT "      Falsche Dateinummer"
339: IF ERR = 53 THEN PRINT "      Datei existiert
      nicht": RESUME NEXT
340: IF ERR = 55 THEN PRINT "      Datei bereits
      geöffnet": CLOSE
341: IF ERR = 57 THEN PRINT "      Datei nicht gefunden"
342: IF ERR = 58 THEN PRINT "      Datei besteht schon"
343: IF ERR = 61 THEN PRINT "      Diskette ist voll"
344: IF ERR = 62 THEN PRINT "      Kein Dateiende
      gefunden"
345: IF ERR = 70 THEN PRINT "      Der Schreibschutz ist
      auf"
346: PRINT
347: COLOR 1,2: PRINT "      Bitte eine TASTE drücken";
      : COLOR 1,0: PRINT
348: WHILE INKEY$="": WEND
349: RESUME menue
350:
351: aendern:
352: CLS : GOSUB menuoff: LOCATE 10,5
353: PRINT "Welche Vokabel soll geändert werden ?"
354: PRINT "      <Return> = Men"
355: INPUT " englisch: "; en$ : IF en$="" THEN RETURN
356: FOR i=1 TO va+1
357: IF en$=en$(i) THEN aen1
358: NEXT i
359: FOR i=1 TO ta
360: IF en$=te$(i) THEN aen2
361: NEXT i
362: PRINT en$;" nicht gefunden": GOSUB warten:
      GOTO aendern
363: aen1:
364: PRINT "      " ; en$(i); " = "; dt$(i)
365: PRINT :PRINT "      Neue deutsche Bedeutung": PRINT
366: INPUT "deutsch: "; dt$(i):ac=1 : i=va+1
367: GOTO aendern
368: aen2:
369: PRINT "      " ; te$(i); " = "; td$(i)
370: PRINT :PRINT "      Neue deutsche Bedeutung": PRINT
371: INPUT " deutsch: "; td$(i):ac=1: i=ta
372: GOTO aendern
373:
374: loeschen:
375: CLS: GOSUB menuoff: LOCATE 10,5
376: PRINT "Welche Vokabeln soll ich löschen ?"
377: PRINT "      <Return> = Men"
378: INPUT " englisch: "; en$ : IF en$="" THEN RETURN
379: FOR i=1 TO va+1
380: IF en$=en$(i) THEN en$(i)="#" : ac=1 : i=va+1:
      va=va-1: GOTO loeschen
381: NEXT i
382: FOR i=1 TO ta
383: IF en$=te$(i) THEN te$(i)="#" : ac=1: i=ta:
      ta=ta-1: GOTO loeschen
384: NEXT i
385: PRINT en$;" nicht gefunden"
386: GOTO loeschen
387:
388: pruefen:
389: IF ta=0 THEN pru2

```

```

390: FOR j=1 TO ta
391: IF te$(j)="" THEN prul
392: IF td$(j)="" THEN prul
393: IF te$(j)="#" THEN prul
394: IF te$(j)=>en$(i) THEN pru2
395: PRINT #2,te$(j)+CHR$(13)+td$(j)
396: te$(j)="" : td$(j)=""
397: prul: NEXT j
398: pru2: RETURN
399:
400: aufnahme:
401: PRINT " Die Vokabel ";
402: COLOR 0,1: PRINT " "en$ " ";: COLOR 1,0:
PRINT " kenne ich noch nicht"
403: auf1: PRINT " Deutsche Bedeutung oder
<Return>: " :PRINT
404: INPUT "deutsch: ";td$(ta):te$(ta)=en$
405: IF td$(ta)="" THEN te$(ta)="#" : td$(ta)="#" :
RETURN
406: ta=ta+1:ac=1
407: IF ta<199 THEN auf3
408: PRINT " Die Datei muß erst gesichert werden"
409: auf2: GOSUB warten
410: auf3: RETURN
411:
412: sort:
413: x=1:y=1
414: sor1: y=y+1: IF y>ta THEN sor3
415: IF te$(x)>te$(y) THEN sor2
416: GOTO sor1
417: sor2: te$(0)=te$(x):te$(x)=te$(y):te$(y)=te$(0)
418: td$(0)=td$(x):td$(x)=td$(y):td$(y)=td$(0)
419: GOTO sor1
420: sor3: x=x+1: IF x=ta THEN RETURN
421: y=x: GOTO sor1
422:
423: eingabe:
424: WINDOW OUTPUT 3
425: CLS
426: ac=1
427: PRINT " Folgende Vokabeln waren unbekannt":
PRINT
428: PRINT " Englisch Deutsch ( Groß/Klein und
Umlaute möglich )"
429: FOR i=tx TO ta-1
430: PRINT :PRINT " ";te$(i);" " :;
INPUT td$(i)
431: IF td$(i)="" THEN te$(i)="#" : td$(i)="#"
432: IF ta>199 THEN ein2
433: NEXT i: RETURN
434: ein2: PRINT " Vor weiteren Zugängen muß die
Datei gesichert werden"
435: GOSUB warten
436: RETURN
437:
438: ende:
439: CLS: LOCATE 10,5
440: IF ac = 0 THEN END
441: PRINT "Bitte erst Datei sichern!"
442: GOSUB warten
443: RETURN
444:
445: text:
446: IF al<>1 THEN al=2: GOSUB sicherheit: RETURN
' Warnung ausgeben v 2.4t
447: CLS :LOCATE 10,5
448: PRINT ,: INPUT" Name der englischen Datei
( Lesen): "; ni$
449: IF ni$="" THEN RETURN
450: OPEN ni$ AS #1
451: IF LOF(1)=0 THEN
452: PRINT :COLOR 2,3: PRINT , " Datei nicht
vorhanden " :; COLOR 1,0 :xx=1
453: ELSE
454: PRINT :COLOR 2,3: PRINT , " Datei ";ni$;
" wird geladen " :; COLOR 1,0
455: END IF:
456: CLOSE #1 : IF xx=1 THEN xx=0: m0=0: m1=0:
RETURN
457: OPEN "r",2,ni$,1
458: ubel:
459: CLS :
460: LOCATE 3,10: PRINT "E S P E R A N T O M A T -
T E X T A U S G A B E"

```

```

461: LOCATE 5,10: PRINT " Sofortbersetzung in die
deutsche Sprache"
462: LINE (0,50)-(664,50),1
463: LINE(0,180)-(670,180),1
464: LOCATE 10,10: PRINT "F1 = B I L D S C H I R M
(gemischte Ausgabe)" : ' a = 1
465: LOCATE 12,10: PRINT "F2 = D R U C K E R
(gemischte Ausgabe)" : ' a = 2
466: LOCATE 14,10: PRINT "F3 = D I S K E T T E
(gemischte Ausgabe)" : ' a = 3
467: LOCATE 16,10: PRINT "F4 = D I S K E T T E
(nur englische Vokabeln)" : ' a = 4
468: LOCATE 18,10: PRINT "F5 = B I L D S C H I R M
(englische Vokabeln ergänzen)" : ' a = 5
469: LOCATE 24,2:PRINT" Bei gemischter Ausgabe werden
englische Vokabeln im deutschen Text mit "
470: LOCATE 25,2:PRINT" ausgegeben. Mit F4 können
noch unbekannte Vokabeln aus dem englischen"
471: LOCATE 26,2:PRINT" Text 'herausgefiltert' werden
. Mit F5 kann diese Datei dann mit "
472: LOCATE 27,2:PRINT" deutschen Texten ergänzt und
der Vokabeldatei hinzugefügt werden"
473: ube21:a$=INKEY$: IF a$<CHR$(128) OR a$>CHR$(133)
THEN ube21
474: a=ASC(a$)-128
475: CLS: LOCATE 10,10
476: PRINT , " OK, es geht los": PRINT :PRINT
477: PRINT : COLOR 2,3: PRINT , " Bitte haben Sie
viel Geduld " : COLOR 1,0 : PRINT
478: IF a=1 THEN ube7
479: IF a=2 THEN LPRINT CHR$(27);"q";CHR$(96);
CHR$(13): GOTO ube7 :
' Elite = 96 Zeichen / Zeile
480: IF a=3 OR a=4 THEN ube3
481: IF a=5 THEN ube7
482:
483: GOTO ubel
484:
485: ube3:
486: PRINT :PRINT ,: INPUT" Dateiname
( Schreiben ): "; no$
487: IF no$=ni$ THEN text
488: IF no$="vok.dat" THEN text
489: IF no$="" THEN text
490: PRINT :PRINT , " Ich schreibe: ";no$
491: OPEN "o",#3,no$
492:
493: ube7: en$="" : ef=0: tx=ta:' Ein Zeichen holen
494: ube8: IF ef=1 THEN ubel4
495: FIELD 2,1 AS a$
496: GET 2
497: IF EOF(2) THEN
498: ef=1: PRINT :PRINT " *** Dateiende
***":' Dateiendezeichen
499: GOTO ubel4
500: END IF
501: IF a$=CHR$(13) THEN ube10
: ' CR ausgeben
502: IF ASC(a$)=39 THEN ube9
: ' Sonderzeichen zulassen
503: IF a$>CHR$(64) AND a$<CHR$(91) THEN a$=CHR$(
ASC(a$)+32) : ' A - Z ==> a - z
504: IF a$<"a" OR a$>"z" THEN ubel10
: ' wenn es kein Kleinbuchstabe #
ist
505: ube9: en$=en$+a$ : IF ef=1 THEN ubel10
: ' en$ auffuellen
506: GOTO ube8
507:
508: ubel10:
509: IF en$=CHR$(39) THEN en$="" : GOTO ube12
: ' chr$(39) ausblenden
510: IF LEFT$(en$,1)=CHR$(39) THEN en$= MID$(en$,2)
: GOTO ubel10
511: IF en$="" THEN ubel12
512: gf=0: FOR i=bu(ASC(en$)-96) TO bu(ASC(en$)-95)
513: IF en$(i)="" THEN i=an: GOTO ubel11
514: IF en$=en$(i) AND a=1 THEN PRINT dt$(i);: gf=2
: i=an : en$=""
515: IF en$=en$(i) AND a=2 THEN LPRINT dt$(i);:
gf=2: i=an : en$=""
516: IF en$=en$(i) AND a=3 THEN PRINT #3,dt$(i);:
gf=2: i=an : en$=""
517: IF en$=en$(i) AND a=4 THEN gf=2: i=an : en$=""

```

```

518: NEXT i
519: ubell1:
520:   IF gf<>0 THEN ubel2
521:   FOR i=1 TO ta
522:     IF en$=te$(i) AND a=1 THEN PRINT td$(i);: gf=2
       : i=ta: en$=""
523:     IF en$=te$(i) AND a=2 THEN LPRINT td$(i);:
       gf=2: i=ta: en$=""
524:     IF en$=te$(i) AND a=3 THEN PRINT #3,td$(i);:
       gf=2: i=ta: en$=""
525:     IF en$=te$(i) AND a=4 THEN gf=2: i=ta: en$=""
526: NEXT i
527: ubel2:
528:   IF en$="" THEN ubel3
529:   IF gf=0 AND a=1 THEN PRINT en$;: en$=""
530:   IF gf=0 AND a=2 THEN LPRINT "*" ;en$;: en$=""
       : ' * = unbekannte Vokabeln
531:   IF gf=0 AND a=3 THEN PRINT #3,"*" +en$;: en$=""
       : ' englisch + deutsch
532:   IF gf=0 AND a=4 THEN PRINT #3,en$;" ";: en$=""
       : ' nur englische Vokabeln
533:   IF gf=0 AND a=5 THEN GOSUB neu: en$=""
       : ' deutsch ergänzen
534:   en$=""
535:   ubel3:
536:     IF a$<>" AND a=1 AND a$<>CHR$(10) AND a$<>CHR$(13) AND ef=0 THEN PRINT a$;
537:     IF a$<>" AND a=2 THEN LPRINT a$;
538:     IF a$<>" AND a=3 THEN PRINT #3,a$;
539:     IF a$=CHR$(13) AND a=1 THEN PRINT
       : ' CR beachten

```

```

540:   IF a$=CHR$(10) AND a=1 THEN PRINT
       : ' Trennzeichen beim Textprogramm
541:   IF a$=CHR$(9) AND a=1 THEN a$=" ";
542:   IF a$=CHR$(32) AND ll>60 AND a=1 THEN PRINT
543:   GOTO ube8
544:   ubel4:
545:   CLOSE
546:   IF a=3 OR a=4 THEN KILL no$+".info"
547:   GOSUB warten
548:   IF a=5 THEN RETURN
549:   IF ta>tx THEN aufnahme
550:   RETURN
551:
552:   neu:
553:   te$(ta)=en$
554:   PRINT :PRINT " Die Vokabel ";
555:   : COLOR 0,1: PRINT " "te$(ta)" ";: COLOR 1,0:
       PRINT " kenne ich noch nicht"
556:   PRINT " Deutsche Bedeutung: (<Return> =
       nicht bersetzen, 'F10' = Ende) :PRINT
557:   INPUT "deutsch: ";td$(ta)
558:   IF td$(ta)=CHR$(138) THEN td$(ta)="#": ef=1:
       RETURN
559:   IF td$(ta)=" THEN RETURN
       : ' nicht uebernehmen
560:   ta=ta+1:ac=1
561:   IF ta<199 THEN neu1
562:   PRINT " Die Datei muß erst gesichert werden"
563:   GOSUB warten
564:   neu1: RETURN

```

KICKSTART SPEZIAL

ESPERANTOMAT

Das Programm übersetzt automatisch englische Texte ins Deutsche. So können komplette Anleitungen (ASCII-Datei) übersetzt werden. Doch auch direkt eingetippte Sätze werden bearbeitet. Durch eine umfangrei-

che Datei von 4500 Vokabeln ist der Großteil des englischen Grundwortschatzes abgedeckt. Neue Vokabeln können jederzeit hinzugenommen werden - das Programm ist lernfähig.

Der Übersetzungskünstler
Englisch-Deutsch

ESPERANTOMAT
DM 19,90 *

KICKUP Die Diskette zum Spiel

KICKUP ist ein rasantes Actionspiel und zugleich der beste Weg, sich über die intimsten Programmieretechniken des AMIGA zu informieren.

Auf dieser Diskette finden Sie neben dem ausführbaren Programm auch den kompletten Sourcecode. (Wo gib's das schon?) Damit können Sie KICKUP als Grundlage zum eigenen Spiel benutzen.

- kompletter Sourcecode des KICKUP-Spiels
- erweiterte ausführbare Version zum direkten Starten
- Viele Sprites (Angreifer, Explosions-effekte) zum Einbinden

MAXON Computer • Industriestr. 26
6236 Eschborn • Tel.: 06196 / 481811

KICKUP - das Spiel:
DM 19,90 *

* zuzüglich Versandkosten
Inland DM 5.-
Ausland DM 10.-

nur Vorkasse möglich

SECOND HAND COMPUTER

Ankauf · Verkauf · Vermittlung · Inzahlung

Gebraucht-Computer:

- Zubehör
- Neugeräte
- alle Marken
- Konkursware
- Ankauf defekter Geräte



NEU:
Jetzt auch
in Kassel
Wir kaufen
und verkaufen:
- Homecomputer
- XT's und AT's
- Büroanlagen

ANGEBOTE:

DD-Disk's 3,5, neu, 10 St. 23,-
HD-Disk's 5,25, neu, 10 St. 18,-
AMIGA u. ATARI ST Laufwerke (NEC), neu 278,-
MANNESMANN MT 81 Drucker, 130 Z/s, NLQ, neu 399,-
BONDWELL Laptop, 8Mhz, 1MB-RAM, 20MB Harddisk,
eingeb. Modem 3798,-

PREISWERTE XT'S UND AT'S!

ACHTUNG: Jeder Kunde, der bei uns einen Einkauf über 2000,- DM macht, erhält kostenlos einen DUPROX-Kopierer (Einzelpreis: 98,-)

ALPHA 2000 GmbH 24 Std. Info: 0 69 - 44 30 00
(vorher: Alpha Team) 6 Frankfurt M. 1 · Ingolstädter Str. 27

ALPHA 2001 GmbH NEUERÖFFNUNG Juni '89
3501 Niestetal (bei Kassel) · Witzenhäuser Str. 10

AMIGA PUBLIC DOMAIN



3500 PD auf 2DD-Disk

5,25" ab DM 1,20

3,5" ab DM 2,60

kostenlose Liste von:

A.P.S. -electronic-
Sonnenborstel 31, 3071 Steimbke

Bei uns werben bringt

GEWINN

Sprechen Sie mit uns.
Heim Verlag ☎ 06151/56057

SPACE SOFT Int.

★★★★ DER AMIGA SHOP ★★★★★

A500 Erweiterung 512 K - 295,-
AMIGA DRIVE 3,5 Zoll!

* Distance * Kein billiger Linearmotor, sondern SYNC-Motor
* Abschaltbar * Durchgesch. Bus * Amigafarben
* 12 Monate Garantie * Markenlaufwerk (NEC, TEAK o.ä.)

Nur: 269,- DM

AMIGA DRIVE 5,25 Zoll!!!

* Leistung wie 3,5 Zoll jedoch zusätzlich:
* 40/80 Track schaltbar * IBM- und AMIGA DOS-kompatibel *

Nur: 299,- DM

Riesiges Zubehörprogramm!!

Komplettes Lieferprogramm gegen 2 DM in Marken!!
* Händleranfragen erwünscht *

SPACE SOFT Int.
R. Wagner, Altewiekring 39
3300 Braunschweig, Tel. 0531/74051

PS: Wir haben und kaufen auch Gebrauchte!!
Public Domain - alle Serien ab 3.90 DM

PRINT & TECHNIK

8000 München 40 Phone (49) (89) 36 81 97
Nikolaustraße 2 FAX (49) (89) 39 97 70

NEU!! ELEKTRONISCHE FILTER!!! DM 298,-
RGB-TRENNER für DigiView und Diamond
Amiga Digitizer jetzt erhältlich
SENSATIONSPREIS DM 298,-

Legen Sie die Filter weg. Mit diesem Zusatz können sie die Bilder von Ihrer FARBKAMERA digitalisieren, in den Computer ablegen, farbig auf dem Schirm darstellen und ausdrucken!! Ein ideales Gerät für jeden Digitizer. Anschlussfertig zwischen VIDEOQUELLE und Digitizer einzusetzen.
Digi View Gold + RGB Filter Paketpreis DM 68,-



AMIGA FLACHBETT A4 SCANNER

Print-Technik Universal **DM 1.198,-**
Der Scanner kann als BILDERFASSUNGSGERÄT/Kopierer und Thermodrucker eingesetzt werden. Die Scandichte beträgt 200 Punkte/Zoll, die Scanzzeit 10 Sekunden. Die Ablage des GANZEN Bildes erfolgt im IFF, die Auflösungen 320 x 200 / 640 x 400 / 320 x 256 / 640 x 512 werden unterstützt. Ausschnittvergrößerungen sind möglich. Komplet mit Software. Binär + 16 Grau Darstellung. Demo DM 10,-

VIDEO TEXT (WELTNEUHEIT)
EMPFANGS-SPEICHER-MODUL DM 298,-
Endlich kann man das VIDEO SIGNAL eines Recorders, ScartTV oder Tuners dazu verwenden den freien Service des Teletextes im IFF oder ASCII Format abzulegen. Empfängt alle Programme und Sie sind über alle Teletextangebote in Europa informiert. Super Grafik-Darstellung.
NEU!!! EURO TIZER/RGB-DIGITIZER DM 498,-
Dieser Digitizer mit Software enthält einen RGB-Trenner. Sofort Farbbild auf dem Schirm.
NEU!!! AMIGA PRO SCANNER DM 2998,-
300 x 300 dpi bis 600 x 600 dpi

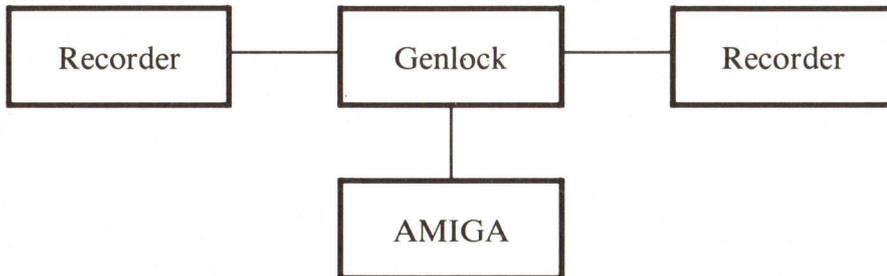
Benelux: 010-450 7696 / NL: 079-41 25 63

OSTERREICH · 1060 WIEN · STUMPERGASSE 34 · TEL. 02 22 / 597 34 23 · TELEX 112 996

SCHWEIZ · MICROTRON · 2542 PIETRELLIN · BAHNHOFSTR. 2 · TEL. 0321872429

Computer & Video

★ **professionell** ★



Suchen Sie Lösungen für professionelle Videoanwendungen auf dem Amiga?

Dann sind Sie bei uns richtig!

Wir statten Sie mit der richtigen Hard- und Software aus. Sei es für das Digitalisieren von Bildern oder das Einblenden von Amiga-Bildern ins Video. Auch die hierfür verwendete Software läßt kaum Wünsche offen.

Lassen Sie sich von Fachleuten beraten! Info-Material und Preise auf Anfrage.

PBC - Peter Biet
Dietershausener Str. 28
D-6409 Friesenhausen
Tel. 0 66 57 / 86 06
Fax: 0 66 57 / 86 05

Musik- und Grafiksoftware Shop
Wasserburger Landstr. 244
D-8000 München 82
Tel. 0 89 / 430 62 07
Fax: 0 89 / 430 41 78

Fa. Zeta Import/Export (Benelux)
Leharstraat 3
NL-2551 La Den Haag
Tel. 0 70 / 68 25 20

RGB-Multiprozessor

- ★ Professionelles Multifunktionsgerät für Videodigitizing und Farbkorrektur
- ★ Videodigitizer kann eingebaut werden
- ★ Drucker-Umschaltbox
- ★ RGB-Splitter, RGB-PAL u. PAL-RGB Wandler, S-VHS-Konverter, Farbprozessor, Überspielerverstärker (Kopierschutzencoder)

Digi-Splitt

- ★ Vollautomatischer RGB-Trenner für Digi-View
- ★ Integrierter Colorprozessor
- ★ Durchgeführtes Videosignal
- ★ RGB-PAL Wandler
- ★ Geschwindigkeitssteigerung von 100 - 150%!

Genlock-Interface

- ★ Die neue Generation in reiner DIGITALTECHNIK
- ★ RGB-Regelung des Computersignals
- ★ Superimposing, Halfbright und Mixbetrieb
- ★ Automatik- und manuelles Fading
- ★ Additiver Keyframe Stanzer
- ★ Invertierung, Horizontalphasenkorrektur
- ★ Integrierter Blackburst-Generator für Stand-Alone Betrieb
- ★ vollbeschalteter RGB-Port

LESERFORUM

Briefe an KICKSTART

Betr.: Probleme mit A1000 - SideCar

Sehr geehrte Damen und Herren, ich besitze einen AMIGA 1000 und habe Ende 88 ein SideCar erworben. Dabei entdeckte ich folgendes Problem:

Vom Sidecar aus läßt sich die parallele Schnittstelle LPT1: nicht ansprechen. Diesen Defekt habe ich mehrfach reklamiert. Der Händler teilte mir mit, daß am SideCar kein Fehler feststellbar sei, man habe es mit mehreren Rechnern ausgetestet. Nun stellt sich mir die Frage, ob die Kombination AMIGA 1000 und SideCar nicht immer mit allen Modellen voll zusammenarbeitet. Haben Sie Erfahrungen darüber, ob dies ein Einzelfall ist oder eine Unverträglichkeit bei einigen AMIGA 1000-Modellen; ist in diesem Fall mit weiteren Schwierigkeiten zu rechnen, z.B. bei einer Festplatte im SideCar?

An einen richtigen Defekt am AMIGA 1000 kann ich eigentlich nicht glauben, da ja sonst alle Funktionen am AMIGA einwandfrei funktionieren.

P. Vorwerk/Pinneberg

Sehr geehrter Herr Vorwerk, die Arbeitsverweigerung des LPT1: hat wahrscheinlich einen sehr einfachen Grund: Falls Sie die Treibersoftware in der Version mit PCBOOT 2.20 benutzen, muß das Programm LPT1S auf dem AMIGA gestartet werden. "LPT1" ab dieser Version arbeitet nur mit dem Bridgeboard zusammen. Dies liegt an der Veränderung von einigen Adressen im DUAL-PORTED-RAM. Die Vorgehensweise zur Verwendung der Parallelport-Emulation ist also folgende:

Kopieren Sie sich das Programm LPT1S von der neuesten PC-Software in Ihren PC-Ordner oder das "LPT1" der original SideCar-Software.

Starten Sie das Programm durch einen Doppelklick vor der Benutzung des Parallelports.

Wollen Sie zwischenzeitlich wieder mit dem AMIGA auf den Parallelport zugreifen, müssen Sie das Close-Gadget des LPT1-Fensters anklicken.

Die Red.



Sehr geehrte Damen und Herren,

ich habe folgende Frage: Da ich einen AMIGA 1000 besitze und dieser leider nur mit 512 kByte ausgerüstet war, habe ich mir eine Speichererweiterung einbauen lassen. Um Kosten zu sparen, wählte ich die "Huckepack"-Methode, bei der die zusätzlichen RAMs auf die vorhandenen aufgelötet und einige Veränderungen an der Adreßlogik vorgenommen werden. Nun zur Frage: Ist ein AMIGA 1000, der wie oben beschrieben umgearbeitet wurde, in der Lage, zusätzlich eine normale externe Speichererweiterung aufzunehmen, oder ist dann das "alte" Megabyte verschenkt?

C. Harnisch/Lohmar

Sehr geehrter Herr Lohmar, Ihre Frage muß man für verschiedene Fälle beantworten.

1. Die externe Speichererweiterung ist Kickstart-autokonfigurierend und liegt im Speicherbereich bei \$C00000.

Diese Speichererweiterung wird mit einer für die Huckepack-Erweiterung modifizier-

ten Kickstart nicht mehr automatisch beim Systemstart eingebunden. Mit dem "ADD-MEM"-Befehl kann der Speicher aber wieder zugänglich gemacht werden. Zu dieser Gruppe von Speichererweiterungen zählen zumeist die preisgünstigeren Modelle mit 1 MB.

2. Die Speichererweiterung unterstützt echtes Autokonfig. Diese Erweiterungen liegen im Bereich \$200000 bis \$a00000. In diesem Falle gibt es keinerlei Einschränkungen beim Betrieb zusammen mit einer internen Erweiterung im Huckepack-Verfahren. Dieser Klasse gehören alle 2 MB- und alle größeren Erweiterungen an.

Grundsätzlich kann man also alle externen Speichererweiterungen verwenden, lediglich beim Komfort sind bei der ersten Lösung Einschränkungen hinzunehmen.

Die Red.



Sehr geehrte Damen und Herren,

Wenn beim AMIGA 500 nach dem Einschalten der Bildschirm grün aufleuchtet und die PowerLED blinkt, was dann?

Des Rätsels Lösung liegt darin, daß das Tastaturkabel nicht richtig auf seinen Platz gesteckt ist. Dies ist mir aufgefallen, als ich meinen A500 aufmachte, um etwas darin zu "erledigen". Nach dem Zusammenbau trat der genannte Fehler auf. Beim Öffnen des Gerätes stellte ich einen falsch plazierten Tastaturstecker fest.

Ich habe noch eine Frage zu Ihrem in Heft 5/89 veröffentlichten Artikel "Kickstart-Umschalter 1.3 und 1.2":

1. Ist das KICKSTART-ROM 1.2 nur 40polig?

2. Ist das KICKSTART-ROM 1.3 also 42polig?
 3. Kann man die Geschichte auch andersrum fabrizieren (ich habe nämlich einen A500 mit Kickstart 1.2 und möchte zwischen 1.2 und 1.3 auch umschalten können)?
 4. Mit wieviel Grad Celsius sollte man an dem Chip löten?
- M. Müller/Öhringen

Sehr geehrter Herr Müller, beide ROMs sind 40polig und werden genau übereinander plaziert. Alle Pins außer Nummer 10 werden verbunden. Natürlich läßt sich ein alter AMIGA (500 o. 2000) so auf Kickstart 1.3 umrüsten. Der LötKolben sollte etwa eine Temperatur von 300-330 Grad aufweisen. Eine kurze Lötzeit ist aber um einiges wichtiger als eine genaue Einhaltung dieser Temperatur. Ein unregelmäßiger LötKolben kann also auch verwendet werden, man sollte halt nicht so lange an den Pins herum" braten" und den ICs nach einigen Pins eine kleine Verschnaufpause zum Abkühlen gönnen.

Die Red.



DIGIVIEW und BUTCHER

Es handelt sich um Probleme des Zusammenwirkens der beiden Programme DigiView 3.0 und Butcher V2.0. Beispielsweise digitalisiere ich mit DigiView ein Videokamera-Bild. Ich speichere es im AMIGA 2000B auf Diskette, sehe es mir wieder an und alles - Konturen, Farben etc. - ist wunderbar. Dabei halte ich mich an die Empfehlungen des Software-Herstellers, also HAM-Modus (4096+), PAL-Format, gegebenenfalls auch Overscan und Interlace.

Nun will man mit solchen Bildern ja in der Regel etwas mehr anfangen, und dazu ist üblicherweise eine Bearbeitung nötig, sei es, um es zu verkleinern, es mit weniger Farben auszustatten (für D-PaintII oder D-Video) oder es in S/W (für Desktop-Publishing) umzuwandeln.

Wenn ich zu solchem Zweck das Bild in Butcher lade, bekomme ich regelmäßig den Requester "Nicht genug Speicherplatz", -

obwohl nach des "Metzgers" eigener Auskunft noch ca. 250 kB Chip-RAM frei sind. Die Folge ist, daß das Bild, wenn überhaupt, gerade noch angenommen wird, aber keine Bearbeitung mehr möglich ist. Wenn ich das Bild auf Lo-Res zu reduzieren versuche, erscheint dann nicht selten der Guru 81000005.

Ein Versuch, sogleich zwecks Verringerung des benötigten Speichers nur mit 26 Farben zu digitalisieren, konnte nicht befriedigen, weil DigiView dabei unklare Farbverläufe, weiße Punkte, verwaschene Konturen etc. produziert.

Was kann ich tun, um das Zusammenwirken der beiden Programme zu verbessern? W. Schneider / Hamburg

Wir haben diesen Versuch noch nicht unternommen, aber vielleicht weiß ein Leser, ob dieses Problem zu beheben ist.

Die Red.



Hallo Martin Silbernel,

FULL SPEED, der Assembler-Test, hat mir so gut gefallen, daß ich ihn sogar zweimal gelesen habe, obwohl ich nicht in Assembler programmiere und es auch nicht vorhaben. Dieser Test beweist, daß man auch mit etwas Fun trockene Tests schreiben kann. Ich jedenfalls könnte mich, wenn ich wollte, dazu entschließen, einen Assembler zu kaufen. Ich bin jetzt ein Fan von Martin Silbernel. Macht bitte weiter so! Ein Autogramm brauche ich nicht, aber ich hoffe für alle Leser, daß ihm die Ideen nicht ausgehen.

M. Krämer / Einhausen

Auch uns gefallen die Testberichte und Artikel von Martin sehr gut und wünschen uns mehr Autoren mit gutem Fachwissen und einem interessanten Schreibstil.

Die Red.



Druckerproblem

Ich habe mir vor kurzem einen Commodore AMIGA gekauft. Nun bekam ich von einem Bekannten einen Thermodrucker der Marke General Electric TXP 1000 (im Handbuch auch als GE 3-8100 bezeichnet).

Ich habe nun einen Probedruck, den ich mit dem Textomat von Data Becker geschrieben habe, beigelegt. Mein Problem ist folgendes: Der Drucker weigert sich, Grafik auszudrucken. Außerdem stimmt der Zeilenabstand nicht, und alle Schriften, die von Grafik abhängig sind, werden verweigert. Welchen Druckertreiber muß ich beim Textomat und bei der Workbench 1.2 installieren?

Beim Drucken mit der Workbench habe ich noch größere Schwierigkeiten als beim Textomat. Diesen Brief habe ich mit dem Druckertreiber "Standard" gedruckt. Ich habe ein Druckerkabel der Firma Lindy (No. 30097) verwendet. Dieses ist am Parallelausgang des AMIGA angeschlossen. Am Drucker ist der Stecker mit dem herausgeführten grünen Kabel angeschlossen.

Reinhard Hauser / Linkenheim-Hochstetten

Da uns der Druckertyp nicht bekannt ist, können wir keine definitive Aussage dazu machen wie er zu einem korrekten Ausdruck zu bewegen ist. Im Prinzip ist der beste Weg, sich an den Hersteller bzw. den Händler zu wenden, bei dem das Gerät gekauft wurde. Dort müssen Erkundigungen darüber eingeholt werden, ob der Drucker zu einem Standard (z.B. EPSON, IBM, NEC) bzw. zu einem Druckertreiber der Workbench kompatibel ist. Diese Angaben erhält man allerdings meistens auch aus dem Handbuch. Dort stehen im allgemeinen auch die Steuer-codes für Fett-, Kursivschrift und die Grafikausgabe. Einige Programme, darunter auch Textomat, benutzen eigene Druckertreiber, die recht einfach geändert werden können und dann auch mit "exotischen" Druckern zusammenarbeiten.

Die Red.



Leserbriefe geben nicht unbedingt die Meinung der Red. wieder. Die Red. behält sich vor, Zuschriften sinngemäß zu kürzen.

Intelligente Texterkennung zum Superpreis

Strukturerkennungsverfahren. Die Erkennung ist deshalb fast unabhängig von Schriftart und -größe. Erkennt ohne weiteres Proportionalchriften (Buchdruck), Schriften ab 8 Punkt. Bis zu 3 zusammenhängende Buchstaben werden erkannt und aufgetrennt.

Optimierungsmöglichkeit durch Lernmodus.

Erkennt mehrspaltigen Satz.

Komplettlösung einschliesslich Texterkennungs- und Malsoftware, A4-Flachbettscanner mit 200 dpi und Amiga-Interface. Scanner kann auch als Kopierer benutzt werden.

Preis nur DM 1.698,- (+ Versandkosten)

Für Schüler, Studenten und Lehrer (Nachweis) 10 % Rabatt.

Novacom GmbH

Helmstrasse 36, 8500 Nürnberg 90, Tel. (0911)339753 oder 372679

512KB RAM Erweiterung

Amiga 500 **A502**

- abschaltbar
- Uhr einfach nachrüstbar
- Megabit-Technologie
- Schnelle RAM's
- nur noch 18% Stromaufnahme im Vgl. zu einer herkömmlichen RAM-Karte

Zum absoluten Knüllerpreis

SOFORT LIEFERBAR

DM **259,-**

3-State Computer Technik

Tel: 02361/492928
02361/179 79

Schaumburgstr. 17 4350 Recklinghausen

Erweiterung für Amiga 1000

von 512k auf 1 MByte

A1002

- intern
- soft- & hardwaremäßig abschaltbar
- läuft problemlos mit allen Erweiterungen (z.B. Sidecar, Festplatten, etc.)

auf Wunsch mit Einbau

DM **398,-**

In Kürze lieferbar: A552
Die Uhrenerweiterung zum Anstecken für ALLE Amiga 500 - RAM-Karten ohne Uhrenlogik.

Versand per Nachnahme + DM 10,-. Es gelten unsere allg. Lieferungs- & Zahlungsbedingungen

Händleranfragen erwünscht

... TELEKOMMUNIKATION ...MODEMS ...

Alle unsere **DISCOVERY MODEMS**

sind HAYES-Kompatibel und können mit einem geeigneten Kabel an nahezu jeden Rechner angeschlossen werden.

Wenn Sie weitere Informationen zu den nebenstehenden Modemtypen wünschen, rufen Sie uns einfach an. Wir senden Ihnen dann umgehend weiteres Informationsmaterial, das Sie auch über unsere besonders günstigen Paketpreise informiert.

AMIGA DFÜ-Paket I: 1200C +, Kabel, Software kompl. DM 298,-
AMIGA BTX-Paket I: 1200C +, MultiTerm, Kabel kompl. DM 398,-

Wir führen auch Steckkarten-Modems f. PC/XT/AT u. Kompatible
Weitere Paketpreise entnehmen Sie bitte unserer Preisliste!

Außerdem führen wir BTX-Software-Decoder für alle gängigen Computersysteme. Die aktuellen Preise und Versionen entnehmen Sie bitte unserer Preisliste oder erfragen sie telefonisch.

MultiTerm Deluxe / BTX-Manager / AMARIS BTX2
134,- 198,- 298,-/448,-

Cameron HANDY-SCANNER inkl. Texterkennung

Typ 2: s/w, 200dpi, deutsches Handbuch DM 498,-

Typ 4: 16 Graustufen, 400dpi, deutsches Handbuch DM 849,-

Turbo Print II DM 89,- AMIGOS DRIVE 3,5" DM 259,-
PD-Buch I, II, III je DM 49,- A2000-Drive 3,5" DM 199,-

48 Stunden Lieferservice für Lagerware per UPS!
Versandpauschale DM 11,40 (O.-b. Auftrag über DM 300,-)

DFÜ-SHOP Norbert Dornhöfer & Michael Böttcher G.b.R. Öffnungszeiten: Mo - Fr 10.00 - 18.30
Kolonnenstraße 33 · 1000 Berlin 62 · Tel./BTX 030 782 71 18
Achtung: Der Anschluß unserer Modems am öffentlichen Telefonnetz der Deutschen Bundespost ist verboten und kann strafrechtlich verfolgt werden

Festplattentreiber

BOIL = 400 KB/S

Bootable OMTI Interface Loader

Boil ist einer der schnellsten Harddisk-Treiber, die für den AMIGA verfügbar sind. Bei Verwendung einer SEAGATE ST 251 mit einem OMTI 5520 erzielen Sie eine Datentransferrate, die größer ist als 400 KB/sec. BOIL arbeitet mit allen Festplatten, die einen OMTI-Controller verwenden.

- DTR größer 400 KB/Sec- Unterstützung aller OMTI's
- **Kompl. Fehlerbehandlung** (ECC Fehlererkennung)
- **Fast-File-System bootfähig** (ohne Bootpartitionen, ab Kickstart Version 1.3, für Kickstart 1.2 in Vorbereitung)
- 2 unterschiedliche Festplatten werden unterstützt (z.B.: 20 + 40 MB).
- Ausführliches deutsches Handbuch
- **Utilities:**
- **Formpart:** Jede Partition ist bootfähig. Nicht bootende Partitionen werden automatisch gemountet.
- **CheckDrive:** Festplattentest
- **CheckInt:** Prüft Interface.
- **PartAccess:** Zugriffsrechte für jede Partition (Lesen, Schreiben, Formatieren)

DM 75.-

TEAC Diskettenlaufwerke NEC

Vollkompatibel, anschlussfertig, abschaltbar, inkl. Kabel, amigafarbenem Metallgehäuse, 2x80 Spuren, alle Laufwerke mit beiger Frontblende. Wir verwenden nur Markenlaufwerke der Firmen **TEAC** (FD 235F oder FD 55 GFR) und **NEC** (1037a). Alle 5.25" Stationen werden mit **40/80 Trackumschaltung** geliefert. Auf alle TEAC Diskettenlaufwerke gewähren wir **12 Monate Garantie**. Aufpreis für durchgeführten Bus: DM 10.-

3.5": 229.- 5.25": 269.-

Festplatten 400 KB/s

Komplett anschlussfertig für AMIGA 500/1000. Das Gehäuse (LxBxH: 32x32x6.5 cm) kann gut als Monitoruntersatz genutzt werden. Weitere Informationen zu dem, im Lieferumfang enthaltenen Boil-Treiber, können Sie nebenstehender Beschreibung entnehmen. Für alle Platten gilt: **leise, Auto-Park, Zugriffszeit < 40 ms, 1 Jahr Garantie.**

32 MB 1198.-

42 MB 1398.-

85 MB 1998.-

Für AMIGA 2000 ermäßigen sich o.g. Preise um DM 150.-

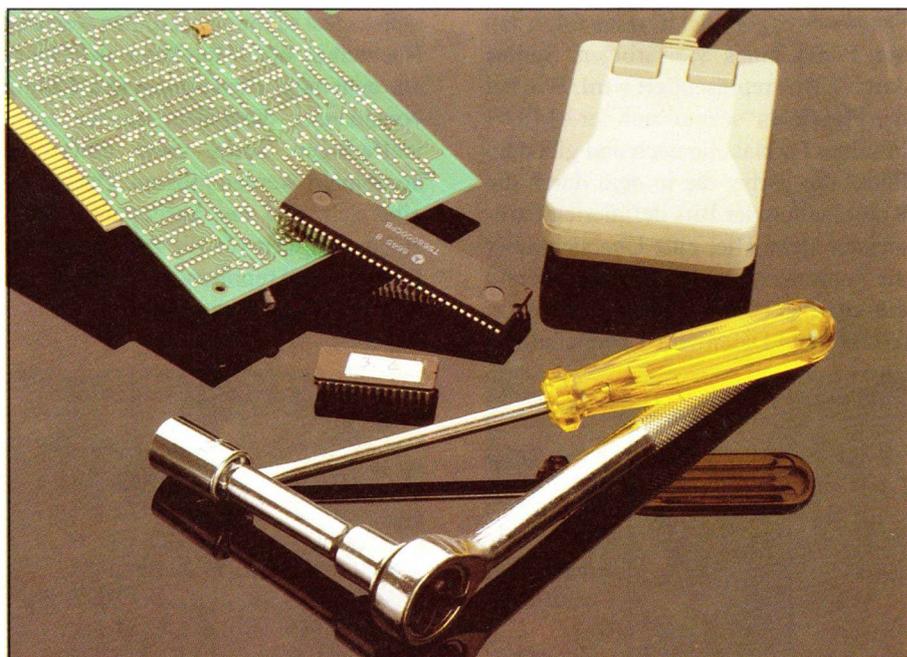
Disketten

3.5" MF 2DD (135 tpi)
Weiße Ware 10 St: 25.-
100 St: 230.-
Mitsubishi 10 St: 29.-
100 St: 270.-

Frank Strauß Elektronik
Schmiedstraße 11
6750 Kaiserslautern
Tel.: 0631 - 67096/97/98
Fax: 60697

Lieferung erfolgt ab Lager mit UPS oder Post per Nachnahme. Versand auch ins Ausland.

Lautlos zieht die STALKER, ein bis an die Antennen mit Extrawaffen bestücktes Raumschiff, ihre Bahn im endlosen Vakuum des unerforschten Sonnensystems, dem die unwissenden Wissenschaftler auf der Erde den Namen AVANON gegeben haben. Plötzlich - für die Radaraugen des Sternenschiffs bisher unsichtbar - erscheinen mehrere Formationen von metallglänzenden Raumschiffen, die sofort gnadenlos das Feuer eröffnen. Von Panik ergriffen versucht der Pilot des herausgeforderten Kreuzers, durch hoffnungslose Ausweichmanöver sein letztes Leben zu retten, um vielleicht doch noch das Ende des Levels zu erreichen, doch es ist aussichtslos - den dumpfen Knall, mit dem sein Raumschiff zu Weltraumschrott zerfällt, hört er schon nicht mehr. Er starrt gebannt auf



Die Hardware- programmierung

einen Schriftzug, der seinen Blick für mehrere Sekunden gefesselt hält.

Noch nie hatte ein GAME OVER ihn schwerer getroffen, obwohl er schon im ersten Level wußte, daß es irgendwann so kommen mußte. Nicht einmal er, der dieses Spiel eigenhändig programmiert hatte, kam über den zweiten Level hinaus - wie, so dachte er bei sich, werde ich jemals verhindern können, daß mein geliebtes Raumschiffsprite getroffen wird? Hilft mir denn keiner? Doch plötzlich - als wäre der Hilferuf sein Stichwort gewesen - begann sein Schreibtisch zu flimmern, und ein Computermagazin erschien. Nachdem er sich die Freudentränen aus dem durch stundenlanges Starren auf den Bildschirm verkrampften Gesicht gewischt hatte, begann er schließlich zu lesen.

Von
Ralf Görlach,
Oliver Siebenhaar,
Sven Stillich, Thorsten Hesse

Teil 3: Wir bringen Bewegung auf den Bildschirm

Grundlagen

In diesem Kursteil soll es um Sprites gehen. Das sind kleine Elemente, die völlig unabhängig voneinander auf dem Bildschirm verwendet werden können. Sprites unterliegen, im Gegensatz zu BOBs (die ja durch den Blitter erzeugt werden), gewissen Beschränkungen. Sie dürfen nur maximal 16 Punkte breit sein, dafür aber die Höhe des gesamten Bildschirms in Anspruch nehmen. Die Anzahl der verfügbaren Farben ist auf vier begrenzt (3 Farben + Hintergrund), kann aber durch spezielle Programmierung auf 16 hochgesetzt werden. Sprites können sowohl vor als auch hinter einem Hintergrund erscheinen. Der Mauszeiger, den wohl alle AMIGA-Besitzer kennen, wird zum Beispiel durch ein

Sprite dargestellt. So, jetzt geht's erst richtig los.

Es wird bunt

Wenn Sie begriffen haben, wie die Farbwahl bei "normalen" Playfields funktioniert, werden Sie mit Sprites keine Schwierigkeiten haben. Ein Sprite stellt eigentlich nur ein Mini-Playfield dar. Die Farben werden vom AMIGA aus der Kombination der Bitplanes berechnet. Der genaue Aufbau eines Sprites wird später erklärt, an dieser Stelle müssen Sie nur wissen, daß eine Spritezeile (16 Punkte) durch ein Word (4 Bytes) dargestellt wird.

Die Farbe des am weitesten links stehenden Punkts des Sprites wird durch die Kombination des 15. Bits der beiden

Datenwörter gebildet. Daraus folgt, daß ein Punkt eines vierfarbigen Sprites durch 2 Bits repräsentiert wird. Wie bei den Playfields "schaut" nun der AMIGA in seiner Farbtabelle nach und gibt dem Punkt die Farbe, die in dem durch die Kombination der Bits definierten Farbregister steht. Hier eine Übersicht über die Zuordnung von Bitkombination und Punktfarben (DW0 bedeutet Datenwort 0, DW1 soll Datenwort 1 bedeuten):

Sprite-Nr.	DW0	DW1	Farbregister
0 & 1	0	0	durchsichtig
	0	1	COLOR17
	1	0	COLOR18
	1	1	COLOR19
2 & 3	0	0	durchsichtig
	0	1	COLOR21
	1	0	COLOR22
4 & 5	1	1	COLOR23
	0	0	durchsichtig
	0	1	COLOR25
	1	0	COLOR26
6 & 7	1	1	COLOR27
	0	0	durchsichtig
	0	1	COLOR29
	1	0	COLOR30
	1	1	COLOR31

Aus dieser Tabelle lassen sich zwei prinzipielle Nachteile von Sprites ablesen. Da wäre zunächst das Problem, daß es keine eigene Farbtabelle für Sprites gibt, das heißt, Playfields und Sprites kommen miteinander in Konflikt oder müssen dieselben Farbwerte benutzen. Da die Sprite-Farben jedoch aus der hinteren Tabellenhälfte (16-31) geholt werden, geschieht dies erst, wenn mit Playfields mit mehr als 16 Farben (4 Bitplanes) gearbeitet wird. Tröstlich, oder? Der zweite, meiner Meinung nach viel gravierendere Nachteil liegt darin, daß immer zwei (aufeinanderfolgende) Sprites die gleichen Farbregister benutzen, so daß eigentlich nur vier verschiedenfarbige Sprites darstellbar sind. Dies läßt sich aber durch geschickte Programmierung des Coppers verhindern.

...und noch bunter

Wem vier Farben nicht ausreichen, der kann die Zahl der möglichen Farben auf 16 heraufsetzen, indem er einen "Multicolorsprite" generiert (Dieser Begriff stammt zwar vom C 64, die Darstellung dieser Sprites unterscheidet sich jedoch stark von der auf dem AMIGA). Doch diese Methode birgt (spritgemäß, kann

man da fast schon sagen) einen weiteren Nachteil in sich: Man kann nicht mehr alle acht möglichen Sprites unabhängig voneinander darstellen, da für einen vielfarbigen Sprite zwei einfarbige kombiniert werden müssen. Daraus ergibt sich, daß ein Spritepunkt durch einen 4-Bitwert (2x die zwei normalen Datenwörter) repräsentiert wird. Hierzu erst einmal die Farbregister:

Sprite-Daten	Farbregister
0 0 0 0	durchsichtig
0 0 0 1	COLOR17
0 0 1 0	COLOR18
0 0 1 1	COLOR19
0 1 0 0	COLOR20
0 1 0 1	COLOR21
0 1 1 0	COLOR22
0 1 1 1	COLOR23
1 0 0 0	COLOR24
1 0 0 1	COLOR25
1 0 1 0	COLOR26
1 0 1 1	COLOR27
1 1 0 0	COLOR28
1 1 0 1	COLOR29
1 1 1 0	COLOR30
1 1 1 1	COLOR31

Es können auf diese Weise jedoch nur die geraden mit einem ungeraden Sprite kombiniert werden, also Nr.0 mit Nr.1, Nr.2 & Nr.3, und so weiter. Die angesprochenen Register sind bei allen Multicolorsprites dieselben (16-31). Wie (werden Sie fragen) erzeugt man denn nun ein Sprite? Im folgenden Absatz wollen wir Ihnen das näherbringen.

Die Sprite Datenlisten

Wie immer geht beim AMIGA nichts ohne Strukturen und Datenlisten. Auch der Aufbau der Sprites wird von einer bestimmten Datenstruktur festgelegt. Am Anfang der Sprite-Datenliste stehen zwei Kontrollwörter, die die Positionen der Sprites auf dem Bildschirm angeben sowie festlegen, ob es sich um ein normales oder ein Multicolorsprite handelt. Sie haben folgenden Aufbau (siehe nächste Spalte).

Dabei bedeuten die Bits x0-x8 die horizontale Startposition des jeweiligen Sprites, also dessen X-Koordinate (HSTART). Die Bits E0-E8 repräsentieren die vertikale Startposition der ersten Zeile des Sprites (VSTART). Wie wir

Erstes Kontrollwort:

Bit-Nummer:	Funktion:
15	e7
14	e6
13	e5
12	e4
11	e3
10	e2
9	e1
8	e0
7	x8
6	x7
5	x6
4	x5
3	x4
2	x3
1	x2
0	x1

Zweites Kontrollwort :

Bit-Nummer:	Funktion:
15	l 7
14	l 6
13	l 5
12	l 4
11	l 3
10	l 2
9	l 1
8	l 0
7	AT
6	0
5	0
4	0
3	0
2	e8
1	l 8
0	x0

schon bei der Bilddarstellung des AMIGA erfahren haben, bedient sich dieser sogenannter DMA-Bausteine, um die Daten für die einzelnen Bitplanes aus dem Speicher zu holen. Nach Erreichen einer bestimmten Bildposition des Rasterstrahls beginnt der AMIGA, per DMA die Bilddaten zeilenweise aus dem Speicher auszulesen und sichtbar darzustellen.

Genau nach dem gleichen Prinzip geschieht die Darstellung der Sprites. Für jedes der Sprites existiert ein eigener DMA-Kanal, was bei einer Anzahl von maximal 8 Sprites 8 DMA-Kanäle ausmacht, die jedoch nicht einzeln an- oder ausgeschaltet werden können. Es existiert nur ein Hauptschalter (SPREN) im DMACON-Register, der angibt, ob Sprite-DMA erlaubt ist oder nicht.

Sprites, die nicht gebraucht werden, müssen also mit leeren Datenlisten versehen werden. Erreicht nun der Raststrahl die X- und Y-Position, die in den Bits H0-H8 und E0-E8 festgelegt ist, so beginnt an dieser Stelle der Sprite-DMA mit der Darstellung des ersten Datenwortes aus der Sprite-Datenliste. Dies ist das Wort, das den beiden Kontrollwörtern folgt. Nach den beiden Kontrollwörtern stehen also die eigentlichen Sprite-Daten, wie sie oben schon beschrieben wurden.

Diese zeilenweise DMA-Übertragung findet solange statt, bis der Raststrahl die Zeile, die die Bits L0-L8 repräsentieren, erreicht hat, also die letzte Zeile des Sprites dargestellt ist (VSTOP). VSTOP errechnet sich also aus der vertikalen Startposition (VSTART) plus der Höhe des Sprites in Zeilen.

Wie Sie festgestellt haben, sind die Bits, die die horizontale und vertikale Position des Sprites bestimmen, etwas unpraktisch über die beiden Kontrollwörter verteilt. Dem Listing können Sie jedoch eine Unterroutine entnehmen, die aus X-,Y-Koordinate und Höhe des Sprites die richtigen Werte für die Kontrollwörter errechnet und diese gleich in die Sprite-Datenliste einträgt. Am Ende der Datenliste müssen zwei Wörter mit dem Inhalt Null stehen, was angibt, daß die DMA-Übertragung hier zu Ende ist. Es gibt noch die Möglichkeit, beliebig viele andere Sprites über diesen DMA-Kanal darzustellen. Wie ist das möglich? Ganz einfach: Nachdem der Raststrahl die Position in VSTOP erreicht hat, wird der DMA-Kanal wieder frei und kann jetzt ein anderes Sprite darstellen, dessen Kontroll- und Datenwörter man an das Ende der ersten Liste hängt. Stehen am Ende einer Sprite-Datenliste also nicht zwei Null-Wörter, weiß der DMA, daß er diese beiden Wörter als Kontrollwörter eines neuen Sprites interpretieren soll.

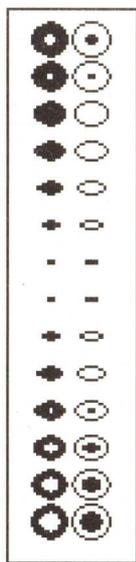


Abb. 2: Alle Bewegungen eines Sprites auf einen Blick

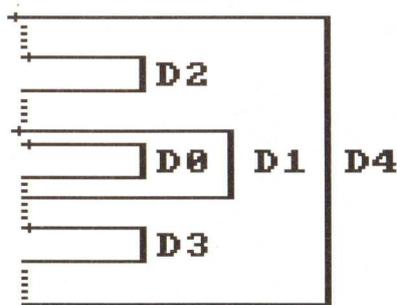


Abb. 1: Die Verschachtelung der Spritebewegung

Es können beliebig viele Sprites über einen DMA-Kanal dargestellt werden. Bedingung ist hierbei nur, daß, auf Grund der Raststrahl-orientierten Darstellung der Sprites die V START-Position des neuen Sprites mindestens um eins größer sein muß als die VSTOP-Position des vorigen Sprites in der Liste, da der DMA ja erst das neue Kontrollwort lesen muß. Das AT-Bit im zweiten Kontrollwort gibt an, ob zwei Sprites miteinander zu einem Multicolorsprite verknüpft werden. Es hat nur in den Sprites mit ungeraden Nummern (1, 3, 5, 7) eine Auswirkung. Ist z.B. für Sprite 1 das AT-Bit gesetzt, so werden dessen Spritedaten mit denen von Sprite 0 als 4 Bit (max. 15 Farben)-Zeiger auf die Farbtabelle interpretiert. Die Reihenfolge der Bits ist dabei wie folgt:

- Sprite 0, erstes Datenwort: Bit 0
- Sprite 0, zweites Datenwort: Bit 1
- Sprite 1, erstes Datenwort: Bit 2
- Sprite 1, zweites Datenwort: Bit 3

Die Bits 0-3 bilden also den erwähnten Zeiger auf die Farbtabelle. Das Verknüpfen zweier normaler zu einem Multicolorsprite funktioniert nur, wenn die Positionen beider Sprites auch übereinstimmen. Ist dies nicht der Fall, wird automatisch die normale Darstellung gewählt. Sie werden sich sicher fragen, wie man dem AMIGA und damit dem Sprite-DMA mitteilt, wo er die Sprite-Datenliste findet. Dazu existieren wie bei den Bitplanes sieben Register (SPR0PTH/SPR0PTL-SPR7PTH/SPR7PTL), in die für jedes Sprite der Anfang seiner Datenliste geschrieben wird. Sie sind wie die Bitplaneregister in High- und Low-Wörter unterteilt. Beginnt der Sprite-DMA mit der Übertragung, wird nach jedem gelesenen Wort der Zeiger in SPRxPTH/SPRxPTL um ein Wort erhöht. Damit in jedem Bild die gleichen Sprites dargestellt werden, muß nach

jedem Bild der Zeiger in SPRx PTH/SPRxPTL wieder auf den alten Wert zurückgesetzt werden. Dies erledigt man am besten innerhalb der vertikalen Auslastlücke mit Hilfe des Coppers. Um die Verwirrung zu steigern, existiert noch ein Register, das die Priorität der Sprites in bezug auf die Playfields angibt - das heißt, wer vor wem erscheinen darf, und wer sich im Hintergrund zu halten hat. Dieses Register trägt den glorreichen Namen BPLCON2 N2. Es hat folgende Aufteilung:

BPLCON2 (\$DFF104):	
Funktion:	Bit-Nummer:
unbel.	15-7
PF2PRI	6
PF2P2	5
PF2P1	4
PF2P0	3
PF1P2	2
PF1P1	1
PF1P0	0

Ist das Bit PF2PRI gesetzt, erscheint Playfield 2 vor Playfield 1. Die Bits PF1P0-PF1P2 (für Playfield 1) und PF2P0-PF2P2 (für Playfield 2) bilden jeweils eine 3 Bit-Zahl, die die Position des Playfields zwischen den Sprites angibt. Zur Festlegung der Prioritäten werden, wie bei der Bildung von Multicolorsprites, jeweils ein Sprite mit gerader und dessen Nachfolger mit ungerader Nummer zu einem Sprite-Paar zusammengefaßt. Der 3 Bit-Wert, der in das Register BPLCON2 geschrieben wird, hat folgende Wirkung auf die Reihenfolge der Sprites und Playfields (siehe Tabelle nächste Seite).

Will man zum Beispiel, daß alle Sprites vor dem entsprechenden Playfield erscheinen, so schreibt man den Wert 4 (%100) in die Bits PFx0-PFx2. Verwendet man nicht den Dual-Playfield-Modus, haben die Bits PLF 2PRI und PL2P0-PL2P2 keine Funktion mehr.

Kollisionen oder der programmierte Unfall

Die Kollisionserkennung zwischen Grafikelementen ist beim AMIGA sehr komfortabel gelöst. Es ist möglich, Kollisionen zwischen Sprites und Sprites, Playfields und Sprites und Playfields

Wert	Reihenfolge von vorne nach hinten				
0	PLF	SPR0/1	SPR2/3	SPR4/5	SPR6/7
1	SPR0/1	PLF	SPR2/3	SPR4/5	SPR6/7
2	SPR0/1	SPR2/3	PLF	SPR4/5	SPR6/7
3	SPR0/1	SPR2/3	SPR4/5	PLF	SPR6/7
4	SPR0/1	SPR2/3	SPR4/5	SPR6/7	PLF

und Playfields zu unterscheiden. Eine Kollision tritt dann auf, wenn sich mindestens zwei Bildpunkte zweier Elemente (Sprites oder Playfields) überlappen, das heißt, die gleiche Koordinate haben. Bei Sprites dürfen dies nur nicht-transparente Punkte sein, bei Playfields läßt sich die Farbe, die zur Kollisionserkennung herangezogen wird, frei einstellen. Im Kollisionsdatenregister CLXDAT wird jede erkannte Kollision gespeichert. Dabei werden wieder jeweils zwei Sprites zu Spritepaaren zusammengefaßt:

planes zur Kollisionserkennung herangezogen werden. Ist das ENBPx-Bit für eine bestimmte Plane gesetzt, ist eine Kollision an den Punkten möglich, die dem Wert in MVBPx entsprechen. Sind alle ENBPx- und MVBPx-Bits gesetzt, ist eine Kollision an allen Punkten mit dem Bitmuster %111111 möglich.

Das Programm

Hier sollen nur zwei Routinen des Programms näher erläutert werden: die Prozedur, die Kontrollwerte der Sprites berechnet, und die Routine zur Erzeugung der Animationsphasen. Der Name der

CLXDAT (\$DFF00E) nur lesen:

Bit-Nummer	Kollision zwischen
15	unbenutzt
14	Sprite 4/5 und Sprite 6/7
13	Sprite 2/3 und Sprite 6/7
12	Sprite 2/3 und Sprite 4/5
11	Sprite 0/1 und Sprite 6/7
10	Sprite 0/1 und Sprite 4/5
9	Sprite 0/1 und Sprite 2/3
8	Playfield 2 (gerade Planes) und Sprite 6/7
7	Playfield 2 (gerade Planes) und Sprite 4/5
6	Playfield 2 (gerade Planes) und Sprite 2/3
5	Playfield 2 (gerade Planes) und Sprite 0/1
4	Playfield 1 (ungerade Planes) und Sprite 6/7
3	Playfield 1 (ungerade Planes) und Sprite 4/5
2	Playfield 1 (ungerade Planes) und Sprite 2/3
1	Playfield 1 (ungerade Planes) und Sprite 0/1
0	Playfield 1 und Playfield 2

Das entsprechende Bit wird gesetzt, wenn eine Kollision aufgetreten ist. Im Kollisionskontrollregister CLXCON läßt sich einstellen, welche Sprites und welche Bitkombination der Playfields zur Kollisionserkennung herangezogen werden.

Die ENSPx-Bits geben an, welches Spritepaar zur Kollisionserkennung herangezogen werden soll. Ist zum Beispiel Bit 15 gesetzt, werden alle Kollisionen der Sprites 6 und 7 in CLXDAT registriert. Es ist so leider nicht möglich, Kollisionen zwischen Sprite 6 und 7 untereinander zu unterscheiden, da diese gemeinsam behandelt werden. Die ENBPx-Bits bestimmen, welche Bit-

Kontrollwortberechnungsroutine (was für'n Wort) lautet sinnigerweise(?) "Kontcal". Schauen Sie sich erst mal selbst die Routine an, und versuchen Sie, sie zu verstehen. HALT! STOOP! Sie haben das Selbstanschauen vergessen. Also lesen Sie bitte noch einmal den vorletzten Satz und ich rate Ihnen, ihn auch zu befolgen, sonst werden Sie nie aus der Endlosschleife herauskommen.

Na, ich will ma' nicht so sein. Die Routine ist so geschrieben, daß man die X-Position in d0 schreibt, die Y-Position (Rasterzeile) in d1 legt und die Länge in d2 schiebt. Außerdem muß in a1 die Startadresse der Sprite-Datenliste sein.

Zunächst einmal werden die Register im Stack gesichert und dann d3 bis d6 gelöscht. Dann wird zu der Länge in d2 die erste Rasterzeile addiert, und wir erhalten die letzte Sprite-Zeile. Mit dem Befehl *move.w d0,d3* erhalten wir in Register d3 die X-Position des Sprites. Mit dem darauffolgenden Befehl (*asr.w #1,d3*) werden die Bits in d3 um eine Position nach rechts geschoben. Dies hat den Zweck, daß d3 so einfach in das Kontrollwort eingesetzt werden kann. Denn in das erste Kontrollwort werden nur die Bits 1 - 8 (x1 - x8) für die X-Position eingetragen. Der nächste Befehl schreibt die Y-Position in d4. Dann wird d4 um 8 Bitpositionen nach links geschoben, um zu erreichen, daß die Bits in d4 mit den Bits e0 - e7 im ersten Kontrollwort übereinstimmen. Dabei wird e8 in das 16. Bit von d4 geschoben. Durch *swap d4* wird erreicht, daß das untere Word mit dem oberen Word von d4 vertauscht wird. Durch *move.w d4,d5* wird das untere Word von d4 und damit auch e8 nach d5 kopiert. Anschließend wird der untere Teil von d4 gelöscht und beide Words wieder vertauscht. Mit *or.w d3,d4* werden d3 und d4 zum ersten Kontrollwort verbunden. Danach wird e8 in d5 auf seine Bitposition gebracht. Die nächsten vier Befehle bringen die Bits für die letzte Sprite-Zeile in ihre Position (10 - 17) und schreiben 18 in d3. *Asl.w #1,d3* schiebt, na, Sie wissen es bestimmt schon, 18 in die Bitposition 1 des zweiten Kontrollwortes (vergleichen Sie dazu die Tabelle auf der nächsten Seite).

Danach wird wieder mit dem *or.w d5,d3*-Befehl e8 und 18 zusammengebracht. *Swap d6* macht die Vertauschung von d6 wieder rückgängig. Mit dem *and.w #1,d0* wird x0 der X-Position isoliert und mit dem nächsten *or.w* mit e8 und 18 vereinigt. *Or.w d3,d6* verbindet x0,18,e8 und 10 - 17 miteinander. *Swap d4* schiebt wieder den unteren Teil von d4 (unser erstes Kontrollwort) in das obere Word. *Or.l* bildet ein Langwort, in dem unsere beiden Kontrollwörter in der richtigen Reihenfolge liegen. *Move.l d6,(a1)* schreibt dann noch unsere eben berechneten Kontrollwörter in die Sprite-Datenliste. Zum Schluß werden die gesicherten Registerinhalte wieder zurück in die Register geschrieben. *Rts* beendet die Unterroutine. Puh! Ich geb' ja zu, daß dies ziemlich (sehr) trocken war, aber wenn Sie mir jetzt

AZTEC-C FÜR AMIGA VERSION 3.6

Wußten Sie, daß eines der verbreitetsten und komplexesten Betriebssysteme - UNIX - in C geschrieben ist ?

Wußten Sie, daß auch das Betriebssystem des AMIGA größtenteils in C geschrieben wurde ?

Wußten Sie, daß C eine der wichtigsten und modernsten Programmiersprachen ist ?

Wußten Sie, daß Aztec-C einer der schnellsten und leistungsfähigsten Compiler für den AMIGA ist ?

Möchten Sie mehr darüber wissen?
Dann schicken Sie uns einen ausreichend frankierten Rückumschlag und Sie erhalten ausführliche Information.

HIERMIT BESTELLE ICH:

- AZTEC-C68K/AM-P
PROFESSIONAL SYSTEM FÜR DM 298,-
- AZTEC-C68K/AM-D
DEVELOPER SYSTEM FÜR DM 598,-
- AZTEC-SDB SOURCE
LEVEL DEBUGGER FÜR DM 149,-

Versandkosten: Inland DM 7,50 Ausland DM 10,-
Auslandbestellungen nur gegen Vorkasse
Nachnahmegebühr DM 3,70

- Vorkasse
 Nachnahme

NAME: _____

VORNAME: _____

STRASSE: _____

ORT: _____

UNTERSCHRIFT: _____



MAXON COMPUTER GMBH
INDUSTRIESTRAßE 26
6236 ESCHBORN
TEL. 06196/481811

CLXCON (\$DFF098) nur schreiben:

Bit-Nummer	Name	Funktion
15	ENSP7	Kollisionserkennung für Sprite 6/7 erlauben
14	ENSP5	Kollisionserkennung für Sprite 4/5 erlauben
13	ENSP3	Kollisionserkennung für Sprite 2/3 erlauben
12	ENSP1	Kollisionserkennung für Sprite 0/1 erlauben
11	ENBP6	Plane 6 mit MVBP6 vergleichen
10	ENBP5	Plane 5 mit MVBP5 vergleichen
9	ENBP4	Plane 4 mit MVBP4 vergleichen
8	ENBP3	Plane 3 mit MVBP3 vergleichen
7	ENBP2	Plane 2 mit MVBP2 vergleichen
6	ENBP1	Plane 1 mit MVBP1 vergleichen
5	MVBP6	Wert für Kollision mit Plane 6
4	MVBP5	Wert für Kollision mit Plane 5
3	MVBP4	Wert für Kollision mit Plane 4
2	MVBP3	Wert für Kollision mit Plane 3
1	MVBP2	Wert für Kollision mit Plane 2
0	MVBP1	Wert für Kollision mit Plane 1

sagen, daß Sie die Routine schon vorher verstanden haben, dann weiß ich nicht, warum ich hier ewig rede (hätt' mir ja mal einer sagen können).

Sprite-Berechnung (THEORIE)

Diese Routine ist ein wahres Meisterwerk, wie wir (ganz unbescheiden) anmerken möchten. Wie Sie sicherlich sehen, sind die Sprite-Daten hinten im Programm abgelegt (*f4col_sprite*, *b_4colsprite*). In der Endversion springt eine sich perspektivisch drehende Metallscheibe in einer sinusförmigen Kurve. Um Ihnen jedoch das Abtippen aller Animationsstufen zu sparen - schließlich lernen Sie ja auch was dabei -, haben wir uns dazu entschlossen, die Daten zu berechnen (wow).

Die eigentlichen Sprite-Daten umfassen nur zwei Halbkreise, einen für die Vorderansicht der Scheiben, einen für die Rückseite. Aus diesem halben Kreis wird eine vollständige Scheibe errechnet (durch Spiegelung an der x-Achse). Soweit, so gut. Um jedoch einen Dreheffekt zu erzielen, wird in der nächsten Animationsstufe die letzte Zeile des Ursprungshalbkreises nicht mehr mit einbezogen. Da er gespiegelt wird, ist die Scheibe nun zwei Zeilen kürzer. Diese zwei Zeilen werden oben und unten wieder als Leerzeilen in den Sprite eingefügt. Diese Prozedur wird zweimal aufgerufen, da für den optischen Effekt Vorder- und Rückseite der Scheibe getrennt berechnet werden müssen. Doch nun zum eigentlichen Programm...

Sprite-Berechnung (PRAXIS)

Zunächst einmal ein Wort zu Abbildung 1. Dort sind - um es ein bißchen übersichtlicher zu machen - die verschachtelten Schleifen der Routine in vereinfachter Form zu erkennen. In Klammern ist immer das Datenregister angegeben, das für den Sprung verantwortlich gemacht werden kann. (Zeitgenossen, die zufällig gerade neben mir sitzen, nennen diesen Programmteil "Crashkurs in Spaghettiprogrammierung des 68000'er" - dabei hat Thorsten selbst daran mitprogrammiert).

Zunächst einmal zu den beiden Variablen "increase" und "decrease". Sie bedeuten einen sehr tiefen Schritt in das 68000'er-Nirwana, da abhängig vom Inhalt dieser Speicherstellen das Programm sich selbst verändert. Assemblerbefehle werden im Speicher durch sogenannte ZAHLEN (Opcodes) abgelegt., genau dieses Phänomen nutzen wir hier geschickt aus. Dadurch sparen wir nicht nur Speicherplatz - das Programm wird auch viel übersichtlicher (räusper). Mit welchen Werten diese Variablen belegt werden und warum, erklären wir, wenn die Zeit reif ist.

Doch jetzt zur Essenz. Das Herzstück unserer kleinen Routine bilden die Schleifen D0-D3. Hier wird nichts anderes getan, als die Spritzeilen der oberen und der unteren Scheibenhälfte zu kopieren, die noch benötigt werden. Außerdem werden - entsprechend der Animationsstufe - oben und unten Leerzeilen im Sprite eingefügt. Die Anzahl der benötigten Zeilen und damit der Anima-

tionsstufe wird durch den Inhalt der Variablen "counter" festgelegt. Insgesamt werden 14 Stufen berechnet. Jetzt wird's kompliziert.

Die Zeit ist reif

Die Prozedur "sprite_anim" berechnet in einem Durchlauf (D0-D3) nur sieben Animationsstufen, die alle von dem Halbkreis der Kreisvorderseite ausgehen. Dann geht eine Metamorphose vor sich - Das Programm verändert sich selbst ("Mami, Mami, mein Programm verändert sich!" - "Mein Kind, das ist schon in Ordnung so."). Um Platz zu sparen, wird, nachdem die obere Hälfte der Scheibe berechnet wurde, der Befehl "MOVE.L (a0),(A1)+" in Zeile xxxx in "MOVE.L -(a0),(a1)+" umgeändert, was die Spiegelung der oberen Scheibenhälfte bewirkt. Dies ist zufällig genau der Wert, den "decrease" beim ersten Aufruf der Routine hat. Wenn Sie mehr über Opcodes wissen wollen, so schauen Sie demnächst nochmal rein ... wenn die nächste Ausgabe erscheint (ask your local dealer for further information). "increase" dreht das Ganze wieder um.

Und nun kommt der Clou: Bisher haben wir nur eine halbe Drehung berechnet. Das Problem, daß sich jetzt stellt, ist die Berechnung der zweiten Hälfte. Schauen Sie sich dazu die Abbildung 2 etwas genauer an (noch genauer!). Die Rückseite der Scheibe muß jetzt nach dem gleichen Schema gedreht, und "rückwärts" in den Speicher geschrieben werden. Zu diesem Zweck verändert sich das Programm ein zweites Mal ("Mami, Mami, mein Programm verändert sich schon wieder!" - "Sei still und lies weiter."). Falls Sie nun darauf warten, daß der Wecker klingelt und Sie aus diesem Alptraum erwachen - keine Chance ("Schriiiiiiiii!" - "Mami?"). Nun läuft alles UMGEKEHRT ab. In der Praxis wird dies erreicht, indem alle "(A1)+" in "-(A1)" umgeändert werden. Lesen die in der nächsten Folge: Die DILETTANTS kehren zurück! (die Fortsetzung des Weltraumdramas) ■

```

1: ; HardwareKurs Teil 3: Spriteprogrammierung
2: ; KICKSTART - written using KICKASS
3: ; (Thorsten Hesse, Sven Stillich, Ralf Görlach,
4: ; Oliver Siebenhaar)
5: ;
6: ; Konstanten/Library Offsets
7: ;
8: Spr0länge = 15
9: Spr1länge = 15
10: OpenLibrary = -552
11: CloseLibrary = -414
12: AllocMem = -198
13: FreeMem = -210
14: execbase = $04
15: SPFix = -30
16: SPFlt = -36
17: Spadd = -66
18: SPMul = -78
19: SPDIV = -84
20: SPSin = -36
21: SPAbs = -54
22: ; CUSTOM Chip Register
23: DIWSTRT = $008E
24: DIWSTOP = $0090
25: DDFSTRT = $0092
26: DDFSTOP = $0094
27: BPL1PTH = $00E0
28: BPL1PTL = $00E2
29: BPL2PTH = $00E4
30: BPL2PTL = $00E6
31: BPL3PTH = $00E8
32: BPL3PTL = $00EA
33: BPL4PTH = $00EC
34: BPL4PTL = $00EE
35: BPL5PTH = $00F0
36: BPL5PTL = $00F2
37: BPL6PTH = $00F4
38: BPL6PTL = $00F6
39: BPL1MOD = $0108
40: BPL2MOD = $010A
41: BPLCON0 = $0100
42: BPLCON1 = $0102
43: BPLCON2 = $0104
44: SPR0PTH = $0120
45: SPR0PTL = $0122
46: SPR1PTH = $0124
47: SPR1PTL = $0126
48: COLOR00 = $0180
49: COLOR16 = $01A0
50: DMACON = $DFF096
51: DMACONR = $DFF002
52: CIAAPRA = $BFE001
53:
54: MOVE.L execbase.s,a6
55: ADDQ.B #1,295(a6) ; Taskswitching sperren
56: ;
57: ; Öffnen der Libraries (TRS,FFP)
58: ;
59: LEA FFPname(pc),a1
60: MOVEQ #0,d0
61: JSR OpenLibrary(a6)
62: BEQ Libclose
63: MOVE.L d0,FFPBase
64: LEA TRSname(pc),a1
65: MOVEQ #00,d0
66: JSR OpenLibrary(a6)
67: BEQ Libclose
68: MOVE.L d0,TRSBASE
69: ;
70: ; Konstanten vorbereiten (mit Matheffp-Library)
71: ;
72: MOVE.L FFPBase(pc),a6; akt. Library wählen
73: MOVEQ #0,d0
74: JSR SPFlt(a6)
75: ; #0 in FLP konvertieren
76: MOVE.L d0,XSummand1
77: MOVE.L d0,XSummand2
78: MOVE.L #200,d0
79: JSR SPFlt(a6) ; #200 in FLP konvertieren
80: MOVE.L d0,Multiplikator; als Multiplikator
81: MOVE.L #300,d0
82: JSR SPFlt(a6) ; #300 in FLP konvertieren
83: MOVE.L d0,d5

```

```

84: MOVEQ #1,d0
85: JSR SPFlt(a6) ; #01 in FLP konvertieren
86: MOVE.L d5,d1
87: JSR SPDIV(a6) ; #01 durch #300 teilen und
88: MOVE.L d0,Step1; Ergebnis in Step1 ablegen
89: MOVE.L #250,d0
90: JSR SPFlt(a6) ; #250 in FLP konvertieren
91: MOVE.L d0,d5
92: MOVEQ #1,d0
93: JSR SPFlt(a6) ; #01 in FLP konvertieren
94: MOVE.L d5,d1
95: JSR SPDIV(a6) ; #01 durch #250 teilen und
96: MOVE.L d0,Step2; Ergebnis in Step2 ablegen
97: ;
98: ; Speicher allocieren
99: ;
100: MOVE.L execbase.s,a6
101: MOVE.L #[[40*256]*5],d0 ; 5 Bitplanes Main
Screen
102: MOVE.L #$10002,d1 ; ChipMem + Clear
103: JSR AllocMem(a6)
104: BEQ Error_End
105: MOVE.L d0,Base_Main
106: MOVEQ #Copsize,d0 ; Größe der
Copperliste
107: MOVE.L #$10002,d1 ; ChipMem + Clear
108: JSR AllocMem(a6)
109: BEQ Error_End
110: MOVE.L d0,Copstart
111: MOVE.L d0,a1 ; Copperlist in Chipmem
112: LEA Newcop(pc),a0 ; kopieren
113: MOVEQ #Copsize-1,d0
114: Copy_Clist: MOVE.B (a0)+,(a1)+
115: DBRA d0,Copy_Clist
116: MOVE.L #Spritesize,d0 ; Größe der
SpriteListe
117: MOVE.L #$10002,d1 ; ChipMem + Clear
118: JSR AllocMem(a6)
119: BEQ Error_End
120: MOVE.L d0,Sprstart
121: MOVE.L d0,a1 ; SpriteList in Chipmem
122: LEA Sprites(pc),a0 ; kopieren
123: MOVEQ #Spritesize-1,d0
124: Copy_Slist: MOVE.B (a0)+,(a1)+
125: DBRA d0,Copy_Slist
126: LEA gfxname(pc),a1 ; graphics library
öffnen
127: MOVEQ #0,d0
128: JSR OpenLibrary(a6)
129: MOVE.L d0,gfxbase
130: ;
131: ; Aufruf der Sprite Berechnungsroutine
132: ;
133: MOVE.W #$22D8,increase ; M.L (a0)+,(a1)+
134: MOVE.W #$22E0,decrease ; M.L -(a0)+,(a1)+
135: MOVE.W #$22FC,b_empty_line ; M.L #$00,(a1)+
136: MOVE.W #$22FC,e_empty_line
137: LEA f4col_sprite(pc),a0 ; = sprite
start (f)
138: LEA animstore(pc),a1 ; spritespeicher
(beg)
139: BSR sprite_anim ; vorderseite berech.
140: MOVE.L #06,counter ; counter ausgangswert
141: MOVE.W #$2318,increase ; M.L(a0)+,-(a1)
142: MOVE.W #$2320,decrease ; M.L -(a0)+,-(a1)
143: MOVE.W #$233C,b_empty_line ; M.L #$00,
-(a1)
144: MOVE.W #$233C,e_empty_line
145: LEA b4col_sprite(pc),a0 ; = sprite
start (b)
146: LEA increase(pc),a1 ; spritespeicher
(end)
147: BSR sprite_anim ; rückseite berechnen
148: ;
149: ; Sprites initialisieren, in Copperlist
eintragen
150: ;
151: MOVE.L Copstart(pc),a0
152: MOVE.L Sprstart(pc),d1
153: MOVE.L d1,d0
154: ADD.L #sprite0_data-Sprites,d0
155: MOVE.W d0,sprite0+6-Newcop(a0) ; Low-Word
156: SWAP d0 →

```

```

157:     MOVE.W d0, sprite0+2-Newcop(a0) ; Hi-Word
158:     MOVE.L d1, d0
159:     ADD.L #spritel_data-Sprites, d0
160:     MOVE.W d0, spritel+6-Newcop(a0) ; Low-Word
161:     SWAP d0
162:     MOVE.W d0, spritel+2-Newcop(a0) ; Hi-Word
163:     ;
164:     ; Main Screen initialisieren
165:     ;
166:     MOVE.L Base_Main(pc), d0
167:     MOVEQ #4, d1 ; 5 Planes
168:     BSR poke_planes
169:     ;
170:     ; Installieren einer neuen Copperlist
171:     ;
172:     MOVE.L gfxbase(pc), a0
173:     MOVE.W #$0080, DMACON
174:     MOVE.L $32(a0), oldcopper
175:     MOVE.L Copstart(pc), $32(a0)
176:     MOVE.W #%1000001110100000, DMACON
177:     ;
178:     ; Hauptprogramm
179:     ;
180:     MOVE.L Sprstart(pc), a2
181: Main: BTST #6, CIAAPRA ; auf Maus-Taste
        prüfen

182:     BEQ Ende
183:     MOVE.L Step1(pc), d0
184:     LEA XSummand1(pc), a0
185:     JSR Sinus(pc)
186:     MOVE.L d0, d1
187:     ADD.L #280, d1 ; y-Offset
188:     MOVE.L #220, d0 ; x-Koordinate
189:     MOVE.W #Spr0länge, d2
190:     LEA sprite0_data-Sprites(a2), a1
191:     JSR Kontcal(pc)
192:     MOVE.L Step2(pc), d0
193:     LEA XSummand2(pc), a0
194:     JSR Sinus(pc)
195:     MOVE.L d0, d1
196:     ADD.L #280, d1 ; y-Offset
197:     MOVE.L #340, d0 ; x-Koordinate
198:     MOVE.W #Sprlänge, d2
199:     LEA spritel_data-Sprites(a2), a1
200:     JSR Kontcal(pc)
201:     ADDQ.L #01, flag
202:     CMP.L #60, flag
203:     BNE.S Main
204:     MOVE.L #00, flag
205:     BSR.S animation ; Sprites animieren
206:     BRA.S Main
207:     ;
208:     ; Unterroutinen
209:     ;
210: animation:
211:     MOVE.L Sprstart(pc), a1
212:     LEA [sprite0_data-Sprites]+4(a1), a1
213:     LEA animstore(pc), a0
214:     ADD.L stufe(pc), a0
215:     MOVEQ #13, d1
216: anim:
217:     MOVE.L (a0)+, (a1)+
218:     DBRA d1, anim
219:     CMP.B #00, ask
220:     BNE.S return
221:     ADD.L #14*4, stufe
222:     CMP.L #12*[14*4], stufe
223:     BLE.S back
224:     MOVE.B #01, ask
225:     BRA.S anim2
226: back:
227:     BRA.S anim2
228: return:
229:     SUB.L #14*4, stufe
230:     TST.L stufe
231:     BEQ.S front
232:     BRA.S anim2
233: front:
234:     CLR.B ask
235: anim2:
236:     MOVE.L Sprstart(pc), a1
237:     LEA [spritel_data-Sprites]+4(a1), a1

```

```

238:     LEA animstore(pc), a0
239:     ADD.L stufe2(pc), a0
240:     MOVEQ #13, d1
241: anim3:
242:     MOVE.L (a0)+, (a1)+
243:     DBRA d1, anim3
244:     CMP.B #00, ask2
245:     BNE.S return2
246:     ADD.L #14*4, stufe2
247:     CMP.L #12*[14*4], stufe2
248:     BLE.S back
249:     MOVE.B #01, ask2
250:     RTS
251: back2:
252:     RTS
253: return2:
254:     SUB.L #14*4, stufe2
255:     TST.L stufe2
256:     BEQ.S front2
257:     RTS
258: front2:
259:     CLR.B ask2
260:     RTS
261: Kontcal: ; Berechnung der Kontrollworte
262:     ; x-> d0, y-> d1, Länge-> d2,
        Spriteptr-> a1

263:
264:     MOVEM.L d0-d7/a0-a6, -(sp)
265:     MOVEQ #0, d3
266:     MOVEQ #0, d4
267:     MOVEQ #0, d5
268:     MOVEQ #0, d6
269:     ADD.W d1, d2 ; letzte zeile des sprites
        nach d2
270:     MOVE.W d0, d3 ; x-position in d3
271:     ASR.W #1, d3
272:     MOVE.W d1, d4 ; erste zeile des sprites
        nach d4

273:     ASL.L #8, d4
274:     SWAP d4
275:     MOVE.W d4, d5
276:     CLR.W d4
277:     SWAP d4
278:     OR.W d3, d4 ; erstes kontrollwort in d4
279:     ASL.W #2, d5
280:     MOVE.W d2, d6
281:     ASL.L #8, d6
282:     SWAP d6
283:     MOVE.W d6, d3
284:     ASL.W #1, d3
285:     OR.W d5, d3
286:     CLR.W d6
287:     SWAP d6
288:     AND.W #1, d0 ; H0 der x-position isolieren
289:     OR.W d0, d3
290:     OR.W d3, d6
291:     SWAP d4
292:     OR.L d4, d6
293:     MOVE.L d6, (a1)
294:     MOVEM.L (sp)+, d0-d7/a0-a6
295:     RTS
296:     ;
297:     ; Sprite Berechnungsroutine
298:     ;
299: sprite_anim:
300:     MOVEQ #00, d3
301:     MOVEQ #06, d4 ; 6 Stufen (7-1 Zeile)
302:
303: copy_loop:
304:     MOVE.W increase(pc), upper_copy ; (a0)+
        sicherstellen
305:     MOVEQ #6, d2 ; anfangswert counter
306:     SUB.L counter(pc), d2 ; - akt. wert counter
307:     TST.L d2 ; test auf null
308:     BEQ.S copy ; ja = keine leerzeile
309:     MOVE.L d2, d3 ; leerzeilen einfügen
310:     SUBQ.L #01, d2
311: b_empty_line:
312:     MOVE.L #$00, -(a1)
313:     DBRA d2, b_empty_line
314: copy:
315:     MOVEQ #01, d1

```

→

AMIGA-GRUNDLEHRGANG

gehört zu jedem Amiga Computer

WICHTIGE MERKMALE:

★ Das Buch für den richtigen Einstieg mit dem Commodore AMIGA ★ Auf über 400 Seiten werden dem Leser leicht verständlich die Grundlagen der Computertechnik und der Umgang mit Hardware erklärt ★ Ein ausführlicher Hauptteil ist dem Einsatz der grafischen Benutzeroberfläche des Betriebssystems gewidmet. Hier erläutert das Buch Fenster, Pull-down-Menüs und die vielen anderen Teile der *Workbench* ★ Wer die Maus nicht mag, der kann aus dem Kapitel über den *Command Line Interpreter (CLI)* entnehmen, wie man den AMIGA auch ohne Maus einsetzen kann ★ Ein weiterer Bereich des Buches ist die Einführung in die Programmiersprache *BASIC*. Eine umfangreiche Befehlsübersicht sowie einige interessante Programme dienen der Erlernung und dem guten Training von *BASIC* ★ Anhänge wie z. B. ein *Index* und eine *Sachworterklärung* bieten das schnelle Nachschlagen und Auffinden wichtiger Punkte ★ Mit dem Buch erhalten Sie eine *Programmdiskette mit allen abgedruckten Listings*. Damit können die *Beispielprogramme* ohne die Mühe und Arbeit des Eintippens auf dem Computer nachvollzogen werden.



Hardcover
Bestell-Nr.
ISBN 3-923250-57-6

59, –
Buch u. Diskette

AUS DEM INHALT:

1. Die Hardware des AMIGA

★ die versch. AMIGA-Modelle ★ die Diskettenstation ★ Anschluß eines Druckers ★ Monitore am AMIGA ★ Erweiterung des AMIGA-Systems ★ Einstieg in die MS-DOS Welt mit dem AMIGA ★ Die „Innereien“ des AMIGA (RAM, ROM u. Prozessoren)

2. Das Betriebssystem des AMIGA

★ Betriebssysteme und ihre Bedeutung ★ Die Benutzeroberfläche des AMIGA ★ Steuerung der *Workbench* ★ Arbeiten mit Maus, Fenstern und Pull-Down-Menüs ★ Verwendung von Disketten, Dateien, Directory ★ Die Programme der *Workbench* Diskette im Einzelnen ★ Der CLI und seine Bedienung ★ Kopieren, Löschen und Batch-Bearbeitung im CLI

3. Programmieren in Amiga-Basic

★ Die Bedienung des Basic-Interpreters ★ Variable in Basic ★ Schleifenstrukturen ★ Die IF-Abfrage ★ Procedures zur Programmstrukturierung ★ Graphik-Programmierung in AMIGA-BASIC ★ Dateiverwaltung ★ ausführliche Befehlsübersicht mit detaillierten Erklärungen

4. Zum Training

★ Programm-Diskette mit allen abgedruckten Listings ★ Sachworterklärung (Fachwörter-Lexikon) ★ Ausführlicher Index (Stichwortverzeichnis mit entspr. Verweisen)

Heim Verlag

Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 061 51-5 60 57

BESTELL-COUPON

an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Ich bestelle _____
zzgl. DM 5,- Versandkosten (unabhängig von der bestellten Stückzahl)
 per Nachnahme Verrechnungsscheck liegt bei

Name, Vorname _____

Straße, Hausnummer _____

PLZ, Ort _____

Benutzen Sie auch die in KICKSTART vorhandene Bestellkarte

```

316: lower_copy:
317:     MOVE.L counter(pc),d0 ; zeilenanzahl
                                (LONG)
318: upper_copy:
319:     MOVE.L -(a0),-(a1) ; Kopieren der oberen
320:     DBRA d0,upper_copy ; Spritehälfte (!)
321:     MOVE.W decrease(pc),upper_copy ; (a0)+
                                in -(a0)
322:     DBRA d1,lower_copy
323:     TST.L d3
324:     BEQ.S cont_copy ; unten gleiche Anzahl
325:     SUBQ.L #01,d3 ; Leerzeilen wie oben
326: e_empty_line:
327:     MOVE.L #$00,-(a1)
328:     DBRA d3,e_empty_line
329: cont_copy:
330:     SUBQ.L #01,counter ; counter - 1 Longword
331:     DBRA d4,copy_loop
332:     RTS
333:     ;
334:     ; Unterprogramm Sinusberechnung
                                (mit Mathematikroutinen)
335:     ; Step-> d0 XSummandptr-> a0 Ergenis-> d0
336:     ;
337: Sinus:
338:     MOVE.L FFPBase(pc),a6 ;akt. Library
                                wählen
339:     MOVE.L (a0),d1 ;akt. X-Koordinate der
                                Sinus-
340:     JSR Spadd(a6) ;funkt. um "Step" erhöhen
341:     MOVE.L d0,(a0) ;und wieder ablegen
342:     MOVE.L TRSBase(pc),a6 ;Spsin liegt im
                                mathtrans.lib
343:     JSR SPSin(a6) ;Sinus von d0 berechnen
344:     ;Ergebnis in d0 ablegen
345:     MOVE.L FFPBase(pc),a6 ;akt. Library
                                wählen
346:     MOVE.L Multiplikator(pc),d1 ;durch eine
                                Multiplikation
347:     JSR SPMul(a6) ;(d0*d1) mit #50 wird der
                                ;Graph in Y-Richtung gedehnt.
348:     JSR SPAbs(a6) ;Betrag bilden
349:     JSR SPFix(a6) ;FFP-Zahl in eine
                                ganzzahlige
350:     ; umrechnen. Ergebnis in d0.
351:     NEG.L d0 ;Ergebnis negieren
352:     RTS ;zurück ins Hauptprogramm
353:
354: poke_planes:
355:     MOVE.L Copstart(pc),a0 ; Bitplanestart-
                                adressen in
356:     LEA planes-Newcop(a0),a0 ; Copperlist
                                eintragen
357:
358: poke:
359:     MOVE.W d0,6(a0)
360:     SWAP d0
361:     MOVE.W d0,2(a0)
362:     SWAP d0
363:     LEA 8(a0),a0
364:     ADD.L #256*40,d0
365:     DBRA d1,poke
366:     RTS
367:     ;
368:     ; Ende des Programmes
369:     ;
370: Ende:
371:     MOVE.L gfxbase(pc),a0 ; alte Copperlist
                                inst.
372:     MOVE.W #$0080,DMACon
373:     MOVE.L oldcopper(pc),$32(a0)
374:     MOVE.W #$8080,DMACon
375:     MOVE.L execbase.s,a6
376:     MOVE.L Base_Main(pc),a1 ; belegten
                                Speicher wieder
377:     JSR FreeMem(a6)
378:     MOVE.L Copstart(pc),a1
379:     MOVEQ #Copsize,d0
380:     JSR FreeMem(a6)
381:     MOVE.L Sprstart(pc),a1
382:     MOVE.L #Spritesize,d0
383:     JSR FreeMem(a6)
384: Libclose:

```

```

385:     MOVE.L execbase.s,a6
386:     MOVE.L TRSBase(pc),a1
387:     JSR CloseLibrary(a6)
388:     MOVE.L FFPBase(pc),a1
389:     JSR CloseLibrary(a6)
390: Error_End:
391:     SUBQ.B #1,295(a6) ; Taskswitching
392:     RTS
393:     ;
394:     ; Konstanten (Librarynames/bases,
                                gesicherte Parameter)
395:     ;
396: gfxname:
397:     DC.B 'graphics.library',0
398: TRSname:
399:     DC.B 'mathtrans.library',0
400: FFPname:     DC.B 'mathffp.library',0
401: TRSBase:     DC.L 0
402: FFPBase:     DC.L 0
403: gfxbase:     DC.L 0
404: Step1:       DC.L 0
405: Step2:       DC.L 0
406: Multiplikator: DC.L 0
407: XSummand1:   DC.L 0
408: XSummand2:   DC.L 0
409: oldcopper:   DC.L 0
410: Copstart:    DC.L 0
411: Sprstart:    DC.L 0
412: Base_Main:   DC.L 0
413: xcoor:       DC.W 8
414: ycoor:       DC.W 3
415: animstore:   BLK.B [2*[14*28]],0
                                ; sprite-speicher
416: increase:    DC.W 0
417: decrease:    DC.W 0
418: counter:     DC.L 6 ; zeilenanzahl-1
419: stufe:       DC.L 0
420: stufe2:      DC.L 4*[14*4]
421: flag:        DC.L 0
422: ask:         DC.B 0
423: ask2:        DC.B 0
424:     ;
425:     ; USER CopperList
426:     ;
427: Newcop:
428:     DC.W DIWSTRT,$2050,DIWSTOP,$2FC8
429:     DC.W DDFSTRT,$0038,DDFSTOP,$00D0
430:     DC.W BPL1MOD,$0000,BPL2MOD,$0000
431: planes:
432:     DC.W BPL1PTH,$0000,BPL1PTL,$0000
433:     DC.W BPL2PTH,$0000,BPL2PTL,$0000
434:     DC.W BPL3PTH,$0000,BPL3PTL,$0000
435:     DC.W BPL4PTH,$0000,BPL4PTL,$0000
436:     DC.W BPL5PTH,$0000,BPL5PTL,$0000
437: sprite0:
438:     DC.W SPR0PTH,$0000,SPR0PTL,$0000
439: spritel:
440:     DC.W SPR1PTH,$0000,SPR1PTL,$0000
441:     DC.W COLOR00,$0000
442: spritecolors:
443:     DC.W COLOR16,$0000,COLOR16+02,$0444
444:     DC.W COLOR16+04,$0888,COLOR16+06,$0CCC
445:     DC.W BPLCON0,$0101001000000000
446:     DC.W $FFFF,$FFFE
447: Copend:
448: Copsize = Copend-Newcop
449:     ;
450:     ; Sprites
451:     ;
452: Sprites:
453: sprite0_data:
454:     DC.W $7088,$7A00
455:     BLK.L 14,0
456:     DC.W 0,0
457: spritel_data:
458:     DC.W $7070,$7A00
459:     BLK.L 14,0
460:     DC.W 0,0
461: Sprlend:
462: Spritesize = Sprlend-Sprites
463: f4col_sprite:
464:     DC.W %00000 0111000000,%0000011111000000
                                ; 2*7 Zeilen = 14

```

```

465: DC.W %000001111110000,%0001100000110000
; (untere Zeilen
466: DC.W %000111111111000,%00100000000001000
; durch Spiegeln
467: DC.W %001111111111100,%0100000000000100
; der oberen 7)
468: DC.W %001111111111100,%0100000000000100
; ANFANGSSPRITE
469: DC.W %011111000111110,%1000001110000010
470: DC.W %011111000011110,%1000011111000010

```

```

471: b4col_sprite:
472: DC.W %0000000111000000,%0000011111000000
; ENDSprite
473: DC.W %000001111110000,%0001100000110000
474: DC.W %000111111111000,%00100000000001000
475: DC.W %0011111101111100,%0100001110000100
476: DC.W %0011110000011100,%0100111111100100
477: DC.W %0111100000011110,%1000111111100010
478: DC.W %0111000000001110,%1001111111110010
479: ; Listing Ende

```

GENLOCK 8802 DM 998,-	SCULPT 3D mit dt. Hb. DM 198,-	NEUI PERFECT VISION	Perfect Sound 14 Seiten DM 9,95
POLAROID PALETTE DM 6750,-	ANIMATE 3D mit dt. Hb. DM 229,-	Color Video Digitizer DM 498,-	Photo Lab 137 Seiten DM 39,95*
<i>mit Imprint neu</i> DM 4500,-	SCULPT 4D DM 898,-	NEUI PAGE STREAM DM 349,-	Photon Paint 73 Seiten DM 24,95
<i>Vorführmodell</i>	3-D Object Disk DM 149,-	Fonts für Page Stream 1-5 je DM 149,-	Pixmate 98 Seiten DM 29,95*
VIDEO EFFECTS 3D in PAL DM 359,-	STUDIO MAGIC DM 159,-	Public Domain Disks: Fish, RPD usw.	Pro Video CGI 37 Seiten DM 14,95
<i>mit dt. Handbuch</i>	<i>mit dt. Handbuch</i> DM 175,-	1 Stk. 4,50 10 Stk. 40,- 50 Stk. 187,50	Video Effects 3D 40 Seiten DM 14,95
3 DEMON DM 215,-	SONIX m.dt.Hb. DM 134,-	Im LOFT Verlag sind bisher die deutschen Übersetzungen folgender Manuals erschienen:	Sculpt 3D 50 Seiten DM 19,95
<i>mit dt. Handbuch</i>	AUDIO MASTER DM 134,-	Vorausgesetzt Sie sind im Besitz des Originals!	Silver m. Upgrade
TURBO SILVER V 3.0 DM 359,-	PERFECT SOUND Stereo DM 179,-	Animate 3D 74 Seiten DM 24,95	Turbo Silver 2.0 109 Seiten DM 34,95*
<i>mit dt. Handbuch</i>	Digitizer m.dt.Hb. DM 188,-	Apprentice Animation 60 Seiten DM 19,95	Sonix 122 Seiten DM 39,95*
THE DIRECTOR DM 139,-	DMCS m.dt. Hb. DM 188,-	Audio Master 48 Seiten DM 19,95	Studio Magic 29 Seiten DM 14,95
<i>mit dt. Handbuch</i>	AZTEC C V3.6 Prof. DM 499,-	Aztec C V3.6 580 Seiten DM 138,-*	Turbo Silver Upgr. 2.0 27 Seiten DM 14,95
MODELER 3D DM 189,-	<i>mit dt. Handbuch</i>	DMCS 59 Seiten DM 19,95	Turbo Silver 3.0 102 Seiten DM 34,95*
<i>mit dt. Handbuch</i>	AZTEC C V3.6 Dev. DM 655,-	Diga 62 Seiten DM 24,95	Videoscape 2.0 65 Seiten DM 24,95
VIDEOSCAPE 3D V 2.0 DM 379,-	<i>mit dt. Handbuch</i> DM 149,-	The Director 69 Seiten DM 24,95	Zing Keys 48 Seiten DM 19,95
<i>mit dt. Handbuch</i>	Source Level Debug. DM 149,-	3 Demon 67 Seiten DM 24,95	* gebunden bzw. incl. Ringordner
PRO VIDEO PLUS DM 569,-	NEUI TOOLKIT DM 79,-	Forms in Flight 93 Seiten DM 29,95*	video LOFT Fiedlerstr. 22 - 32
PROF. DRAW DM 349,-	für THE DIRECTOR DM 79,-	Interchange 26 Seiten DM 14,95	D - 3500 Kassel
<i>mit dt. Handbuch</i>	FARBDIAS von Ihren IFF oder HAM - AMIGA Grafiken:	Modeler 3D 76 Seiten DM 24,95*	Tel.: 0561 - 87 79 28
PXMATE m.dt.Hb. DM 149,-	je DIA 4,95		87 33 99
DIGA m. dt.Hb. DM 159,-	ab 36 Stk. 3,95		
ZING KEYS m dt. Hb. DM 98,-			
3,5" No Name Disks 2DD			
100 Stk. - 199,95 50 Stk. - 104,95			
10 Stk. - 21,95			

WAHNSINN! AMIGA AKTUELL

Hurricane A 2000 Board - komplett bestückt DM 1759,-	A 2000 Plexiglasgehäuse DM 749,-	ST 250R, 5,25", 40 MB inclusive Controller DM 899,-
Animate-Turbo-Board I - III - z. B. Animate T.Board I DM 995,-	2 MB Hurricane Board bestückt mit 100ns. auf Anfr.	Prozessoren:
PC-XT Karte DM 769,-	Seagate Festplatten	Prozessor MC 68020-12E DM 379,-
2 MB Erweiterung DM 1239,-	ST 225,21,6 MB, 65ms DM 469,-	Co-Proz. MC 68881-12B DM 279,-
- 28 Mhz-Board - superschnell DM 2999,-	ST 238R,32,7 MB, 65ms DM 499,-	Co-Proz. MC 68881-16B DM 429,-
Processor Accelerator voll kompatibel über 500 % schneller DM 919,-	ST 251-1,43 MB, 28ms DM 839,-	Co-Proz. MC 68882-16A DM 629,-
	Seagate Festplatten Kits	Amiga Schnecke
	ST 238R, 5,25", 30 MB inclusive Controller DM 799,-	High-Score Killer
		regelt stufenlos bis 0 DM 99,-

S.C.S. SCHOMBURG COMPUTER SYSTEME
Elsflether Str. 16 · 2800 Bremen 1 · Tel. 0421/38 86 67 oder 0421/39 16 78 9 - 12.30 und 15 - 18.30 · Fax: 0421/349 95 17

RHEIN-MAIN-SOFT

Ihr Public Domain-Partner für Amiga

mit über 3000 Disketten aus ca. 50 Serien wie Fish, RPD, Taifun, Chiron, Kickstart, Panorama, Auge usw.

Fish -210	Taifun -100	S.A.F.E. -36a
RPD -164	ACS -160	Franz -25
Auge -33	Faug -75	GERMAN -40 (5.- DM)
Kickstart -170	Cactus -27	usw.

ab 0,80

Preise: 3,5"/5,25"-Diskette(n) Disketten von uns von Ihnen **3,5" 2,60 DM**
0,80 DM 5,25" 1,20 DM

Katalogdisketten gegen 7,00 DM (V/Scheck/Briefmarken) anfordern

Preise zzgl. 5,00 DM Versandkosten b. Vorkasse (8,00 DM b. Nachnahme)

Auch Sonderserien, z. B. wie Amiga-PD-Buch, Buchhaltung, Haushaltsbuch, Etikettendruck, Perfect English usw., Abomöglichkeit

Rhein-Main-Soft · Postfach 39 · 6500 Mainz 32

fischer Leddinweg 14
Hard & Software 3000 Hannover 61
Testen Sie uns 05 11/57 23 58

PUBLIC DOMAIN

kopiert auf COLOSSUS Markendisk

ab 10 Stck. 3.20 incl. 2 DD 3,5" Disk

ab 10 Stck. 2.75 incl. 2 DD 3,5" Disk

ab 10 Stck. 1.50 incl. 2 D 5,25" Disk

!!! Laufend Neuerscheinungen !!!
Versandkosten NN + 8 DM, VK + 4 DM

AMIGA SUPERPREISE!!

ÜBER 2500 Disks im Archiv!

Jede 3,5" PD-Disk **2,30 DM**
ab 200 Stück je 2,20 DM

Wir verwenden nur **errorfreie Qualitätsdisketten!**
Wir liefern: Fish, Chiron, RPD, Poseidon, Kickstart, Auge, Tornado, Panorama, Bordello, Amicus, Faug, Ruhr, Cactus, ACS, Taifun, Franz, RHS, PornoShow, TBAG, SACC und ca. 25 andere Serien!

Leerdisketten 3,5" 2DD 135 TPI	10 Stück DM 19,-
Spielpaket 10 Disketten - ca. 40 PD-Spiele	DM 40,-
Einsteigerpaket für Amiga-Anfänger (Utilities, CLI-Hilfen, Infos usw.)	10 Disk. DM 40,-
Das Superpaket bestehend aus Textverarbeitung, CAD, Haushaltsprogramm, Anti-Virus-Disk usw., alles mit deutscher Anleitung!	15 Disk. DM 55,-
Das Soundpaket: 10 Disketten mit tollen Sonix-Super-Sounds, inkl. Sonix-Player-Disk	DM 40,-
Das Super-Mix (je 5 Disketten aus vorgenannten Paketen)	20 Disk. zum SONDERPREIS von nur DM 69,-

Alle Preise zuzüglich versanospesen. Versandkosten NN 8,- DM - Vorkasse 5,- DM

Tel. 05202/ 71099

RBC-SOFT
I. Güldenpennig, Postfach 1124, 4811 Oerlinghausen

Die Lösung wird mit Hilfe des rechten abgesetzten Ziffernblockes ausgeführt. Jedoch können Sie es genauso mit dem Joystick spielen.

Ich habe aber die Tastatur bevorzugt, da sie sehr viel genauer ist. Viel Spaß bei der Lösung und beim Spielen wünscht Ihnen Chris M.

1. Die Brücke

Nachdem sich unser Held umgedreht hat und dann in das Loch der morschen Brücke gefallen ist, drücken Sie einige Male die 0-Taste (Feuer), bis Sie den Kontrollton (hohes, kurzes Piepsen) genau 2mal gehört haben. In dem Moment, in dem der Ritter zum Schwert greift und es zieht, müssen Sie ein einziges Mal die 8 drücken. Jetzt zieht sich unser Ritter hoch und rennt voller Angst in die Burg hinein.

2. Der Zaubertrank

Hier müssen Sie unbedingt nach rechts gehen. Dies geschieht durch einmaliges Drücken der Taste 6. Nachdem unser Held den Trank einmal kurz genau unter die Lupe genommen hat (oder gab es damals noch gar keine Lupe?), wendet er sich ab, stößt die Tür auf und flüchtet.

3. Die Brücke

Hier geht es ganz genau so wie unter Punkt 1. Jedoch nicht durch die Spiegelverkehrtheit verwirren lassen!

4. Die Wendeltreppe

Dies ist auch wieder ganz einfach! Wenn unser Ritter die Stufen der Wendeltreppe hinuntergeht, drücken Sie, nachdem ein Teil der Treppe geblinkt hat, die Taste, wo sich der Abgrund befindet. Das heißt, wenn sich der Abgrund links von Ihrem Mann befindet, müssen Sie die Taste 4 drücken! Ansonsten stürzt der Ritter schreiend in den Abgrund!!

5. Das grüne Tentakel

Hier muß man sehr gut aufpassen, und es ist auch sehr wichtig, wo sich die Treppe befindet und von welcher Seite der Decke sich der Schwanz des Monsters herabläßt.

Bitte befolgen Sie diese Anweisung nur,

DRAGON'S LAIR

Lösung

wenn sich das Ende des Drachens von der rechten Seite der Decke herabläßt und sich die Treppe in dem Raum links befindet:

1. Wenn sich unser Ritter umgedreht hat und schon das Geheul des Tentakels zu hören ist, ein einziges Mal die 0 drücken.

2. Während der Ritter das Schwert zieht und mit dem Drachen kämpft, auch wieder ein einziges Mal die 8 betätigen und

3. gleich danach, wenn er zum Absprung bereit steht, die 4 drücken. Unser Held springt zuerst nach vorne und dann nach rechts.

4. Jetzt sofort die 2 und danach die 6 drücken. Nun springt er zur Treppe, geht einige Stufen hinauf und springt dann auf den Schemel zurück.

5. Jetzt nur noch die 8 drücken, und alles ist schon fertig.

Wenn sich aber die Treppe rechts befindet und das Schwanzende sich von oben herabläßt, bitte folgende Tasten wie oben geschildert drücken:

0 - 8 - 6 - 2 - 4 - 8, FERTIG!



6. Das Kesselmonster

Dies ist im Gegensatz zum Vorhergehenden wieder sehr, sehr einfach. Nachdem der Kessel umgekippt und eine grüne Masse herausgeflossen ist, sollten Sie dorthin springen, wo sich eine kleine Kommode mit Reagenzgläsern befindet.

Jetzt findet ein Bildwechsel statt, und unserer Mutiger nimmt ein Glas in die Hand. Gleich danach kommt von oben ein weißes Tuch.

Nun kommt wieder ein Bildwechsel, nach dem Sie gleich einige Male die 0-Taste drücken müssen.

Und jetzt zieht er das Schwert, während das Monster mal ordentlich vor dem Mampfi (Essen) gähnt. Nun schlägt das eigentliche Mittagessen (oder vielleicht

etwa schon Abendessen?) das liebe Tierchen in 2 Teile und flüchtet.

7. Die Strudel

Dies ist genauso einfach wie das Kesselmonster. Nach dem Geläute begibt sich unser mutiger Freund auf eine gefährliche Wasserfahrt. Er muß an 4 reißenden Strudeln vorbei. Wenn ein Strudel links

8. Die Wildwasserfahrt

Der Fluß wechselt jetzt die Farbe - zuerst ist er rot und gleich danach blau. Nun sehen Sie eine Höhle und einen blauen Fluß mit einem hellblauen Streifen in der Mitte, der zur Höhle führt. Bei dieser Wildwasserfahrt müssen Sie ziemlich schnell reagieren. Wenn Sie also mit dem Boot von der linken unteren Ecke kommen sollten, dann sofort die 6er-Taste drücken (nach rechts). Gleich danach zwei- oder dreimal nach vorne, d.h. die 8 betätigen. Jauchzend fährt er dann in die Höhle hinein. Gleich danach kommt er aus der anderen Ecke geschossen. Hier muß man genauso verfahren, jedoch statt der 6er-Taste die 4 drücken. Diese Prozedur geschieht auch wieder genauso wie vorher viermal abwechselnd.

9. Der Kugelraum

In diesem Raum muß der Ritter zwischen den bunten Kugeln hindurchrennen, die die Hänge links und rechts von ihm hinauf- und wieder hinabrollen, ohne von der großen schwarzen Kugel hinter ihm zerdrückt zu werden. Um hier hindurchzukommen, erfordert

schon erraten haben - die 2 drücken.

Bei den anderen Kugeln ganz genauso verfahren! Jedoch ist es am besten, Sie drücken die 2er-Taste, wenn eine bunte Kugel Sie gerade noch passiert. Diese Kugel rollt wieder herab und zerschellt in tausend Stücke an der schwarzen Kugel.

10. Der Schachbrettraum

An dieser Stelle ist größte Vorsicht geboten! Hier gibt es den berühmten Schwarzen Mann, der bestimmte Teile des schachbrettartigen Bodens magnetisiert. Man muß mit seinem Mann zu allen Feldern hüpfen, die noch blau und weiß kariert sind, d.h. die nicht magnetisiert wurden. Bitte diese Anweisung nur befolgen, wenn sich das Schwert des Schwarzen Mannes vom Spieler aus gesehen links befindet: RECHTS (6), LINKS (4), VOR (8), LINKS (4) UND SOFORT NACH DEM DRUECKEN DIESER TASTE RECHTS (6), UND DANACH SOFORT LINKS (4), UND SOFORT DANACH RECHTS (6). Nun hört das Magnetisieren auf, und Sie müssen nach VORNE (8) hüpfen. Jetzt sollten Sie sofort nach dem Aufkommen vor dem Schwarzen Mann einige Male FEUER (0) drücken.

Unser wagemutiger Ritter zieht das Schwert, holt kräftig aus und zerteilt den Schwarzen Mann in zwei Teile, die sofort nach dem Zuschlagen verschwinden.

Wenn er das Schwert aber rechts hält, keine Panik hier sind die Zahlen, die genau wie oben befolgt werden müssen:

*4 - 6 - 8 - 6 - 4 - 6 - 4 - 8 und noch 0,
FERDIG!*

Bei diesem Raum braucht man ein wenig Übung, bis man es schafft. Also nicht verzweifeln, wenn es auf Anhieb nicht immer klappt!

11. Das Schlafgemach

Wenn unser Held das etwas karge Schlafgemach der Prinzessin betritt, mauert sich das Zimmer in der Mitte automatisch zu. In dem Moment, in dem die Tür blinkt, ein- oder zweimal die 8 betätigen. Mit ein wenig Schwung springt er gerade noch durch die Steine hindurch. Wenn Sie ihn zu spät springen lassen, werden seine Beine eingemauert, und Sie haben ein Leben verloren.

erscheint, sollten Sie ein- oder zweimal die 6er-Taste drücken. Nun fährt er knapp an dem Wirbel vorbei.

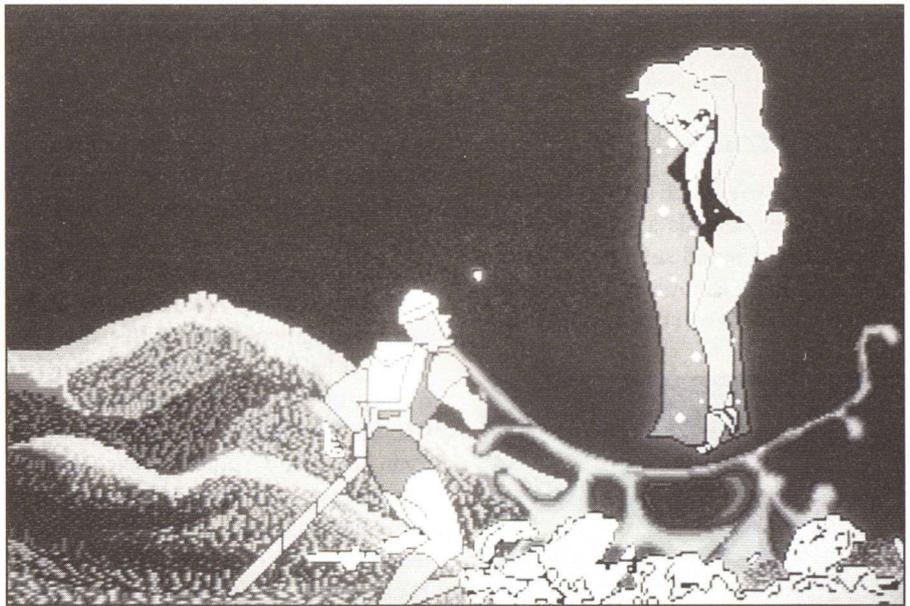
Wenn sich der Strudel an der anderen Seite befindet, dann natürlich die 4 drücken. Pro Durchgang kommen immer abwechselnd vier Wirbel.

es ein relativ genaues Zeitgefühl: Sobald das Bild gewechselt hat und man die schwarze Kugel hinter sich sieht, gleich die 2 drücken. Dies jedoch nur einmal. Wenn jetzt eine bunte Kugel Sie passiert hat und wieder auf der anderen Seite hinaufrollt, sofort wieder - Sie werden es

12. Die Schatzkammer

In dem Moment, in dem Sie den Turm sehen, wie er gerade gefährlich wackelt, müssen Sie zuerst die 4er-Taste drücken.

Jetzt sehen Sie die hübsche Prinzessin in der Kugel, wie sie sich streckt. Sie müssen gleich nach diesem Bild einmal die 6er-Taste drücken und sofort danach nach unten, also die 2! Nun noch einmal die 2 drücken, und jetzt erscheint schon wieder die Prinzessin. Wenn Sie anfängt, irgend etwas davon zu labern, daß Sie das magische Schwert nehmen sollen, einmal die 0 drücken. Jetzt zieht er "the magic sword" aus dem Steinchen.



13. Das grüne Monster Teil 1

Nach dem Einlegen der sechsten Disk sieht man ein Bild mit einer Säule, vor der unser kleiner Held ziemlich verwundert steht. Sobald Sie dieses Bild vor die Augen bekommen, einmal die 2 drücken! Nun sollten Sie auf die Mimik im Gesicht unseres Helden aufmerksam schauen! Wenn dieser nach links schaut, müssen Sie folglich ein einziges Mal nach links, also die 4 drücken, wenn er in die andere Richtung schaut, dann natürlich die 6! Mann, hey, das könnte fast ein neuer Rekord werden, denn nun rennt er geschwind auf das Schwert zu!

14. Das grüne Monster Teil 2

Nun sehen unseren Helden vor dem grünen Drachen stehen. Wenn dieses Bild auf dem Monitor erscheint, sollten Sie die 2er-Taste drücken. Jetzt schlägt er mit dem Schwanz nach Ihnen aus. Dies geschieht noch genau 2x! Jedesmal müssen Sie, gleich nachdem der Drache den Schwanz bewegt hat, die 2 betätigen! Jetzt erschallt ein lautes Tātärätä! Sobald das nächste Bild erscheint, gleich 1- oder 2mal die Feuertaste, also die 0 drücken. Durch das Drücken sammelt Dirk seine ganze Kraft und sticht dann dem Drachen das magische Schwert in den Magen, so daß das Blut wirklich nur so spritzt!

So, jetzt können Sie die Schlußszene mal richtig in Ruhe genießen, also ent-



spannen Sie sich, wischen Sie sich mal die geschwitzten Hände ab und lehnen Sie sich in den Stuhl zurück, drehen Sie den Sound voll auf, und trommeln Sie zum Schluß noch alle Freunde und die ganze Familie zusammen!

Das Aha!

Für alle gestreßten Dragons Lair-Fanatiker biete ich hier einen kleinen Tip, mit dem man sich das ganze Spiel in Ruhe anschauen kann, ohne auch nur einen Finger zu rühren! Nachdem das Titelbild erschienen ist, nicht warten, bis das "Kurzdemo" beginnt, sondern die Ta-

stern drücken. Wenn Sie jedoch die amerikanische Version von Dragons Lair besitzen, dann drücken Sie einfach die Tasten:

Esc + R + L + N + 7 + /

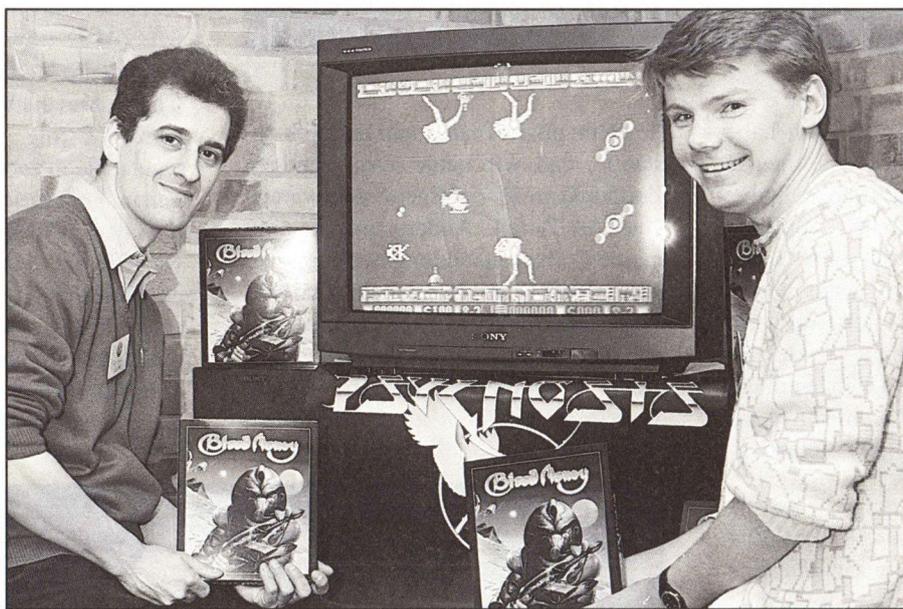
Fertig! Nun blitzt der Bildschirm kurz auf, das heißt, es blinkt einmal kurz. Dann muß noch gesagt werden, daß diese Tasten ca. 2 sec. gleichzeitig gedrückt werden müssen! Nun müssen Sie nur noch Feuer oder die 0-Taste an Ihrem abgesetzten Ziffernblock drücken!

Lehnen Sie sich zurück und genießen Sie das Spektakel!

Esc + R + L + N + 7 + -

BLOOD MONEY

Das neue Action- und Strategiespiel



Als Ende 1988 MENACE erschien, waren viele erstaunt über das neue Programmiererteam DMA DESIGN. MENACE war das erste Spiel des Teams und erschien unter dem PSYCLAPSE-Label von Psygnosis. Der Verkauf war phänomenal - mehr als 15.000 Programme fanden den Weg in die Softwaregeschäfte.

Ein Gespräch mit den Programmierern

VON DEREK DE LA FUENTE

Beide, der Programmierer David Jones und der Graphiker Tony Smith, haben in den letzten sechs Monaten viel dazugelernt. Ihr neuestes Programm BLOOD MONEY zeigt die Fortschritte deutlich.

Inzwischen haben sie das dritte Spiel bereits angefangen, und David perfektioniert gerade Routinen, die erlauben, daß Programmteile während des Spielens geladen werden und so den AMIGA noch näher an die Qualität echter Spielhallenautomaten heranbringen.

Zurück zur Gegenwart und BLOOD MONEY, welches nicht nur ein packendes Schießspiel ist, bei dem zwei Spieler gleichzeitig antreten können, sondern auch sehr beeindruckende Graphik enthält. Die Objekte werden mittels neuer Routinen gesteuert, die David während der letzten Monate erstellt hat. Im folgenden Gespräch mit Derek de la Fuente geben David und Tony ihre eigene Einschätzung der neuen Programme wieder.

David: "BLOOD MONEY ist ein großer Fortschritt gegenüber MENACE. Das erste, was wir getan haben, war die Routinen zu verbessern, die die Bewe-

gemalt. Später bin ich durch einen meiner Freunde auf David gestoßen, und wir haben uns gesagt, daß es besser sei, ein richtiges Programm zu machen. Eigent-

Tony: "Der Job, den ich davor hatte, hat mich angeödet. Sich selbständig zu machen, hat sich ausgezahlt. Empfehlen kann ich das aber wirklich nur Spitzenprogrammierern, und auch dann ist es ein großes finanzielles Risiko. Träume vom "schnellen Geld" sind nicht angebracht. Mir selbst macht die Zukunft keine Angst, denn ich bin bereit, mich neuen Maschinen anzupassen und hart zu arbeiten, um mit den Erfordernissen neuer Computer fertig zu werden. Wir haben auch schon ein Angebot, BLOOD MONEY für einen Spielhallenautomaten umzusetzen. Momentan fühle ich mich allerdings sehr wohl auf dem AMIGA und glaube, daß er noch für mehrere Jahre ein gutes Potential hat. MENACE war in England zwischenzeitlich die Nummer 1 in den AMIGA-Charts, und auch in Amerika war es ein großer Erfolg. Wie gut es sich in Deutschland verkauft hat, weiß ich nicht genau. Zum gleichen Zeitpunkt erschien damals KATAKIS von TLC und dies war ebenfalls ein tolles Spiel in einem ähnlichen Stil und mit ganz phantastischer Musik. Gefreut hat mich allerdings, daß wir in der KICKSTART eine 1 bekommen haben, und das war in diesem Fall die bessere Note. Von der

Bild 1: BLOOD MONEY, Blutiges Geld, nennt sich das neueste Ballerspiel von DMA.



gungen der Objekte steuern. Ich habe dies getan, indem ich mir ein eigenes Graphik-Handling geschrieben habe, das insgesamt 16 Kommandos umfaßt. Die Befehle können auf jedes Sprite oder Bob individuell angewandt werden. So kann ich beispielsweise ein Objekt an eine bestimmte Stelle des Bildschirms gehen lassen, ihm befehlen, sich dort zu teilen, und dann sofort die beiden entstandenen Objekte unabhängig voneinander steuern."

Enorme Bobs in der Größe von 100 x 144 Pixel, die bis zu 18 verschiedene Bewegungen ausführen, und insgesamt mehr als 200 verschiedene Aliens sind ein beeindruckendes Zeugnis für die Richtigkeit von Davids Aussage. Bei DMA Design stimmt aber nicht nur die Programmierung, auch die Graphik ist erstklassig. Tony, der ein alter Freund von David und Derek ist, schildert, wie er den Weg sieht, den er bis heute hinter sich gebracht hat.

Tony: "Ich habe meinen AMIGA vor ungefähr zwei Jahren gekauft. Vorher war ich im Besitz eines Commodore 64. Zeichnen am Computer hat mich vor dem AMIGA überhaupt nicht interessiert. Zwar habe ich vier Jahre Graphik-Design studiert, aber dies war noch vor 1980. Bis zu dem Tag, als ich DPaint 2 auf dem AMIGA sah, konnte ich mir auch nicht vorstellen, daß ein Computer dieser Preisklasse zu einer Qualität von Graphik fähig sei, die mich interessieren könne. Am Anfang habe ich dann Bilder für mich selbst und meine Freunde

lich kam unser Team mehr dadurch zustande, daß wir beide zum richtigen Zeitpunkt an der richtigen Stelle waren - eine Art Zufall. Andererseits hat mit meine Colleagueausbildung doch sehr geholfen, und wenn ich mich im AMIGA-Markt umschaue, wünsche ich mir oft, daß mehr professionelle Graphiker sich dieses Computers annehmen würden. Vom Programmieren selbst verstehe ich

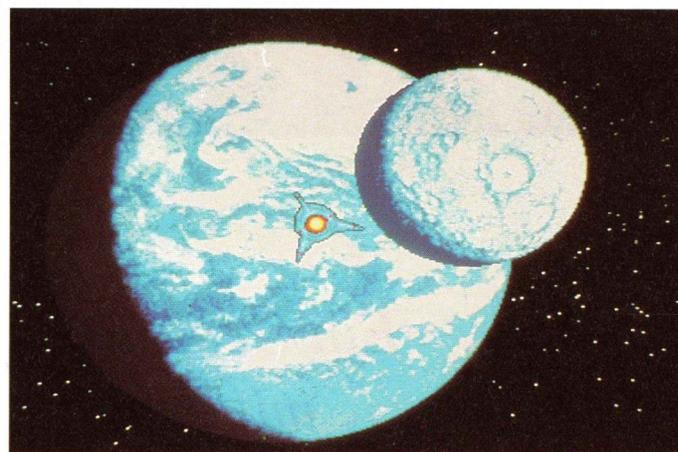


Bild 2: Der Titelscreen von BLOOD MONEY zeigt eindrucksvoll die Qualität der Graphik.

gar nichts. Interessanterweise ist dies aber eher ein Vorteil, denn wenn ich David einige Sprites schicke und ihm sage, wie ich sie bewegt haben möchte, sagt er zuerst einmal, dies sei unmöglich. Aber David liebt Herausforderungen, und meist hat er nach zwei Tagen einen Weg gefunden."

Derek: "Vor einem Jahr habt Ihr Euch selbständig gemacht. Wie hat dies Deinen Lebensstil verändert?"

graphischen Seite kann ich sagen, daß wir inzwischen weiterschritten sind. Selbstverständlich bin ich nie hundertprozentig zufrieden, aber wenn ich nicht gelernt hätte, ein Bild einfach als "fertig" zu erachten, würde ich heute noch über Vorspannbildern sitzen. Mein größtes Vorbild ist übrigens Frank Fazzetta, ein Amerikaner, der Fantasy-Bilder malt. Er hat beispielsweise die Bilder für die Conan-Bücher gemacht. Seine Bücher inspirieren mich beson-

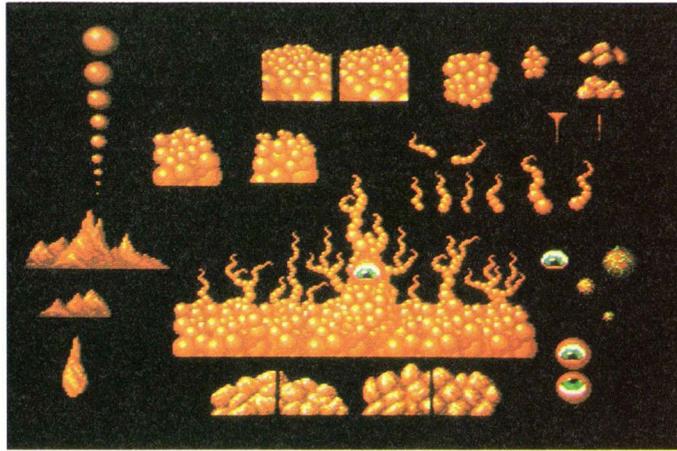


Bild 3: Der Bildschirmrand in Level 3 wurde mit den abgebildeten graphischen Objekten erstellt.

ders bei der Animation von Figuren und Körpern. Das nächste Spiel, das wir machen, wird ein "Hack and Slay Style" sein, so daß wir uns mehr an "Fantasy" anlehnen können. Dies eröffnet mir ein komplett neues Betätigungsfeld, und ich habe jetzt schon sehr viele Ideen, über

die ich allerdings derzeit noch nicht viel verraten kann."

Derek: "Ich habe gesehen, daß Du und David nicht mehr die einzigen im Team von DMA seid."

Tony: "Das stimmt, wir haben noch drei andere Programmierer in unser Entwickler-Team aufgenommen, je einen für PC, C64 und ATARI ST. David und ich halten aber jeder ein Drittel von DMA DEVELOPMENT, wie die Firma jetzt heißt. Wenn ich zurückschaue, so war MENACE ein Versuch, und wir haben sehr viel Glück gehabt, daß es geklappt hat. Ich glaube, daß auch BLOOD MONEY beim Käufer gut ankommen wird, obwohl - oder gerade - weil es sehr viel schwieriger ist als MENACE. Sicher hat bei MENACE auch geholfen, daß es nie eine 100%ig funktionierende "cracked Version" gegeben hat. Der Schutz von BLOOD MONEY ist noch wesentlich besser, und wir hoffen, daß er mindestens vier Wochen hält - man wird sehen."

ddf (Übersetzung:cpl)

MVC
Musik Video Computer

NEU! NEU! NEU!
Ihr Software-Partner

Light's Camera Action 95,00	Holiday Maker 85,00
Carrier Command 69,00	Jeanne d'Arc 55,00
Populoos 65,00	Lombard Rallye 78,00
Driller 74,00	Lord of the rising sun 79,00
Elite 72,00	Super Hang On 79,00
D'Paint III 275,00	Dungeon master 69,95
Fish 69,00	Wallstreet Wizard 55,00

Weitere Superpreise in unserer Preisliste. Bitte kostenlos anfordern.

MVC Musik Video Computer
Hammer Straße 103 - 4730 Ahlen - Telefon (0 23 82) 25 03

DONAU-SOFT
24 h-Schnellversand

Ihr Amiga-PD-Partner
mit ca. 2900 PD-Disk im Archiv
● ab 2,70 DM ●

Alle gängigen Serien sind lieferbar	3 ausführliche Katalogdisketten mit Kurzbeschreibung aller Programme gegen 10,- DM (V-Scheck/Briefmarken) anfordern! gratis zu unseren Katalogen: Viruskiller, CLJ-Wizard + Turbo Backup																								
<table border="1"> <tr> <td>Einzeldisk</td> <td>4,50 DM</td> </tr> <tr> <td>ab 10 Disk</td> <td>4,- DM</td> </tr> <tr> <td>ab 50 Disk</td> <td>3,50 DM</td> </tr> <tr> <td>ab 100 Disk</td> <td>3,30 DM</td> </tr> <tr> <td>ab 200 Disk</td> <td>3,- DM</td> </tr> <tr> <td>bei Serienabnahme:</td> <td>ab 2,70 DM</td> </tr> </table>	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 2,70 DM	<p>Nicht nur für Einsteiger: Das große Amiga-PD-Handbuch</p> <table border="1"> <tr> <td>Band I, II, III</td> <td>je 49,- DM</td> </tr> <tr> <td>Kombi: I + II + III</td> <td>139,- DM</td> </tr> <tr> <td>10 (11) Disks zu I, II, III</td> <td>je 55,- DM</td> </tr> <tr> <td>alle 31 Disks</td> <td>135,- DM</td> </tr> <tr> <td>Ein Band + Disks</td> <td>100,- DM</td> </tr> <tr> <td>alle 3 Bände + alle Disks + 3 Katalogdisketten</td> <td>255,- DM</td> </tr> </table> <p>Amiga Spielebuch 49,- DM</p>	Band I, II, III	je 49,- DM	Kombi: I + II + III	139,- DM	10 (11) Disks zu I, II, III	je 55,- DM	alle 31 Disks	135,- DM	Ein Band + Disks	100,- DM	alle 3 Bände + alle Disks + 3 Katalogdisketten	255,- DM
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 2,70 DM																								
Band I, II, III	je 49,- DM																								
Kombi: I + II + III	139,- DM																								
10 (11) Disks zu I, II, III	je 55,- DM																								
alle 31 Disks	135,- DM																								
Ein Band + Disks	100,- DM																								
alle 3 Bände + alle Disks + 3 Katalogdisketten	255,- DM																								
<p>Preise incl. 3,5" DD-Disks — Mit Qualitätsgarantie — Wir kopieren nur mit doppeltem Verify. Alle Disks sind: — 100 % Virus- und Error frei — etikettiert.</p>																									
<p>Leerdisketten 3,5" 2 DD</p> <table border="1"> <tr> <td>NoName 100 %</td> <td>ab 2,20 DM</td> </tr> <tr> <td>Markendisk</td> <td>ab 2,50 DM</td> </tr> </table>	NoName 100 %	ab 2,20 DM	Markendisk	ab 2,50 DM																					
NoName 100 %	ab 2,20 DM																								
Markendisk	ab 2,50 DM																								
<p>+ DM 5,- bei Vorkasse, + DM 8,- bei Nachnahme Ausland: + DM 10,- (nur Vorkasse)</p>																									
<p>MAIK HAUER Postfach 1401, 8858 Neuburg Fax: 08431/49800 Tel.: 08431/49798 (bis 22 Uhr) BTX: *Donau-Soft #</p>																									

Achtung ★ Achtung ★ Achtung
Amiga-Programmierer

Wir suchen Amiga-Programme aus allen Bereichen (Grafik, Anwendung, Spiele usw.).
Assembler- und C-Programme bevorzugt!
Wir bieten gute Umsatzbeteiligung!

Sollten Sie Interesse haben, wenden Sie sich noch heute an:

SOFTWARE 2000
Software 2000 A. Wardenga
Lübecker Straße 10, 2320 Plön
Tel. 0 45 22/13 79
Fax: 0 45 22/32 96

WUNDERLAND BTX

Hoch auf dem gelben Kabel

Auf der Rückseite Ihres AMIGAs befindet sich eine Anschlußbuchse, der Sie bisher sicherlich noch keine Bedeutung zugemessen haben. Es ist der 'SERIAL PORT'. Auf diese kleine Buchse, die ich selbst erst vor wenigen Wochen entdeckt habe, wird sich in Zukunft das Hauptinteresse verlagern. In unseren letzten KICKSTART-Ausgaben wurde Ihnen unter dem Titel "BTX, was ist das?" BTX-Hardware und BTX-Software vorgestellt.

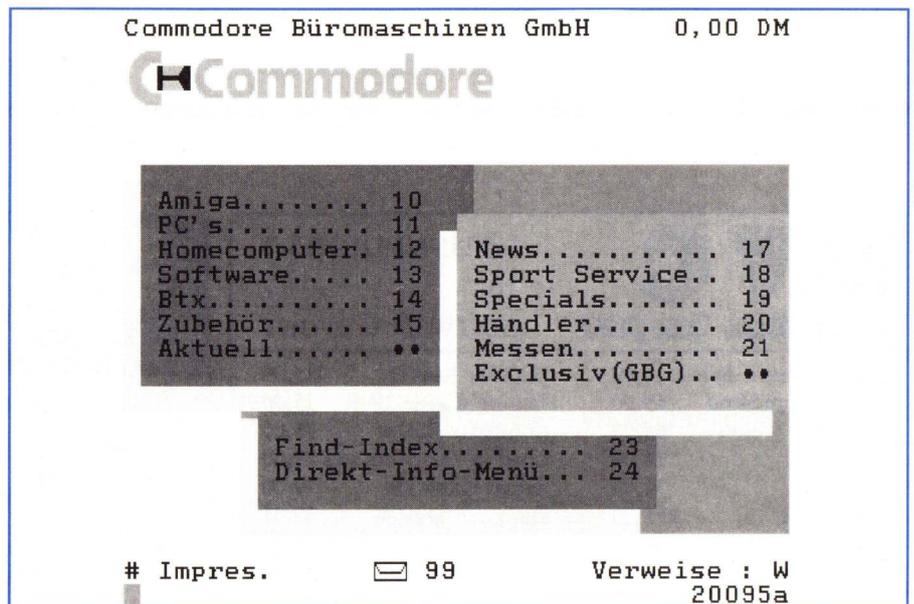


Bild 1: Commodore ist auch über BTX erreichbar.

Damit gelangen Sie problemlos auch in andere öffentliche Datennetze, z. B. in das Telexnetz, Teletexnetz usw. In diesen Netzen erreicht man per Geschäftsverbindung alle Firmen. Man gelangt sogar in die BTX-Netze fremder Länder. Der AMIGA erhält Weltanschluß. Eine kleine Auswahl der Möglichkeiten möchte ich Ihnen näher vorstellen. Folgen Sie mir ins Wunderland der 600 000 BTX-Bilder.

Wir installieren zunächst die Hardware, wie es in den letzten Heften unter dem Titel "BTX, was ist das?" beschrieben

worden ist. Der Koppler muß auf "ORIG, 1200 und I (induktiv)", das Kabel auf "Commodore" eingestellt sein. Dann wird das BTX-Programm gestartet, z.B. das von Commodore. Der Modemtyp ist mit '1200/75' einzustellen. Wir wählen über die Rufnummer '190' oder '01910' die nächstgelegene Bildschirmtext(BTX)-Zentrale an. Sobald ein Pfeifton zu hören ist, muß man den Hörer in die Gummimuschel des Akustikkopplers legen und mit den Tasten 'AMIGA+S' die Datenübertragung starten. Nach kurzer Zeit begrüßt uns die BTX-Zentrale als Gast (Pic 1-4).

Wie orientieren wir uns im BTX-Wunderland?

Aus dem Begrüßungsbild gelangen wir in die BTX-Gesamtübersicht. Hier werden uns das Anbieter-, das Schlagwort- und das Sachgebietsverzeichnis zur Auswahl angeboten. Eine ausführliche Bedienungsanleitung erklärt den Umgang mit BTX. Wenn man noch nichts über die Anbieter weiß und sich auch noch nicht über das Sachgebiet im klaren ist, wählt man als Suchhilfe zuerst das Schlagwortverzeichnis. Hierzu gibt man über die Tastatur die im Anzeigefeld angegebene Wahlziffer ein oder klickt dieses Feld mit der Maus an. Nach Anklicken der entsprechenden Zahlen für die ausgewählten Suchwörter erhalten wir Rufnummern der BTX-Anbieter bzw. weitere Informationen und Auswahlfelder.

Speichern der Bilder

Die gezeigten Bilder können mit den Tasten 'AMIGA+G' als IFF-File gespeichert werden. Reine Texttafeln wie z. B. die Bedienungsanleitung werden besser mit den Tasten 'AMIGA+E' als reine ASCII-Dateien gespeichert. Diese Textdateien können Sie später mit jedem Textprogramm laden und weiterbearbeiten. So lassen sich auf mehrere Tafeln verteilte Texte zu einem Text zusammenfassen. Das Format der Bilder umfaßt 24 Zeilen mit je 40 Buchstaben. Durch die Bearbeitung mit einem Textprogramm können Sie die Datei auf eine beliebige Zeichenanzahl pro Zeile umformatieren.

Sofortdruck der Bilder

Mit den Tasten 'AMIGA+P' erhalten wir einen Sofortausdruck des Textteils der BTX-Bilder. Nun kann die Reise losgehen, die Möglichkeiten sind nahezu unbegrenzt. Wie in einem Warenhaus können Sie sich in allen Abteilungen ausgiebig umsehen. Das Angebot ist so überwältigend, daß unsere Reise nur einen kleinen Einblick in das große Wunderland BTX vermitteln kann. Als Gast brauchen Sie sich zunächst nur die gebührenfreien Bilder anzusehen. Jedoch können Sie dann noch keine Bestellungen absenden.

Deutsche Bundesbahn				0,00 DM
	STÄDTEVERBINDUNGEN			08.05.89
HAMBURG - MÜNCHEN				
AB	ZUG	AN	BEMERKUNGEN	
6.25	IC 581	13.06		
7.17	FD 1981	15.01	X	
7.25	IC 583	14.06		
8.02	FD 1971	15.01	(Y) @Hann FD X	
8.25	IC 681	15.06	@Würzb IC	
8.25	IC 681	15.28		
9.25	IC 783	16.06		
10.27	EC 91	17.06	@Würzb EC	
11.27	IC 785	18.06		
— einzige Seite —				
vorherige Seite *# Rückfahrt 1 weiter #				
zurück 0 Bedienhilfe *1# Inhalt *8#				
Übrigens - haben Sie schon an Ihren				
— Rail&Road-Mietwagen gedacht? —				
Reservieren können Sie ihn sich über				
— Telefon (0130)2211 zum Ortstarif —				
1200833401083a				

Bild 2: Fahren Sie gerne mit der Bahn...

AMIGA Btx Terminal		Online		
Lufthansa		0,00 DM		
Flugverbindungen am FR 05.05.89				
Von : HAMBURG				
Via : MÜNCHEN/MUNICH				
Nach: TUNIS				
Check-in: 20 Min Zeitunterschied: + 0.00				
Ab	An	Flugnummer	Typ Cl. Stop	
08.30-09.50		LH@		0
11.25-13.35		LH1972	737 F/C	0
737 Boeing 737 Jet				
F/C First/Business Class				
Fluggesellschaften 31		Sonderzeich. 32		
Rückflug 21		Vortag 22		
Neue Suche 0		Dialog-Ende *#		
50000620a				

Bild 3: ...oder fliegen Sie lieber mit dem Flugzeug?

Schauen wir zunächst einmal bei *commodore# rein. Sie sehen hier, wie Commodore anzuwählen ist. Zunächst geben Sie einen Stern (*) auf dem Ziffernblock, dann den Namen und zum Schluß das Nummernzeichen (#) ein, also *commodore#. Hier erfahren wir Neuigkeiten über den AMIGA, über BTX und die dazugehörige Software. Das wird sicher Ihr Interesse finden. Mit '*0#' gelangen wir jeweils ins Ausgangsbild zurück. Möchten Sie verreisen? Die *bundesbahn# nennt Ihnen die günstigsten Bahnverbindungen. Wenn Sie die gewünschte Abfahrtszeit in das Formular eintragen, erhalten Sie nach kurzer Zeit einen Fahrplan nach Ihren eigenen Bedürfnissen. Diesen können

Sie sich bekanntlich mit den Tasten 'AMIGA+P' sofort ausdrucken lassen. Haben Sie es sich anders überlegt? Geduldig nennt Ihnen BTX immer die jeweils günstigsten Zugverbindungen. Möchten Sie lieber fliegen? Die *lufthansa# nennt Ihnen die aktuellen Flugpläne. Die *swissair# schaltet sofort durch ins Rechenzentrum Zürich und gibt Ihnen von dort aus Hinweise auf die Flugpläne. Reisebüros machen Ihnen Angebote. Für Kurzentschlossene werden stark ermäßigte Flugreisen aus Angebotsüberschüssen oder Stornierungen angeboten. Wollen Sie einkaufen? Zahlreiche Versender, z. B. *conrad#, *quelle#, *neckermann#, *otto# bieten Ihnen die übliche Katalogware an

und unterbreiten stets aktuelle Sonderangebote. Als Gast sind Sie jedoch für die Bestellung noch "gesperrt", was Sie nicht in jedem Falle gleich erkennen können. Der Anbieter aber erkennt an Ihrer Gastken- nung, daß Sie noch kein eingetragener BTX-Kunde sind.

Für Autofahrer bie- tet u.a. der *adac# seinen Service an: Besonders wichtig sind die stets aktua- lisierten Straßenzu- stands- und Ver- kehrsberichte. Der Auslandsurlauber ist dankbar für Tips und Informationen aus dem Ausland.

Natürlich sind Sie ein Computerfreak. Der *wdt# stellt auf zahlreichen Bildern seinen Computerc- lub vor. Möchten Sie spekulieren? Die Frankfurter Wertpapierbörse *672 43# meldet sich mit einem hübs- chen Bild. Die Stiftung Warentest *222000# veröf- fentlicht ständig Testergebnisse.

Am 'Spiel und Spaß' ist das Angebot überwältigend, selbst einfache Zeichentrickfilme (Animationen) können Sie sich holen. Selbstverständlich öffnet BTX Ihnen den Zugang zu anderen öf- fentlichen Fernmeldenetzen.

Telexnetz

Der Übergang von BTX auf Telex mit- tels Eingabe *1051# ist sehr kostengün- stig. Sie benötigen keinen teuren Fern- schreiber zum Preis von über 10 000 DM. Der AMIGA, der Drucker und wei- teres Zubehör sind Ihre Telexendgeräte.



Bild 4: Der WDR Computer-Club bietet dem User eine ganze Menge.



Bild 5: Spekulieren Sie gerne? Die Frankfurter Wertpapierbörse gibt Ihnen Auskunft.

Über das Auswahlm- enü können Sie auch das Ausland erreichen. Danach erscheint die eigent- liche Textseite. In diese tragen Sie die Telexnummer des gewünschten Telex- teilnehmers ein und den Mitteilungstext. Unmittelbar nach- dem Sie den Text abge- schickt haben, springt beim angege- benen Telexteilnehmer (z.B. in Japan) der Fernschreiber an und drückt den von Ihnen einge- gebenen Text aus. Natürlich sind Sie über das Telexnetz auch erreichbar. Die Mitteilungen anderer Telexteilnehmer an Sie werden Ihnen bei Ihrer nächsten Verbindung mit der BTX-Zentrale auf dem Bildschirm angeboten.

dieses Beitrags sprengen. Treten Sie doch selber die Reise ins BTX-Wunder- land an. Sie kostet wenig und bringt sicherlich viele erfreuliche Stunden. Das Programm können Sie mit den Tasten 'AMIGA+Q' verlassen.

Möchten Sie Kontakt mit gleichgesin- nten Computerfans aufnehmen? Dann schalten Sie das Kabel auf 'Terminal' und den Koppler auf '300'. Mit einem der vielen Terminalprogramme (siehe KICKSTART-PD) können Sie Kontakt mit Mailboxen aufnehmen. Darüber hat die 'KICKSTART' ja schon in älteren Ausgaben ausführlich berichtet. ■

Cityruf

Auch der neue Dienst Cityruf ist über BTX erreichbar. Sie können eine Nachricht von ma- ximal 80 Zeichen an jeden Cityrufteil- nehmer versenden. Diese Nachricht erscheint auf dem Display eines klei- nen Taschenfunk- empfängers beim angerufenen City- rufteilnehmer. Der Cityruf ist mit dem europäischen Funk- rufdienst vergleich- bar. Er ist jedoch auf dem jeweiligen Stadt- bereich beschränkt.

Darüber hinaus gibt es den Zugang zum Teletexnetz, zu den Dutex-P- und Da- tex-L-Netzen und zu den Telefaxein- richtungen im öf- fentlichen Fern- sprechnetz.

Die BTX-Netze der Nachbarländer sind problemlos über das Auswahlm- enü er- reichbar.

Es gäbe über BTX noch viel mehr zu berichten. Das wür- de aber den Rahmen

Flexko Multisync

Prädikat: Augenschonend

Wer bisher ein gutes, flimmerfreies Bild am AMIGA erreichen wollte, hatte nur eine teure Möglichkeit: einen Color-Multiscan.

Eine wesentlich billigere Alternative, die zudem noch etwas besser in der Bildqualität liegt, eröffnet sich mit einem monochromen Multifrequenzmonitor. Ein solches Gerät hatten wir im Test.



Das Dilemma Bildqualität-Monitorpreis ist wohl jedem AMIGA-Besitzer bekannt. Entweder kauft man einen billigen Farbmonitor (z.B. 1084 von Commodore) und hat mit flackerigem Bild, Farbbrändern und allgemeiner Bildunschärfe zu kämpfen, oder legt sich für ca. 1600,- DM einen Farbmultifrequenzmonitor zu, der die vorgenannten Nachteile nicht hat, dafür aber die nächsten 2 Urlaubsreisen aus finanziellem Notstand scheitern läßt.

Normale Monochrommonitore lassen sich nur über den FBAS- bzw. BAS-Ausgang des AMIGA betreiben, da Monitore mit TTL-Eingang normalerweise nicht mit den AMIGA-Bildsignalen zurechtkommen. Als preiswerte Alternative mit sehr guter Bildqualität bietet sich nun der Flexko MG-11 an, ein monochromer Mehrfrequenzmonitor.

Der MG-11 ist in der Lage, sich auf alle zur Zeit üblichen Videosignale von IBM-Grafikkarten einzustellen. Da die AMIGA-Signale innerhalb dieses Bereiches liegen, gibt es keine Schwierigkeiten beim Betrieb des Flexko am

AMIGA. Der MG-11 ist mit einer Auflösung von 1024x768 Punkten spezifiziert, was auf jeden Fall weit jenseits der Erfordernisse eines AMIGAs liegt.

Mit einem Drehfuß versehen, kann der Monitor optimal auf die Blickrichtung des Anwenders ausgerichtet werden. Bei der Positionierung auf einem A2000 ist die Höhe des Monitors dann aber zu hoch, was aber durch Abnehmen des Fußes beseitigt werden kann. Das Entfernen des Ständers ist zudem recht einfach.

Alle Regler sind von außen zugänglich. Dabei befinden sich die häufiger benötigten Regler für Helligkeit und Kontrast auf der linken Monitorseite (der Netzschalter befindet sich übrigens seitlich rechts am Monitor), alle anderen Regler findet man auf der Rückseite. Die Möglichkeiten der Einflußnahme auf das Bild sind ausreichend, lediglich die Bildweite könnte einen etwas größeren Einstellbereich haben.

Als Abweichung von den meisten Farbmultisyncs sei die manuelle Einstellung

der vertikalen Synchronisation genannt. Inwieweit dies aber ein Manko ist, sei dahingestellt, da diese Einstellung ja nur beim Umstecken des Monitors an einen anderen Rechner notwendig wird.

Als Schmankerl bietet der MG-11 an der Unterseite des Gehäuses einen Invert-Taster. Betätigt man diesen, wird die Helligkeitsinformation des Bildes invertiert, d. h. daß helle Stellen dunkel dargestellt werden und umgekehrt. Dieser Modus, den meisten wohl vom ATARI ST her bekannt, ist recht angenehm und daher einen Versuch wert, ihn einmal auszuprobieren.

Bildqualität

Die Qualität des Bildes darf getrost als sehr gut bezeichnet werden. Bildschärfe, Linearität und Kontrast lassen keine Wünsche offen. Der Bildschirm ist wirklich flach, selbst bei einem extremen Blickwinkel kommt ein Eindruck der Bauchigkeit nicht auf. Die Entspiegelung des Schirmes ist gut, den Rasierspiegel muß man also extra kaufen. Die

Darstellung Weiß auf Schwarz ist angenehm, in der Reversstellung stört je nach Einstellung der Bildgröße ein hellerer Rand an der rechten Seite. Das Nachleuchten der Phosphorschicht ist länger als bei herkömmlichen Farbmonitoren. Dadurch ist das Bild recht flimmerfrei. Die Nachleuchteffekte halten sich dennoch in Grenzen, lediglich bei sehr schnellen Bewegungen fallen Sie etwas störend auf.

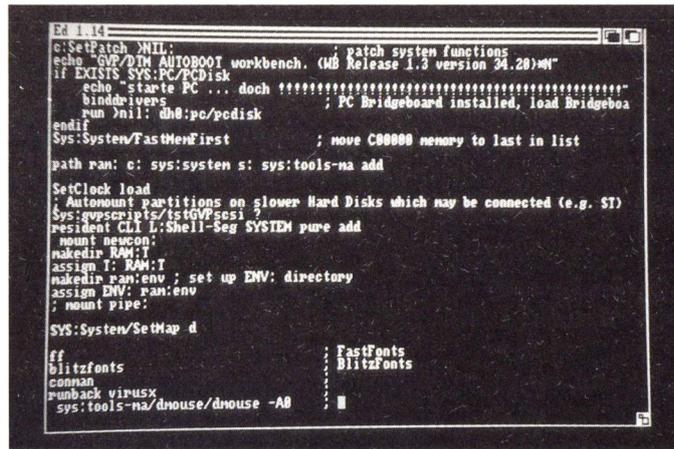
Die lange Nachleuchtdauer hat zudem einen lindernen Effekt auf das Interlace-Flackern. Bei relativ kontrastarmen Bildern und etwas zurückgedrehter Helligkeit kann man diesen Modus sogar längere Zeit ertragen. Das Flimmern ganz beseitigen kann aber nur ein extrem lange nachleuchtender Monitor, der dann aber zu stark schliert.

Farbe in Schwarzweiß

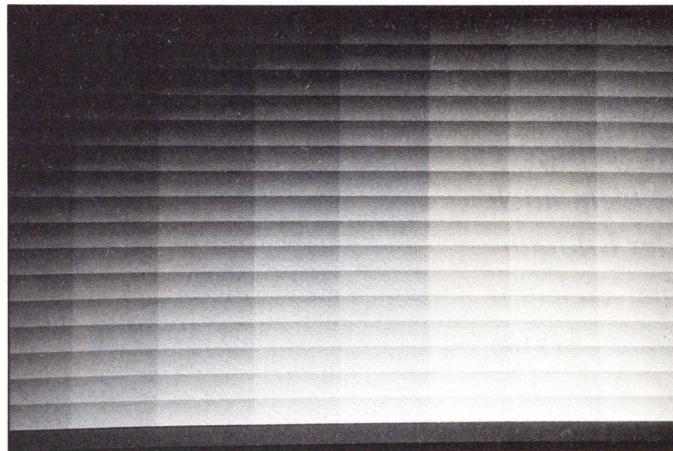
Das Handikap des MG-11 ist wohl die Umsetzung der Farben in Grautöne. Da auf dem AMIGA sehr viel mit Farben gearbeitet wird, ist es von Interesse, inwieweit Informationen verlorengehen bei dieser Umsetzung. Bei den ersten Versuchen mit einer Workbench in der Standardeinstellung waren wir überrascht. Die Farben wurden gut in Graustufen umgesetzt, ein Verstellen der Farben war nicht notwendig. Das gleiche Resultat ergab sich bei allen Anwendungen, die mit Text arbeiten. Einzig bei Programmen, die man als farbüberladen bezeichnen kann, gab es leichte Probleme, die sich zum Teil noch durch das Invertieren beseitigen ließen. Als Beispiel seien hier nur die Kopierprogramme aufgeführt, die ja durch eine besondere grafische Gestaltung auffallen.

Bei Spielen machte sich die Umsetzung Farbe-Grautöne auch etwas dämpfend auf die Bildwirkung bemerkbar. Manche Sprites sind nicht mehr so schnell zu erkennen.

Dadurch ergibt sich ein eindeutiges Anwendungsfeld für den monochromen Multifrequenzmonitor. Bei allen "ernsthaften" Arbeiten, die langes, konzentriertes Sitzen vor dem Monitor verlangen, ist der Monitor hervorragend geeignet, bei Spielen oder grafischen Anwendungen (Malprogramme) benötigt man die hohe Bildqualität nicht und ist mit einer "Farbmöhre" auch gut bedient.



Der Monitor besticht durch ein scharfes und augenschonendes Bild.



Die Umsetzung von 4096 Farben auf dem MG-11

Anschluß gesucht

Beim MG-11 wird das allgemein übliche Monitorkabel für IBM-Grafikkarten mitgeliefert. Der AMIGA-Besitzer ist daher auf einen Eigenbau des Kabels angewiesen, der zu einem geringen Preis und mit etwas Lötkunst in einer halben Stunde herstellbar ist. Mit 525,- DM ist der Monitor noch erschwinglich. Betrachtet man sich das Verhältnis Bildqualität/Preis, so kann der Monitor nur empfohlen werden. Je

nach Anwendungsbereich des AMIGAs kann er als optimales Sichtgerät bezeichnet werden und in der Wahl noch vor einem Farbmultifrequenzmonitor stehen. Personen, die keine farbigen Grafiken benötigen und hauptsächlich mit Texten umgehen (Programmierer und anderes Getier), sei ein monochromer Multisync wärmstens empfohlen.

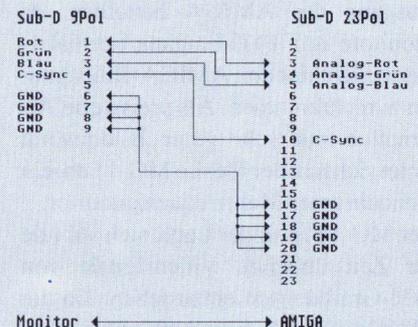
Technische Daten:

Bildröhre: 14 Zoll, papierweiße Darstellung, entspiegelt
 Synchronisationsfrequenzen:
 horizontal: 15-38 KHz
 vertikal: 45-75 Hz

Bandbreite: 30 MHz
 Auflösung: 1024x768 Punkte
 Verwendbare Videosignale:
 TTL: BAS/FBAS, RBG, RGBI, RrGgBb
 Analog: RGB-Analog

Flexko MG-11
 525,- DM
 KOGA GMBH, Frankfurt

Anschlußbelegung:



T & T

Tips & Tricks

DOS-Aufrufe bei WB-Start in M2-Hase (AM-Modula)

Bekanntlich können von Workbench gestarteten Programmen normalerweise keine DOS-Aufrufe mittels *Dos.Execute* ausgeführt werden, da dafür eine DOS-Umgebung - sprich ein offenes CLI - erforderlich ist.

Dieses Problem ist einfach zu umgehen, indem mittels *Dos.Open* auf das nicht existierende Device 'NIL:' eine CLI-Umgebung simuliert wird. So können aus Programmen beliebige Aufrufe - *NewCLI*, *SetMap* etc. - auch dann erfolgen, wenn das Programm von der Workbench aus gestartet wurde.

Bei der Gelegenheit bin ich auf ein pikantes Detail gestoßen, das vor allem die Schweizer Kollegen interessieren dürfte: 'SetMap ch1' und 'SetMap ch2' lassen sich so nämlich nicht ausführen (funktionieren auch im CLI nicht korrekt, schon bemerkt?). Nun, dieses Hakens Ursache dürfte wohl für immer verborgen bleiben. Immerhin habe ich die Lösung parat: Man "rename" die beiden KeyMaps in 'chf' und 'chd', rufe *SetMap* mit den neuen Namen, und siehe da...

Der folgende Programmausschnitt ist mit AM-Modula entworfen, ist aber sinngemäß auch auf alle anderen Sprachen anwendbar. Er führt als erstes 'SetMap f' aus, eine Belegung, die bei Betätigung der Zahlentasten sofort er-

kannt werden kann. Jedes nachfolgend geöffnete Console-Device (so z.B. auch ein CLI) benutzt die neue Belegung. Danach wird tatsächlich vom laufenden Programm aus ein CLI geöffnet. Erst

nach Eingabe von 'Endcli' wird ins Programm zurückgekehrt, welches damit allerdings endet.

Der DOS-Aufruf ist als separate Prozedur ausgeführt, die in jedes beliebige Programm eingebaut werden kann. Er erwartet die Adresse eines mit *OC* abgeschlossenen DOS-Befehls-Strings, wie er im CLI eingegeben würde. Die *BOOLEAN*-Funktion gibt den Erfolg an (= *TRUE*). Die Systemdisk, auf der sich natürlich die angeforderten DOS-Befehle befinden müssen, wird gegebenenfalls automatisch verlangt.

Roland Kappeler, Monte-Carlo

```
(* ----- *)
MODULE DosExeDemo;

(* Author   : R.Kappeler
   Last Edit: 20.04.89
   System   : AM-Amiga
   *)

FROM SYSTEM IMPORT  ADR, ADDRESS;
FROM Dos          IMPORT  FileHandlePtr, Open, Close, oldFile, Execute;

(* -----*)

PROCEDURE
  DosExecute (Order: ADDRESS): BOOLEAN;

VAR Handle:
  FileHandlePtr;
      Error : LONGINT;

BEGIN
  Handle:= Open (ADR ('NIL:'), oldFile);
  IF Handle # NIL THEN
    Error:= Execute (Order, Handle, Handle);
    Close (Handle);
    RETURN Error # 0;          (* ungleich 0: ok! *)
  END;
  RETURN FALSE;
END DosExecute;

(* -----*)

CONST
  SetMapOrder = 'SetMap f';      (* auszuführende DOS-Befehle *)
  NewCliOrder = 'NewCli';

BEGIN
  IF NOT DosExecute (ADR (SetMapOrder))
  THEN
    (* wieso hat denn das nicht geklappt...? *)
  END;
  IF NOT DosExecute (ADR (NewCliOrder)) THEN
    (* gibt's nicht: schon wieder nicht...?! *)
  END;
END DosExeDemo.

(* -----*)
```

Keyboards ermitteln

Wem beim Programmieren die Funktion aus der DOS.Library "READ" nicht genügend Komfort bietet, der wird bald anfangen, sich seine eigene Eingaberoutine zu schreiben. Da man von der Tastatur nicht den ASCII-Wert der zuletzt betätigten Taste zurückerhält, muß man den Wert erst umrechnen. Mit untenstehender Tabelle kann man sofort erkennen, welcher Code welcher Taste entspricht. In der Spalte ROR finden sich die Werte, die man nach dem Umrechnen (siehe Programm) erhält.

Will man nun eine Taste abfragen, die in der Tabelle nicht aufgeführt ist, braucht man nur untenstehendes Programm zu

starten. Es wartet auf einen Tastendruck und gibt in D0 den Originalwert und in D1 den umgerechneten Wert zurück.

(Nikolas Heide)

```

; (*Assembler: KICK-ASS (MAXON)*)
clr.l d1
clr.l d0
move.b #0,$bfec01
wait:
move.b $bfec01,d0
beq.s wait
move.b d0,d1
ror.b #1,d0
not.b d0
rts
    
```

Key	Code	ROR	Key	Code	ROR	Key	Code	ROR
A	BF	20	B	95	35	C	99	33
D	BB	22	E	DB	12	F	B9	23
G	B7	24	H	B5	25	I	D1	17
J	B3	26	K	B1	27	L	AF	28
M	91	37	N	93	36	O	CF	18
P	CD	19	Q	DF	10	R	D9	13
S	BD	21	T	D7	14	U	D3	16
V	97	34	W	DD	11	X	9B	32
Y	9D	31	Z	D5	15	**	**	**
CUP	67	4C	CDOWN	65	4D	CLEFT	61	4F
CRIGH	63	4E	HELP	41	5F	ESC	75	45
F1	5F	50	F2	5D	51	F3	5B	52
CNTRL	39	63	**	**	**	**	**	**

Deluxe Photolab

Stellen Sie sich vor, Sie besitzen das Programm Deluxe PHOTOLAB und haben es trotz 1MByte Speicher nie geschafft, einen PAL-Grafikbildschirm mit 640 mal 512 Punkten bei maximaler Tiefe (4) zu öffnen. Schlecht, oder? Mit ein paar Tricks können Sie aber dennoch in den Genuß der speicherverschlungenen Hochauflösung kommen. Zunächst mal muß eine Sicherheitskopie der PHOTOLAB-Disk 1 angefertigt werden (Siehe AMIGA-Benutzerhandbuch). Die nun entstandene Diskette mit Namen "copy of PHOTOLAB" in "PHOTOLAB" "renamen". Danach den CLI starten und folgende Befehle eingeben:

```

Delete PHOTOLAB:Trashcan.info
Delete PHOTOLAB:Trashcan
Copy SYS:System/SlowMemLast to PHOTOLAB:C/
Ed PHOTOLAB:S/Startup-Sequence
    
```

Nun erscheint auf dem Bildschirm die folgende Datei:

```

List > nil: RAM:
BindDrivers
LoadWB
setmap d
endcli > nil:
    
```

Fügen Sie als erste Zeile (ohne eine der übrigen zu löschen)

```
SlowMemLast
```

ein, und verlassen Sie den Editor durch Drücken von "Escape", danach "x" und "Return". "SlowMemLast" ist identisch mit dem Programm "FastMemFirst". Falls sich also das erstere nicht auf der Systemdiskette befindet, geben Sie überall für "SlowMemLast" "FastMemFirst" ein. Die Systemdiskette ist jetzt fertig. Nun geht's ans Austesten.

Booten Sie den Computer. Anstelle der Workbench die neue PHOTOLAB-Disk in DF0: legen. Das Programm "Paint" (oder "Colors") starten und eine Maloberfläche mit folgenden Parametern öffnen:

```
niedrige Auflösung, Non-Interlace,
Tiefe 1, Größe 3/4
```

Den Workbenchscreen schließen (geschieht durch das Roll-Up-Menü Par -

Einstellungen - Workbench), danach im Projektmenü "Schließen" anwählen. Nun können Sie eine neue Maloberfläche mit den Parametern

```
hohe Auflösung, Interlace,
Tiefe 4, Größe Voll
```

öffnen...und...o Wunder - die bestmögliche Auflösung mit der maximalen Farbenzahl in voller PAL-Größe wartet auf Ihren Befehl.

Alle Roll-Up-Menüs sind sichtbar, sogar Pinsel können definiert werden. Auch die RAM-Disk ist benutzbar. DENNOCH: Nie vergessen, der Speicher ist knapp, und falls Ihnen mal die Roll-Up-Menüs abhanden gekommen sind - eventuell hilft es, einen 1 Pixel großen Pinsel zu definieren (Scherre in der Kommandozeile).

Einzige Einschränkung: Ein Zweitlaufwerk muß (bei ABGESCHALTETER FREUNDIN) entfernt werden, da sonst der Speicher zu knapp werden würde!!

(Heiko Schmitt)

TEXT, RUCK-ZUCK

Wer hat nicht schon einmal Probleme gehabt, einen Text ohne großen Aufwand in einer Bitplane darzustellen? Abhilfe schafft hier diese Routine. Sie benutzt den internen TOPAZ.FONT, der ab der Adresse \$FC90F4 beginnt. Diese Routine ist viel schneller als die normale Textroutine der Graphics. library. Durch Abwandlung der Routine kann man sehr schöne Effekte erreichen, z.B. kursive Darstellung, Unterstreichen, mehrfarbige und wellenförmige Schrift (Intros).

(Stefan Müller)

Tips & Tricks

Diese Rubrik dient dazu, wichtige Informationen an den Leser zu bringen. Daß dafür die Mitarbeit aus der Leserschaft erforderlich ist, versteht sich von selbst.

Falls Sie (Ja, genau Sie, nicht die Person hinter Ihnen!) noch etwas in der Trickkiste haben, so holen Sie es schnellstens heraus und senden es uns zu. Einsendungen werden natürlich honoriert.

MAXON Computer
Redaktion KICKSTART
Tips&Tricks
6236 Eschborn

```

custom      = $DFF000
bplcon0     = $100
bplcon1     = $102
bpl1mod     = $108
bpl2mod     = $10A
dffstrt     = $092
diwstrt     = $08E
color00     = $180
color01     = $182
dmacon      = $096
intena      = $09A
intreq      = $09C
intreqr     = $01E
bpl1pth     = $0E0
bpl2pth     = $0E4
execbase    = 4
allocmem    = -198
freemem     = -210

movem.l d0-d7/a0-a6, -(a7)
move.l execbase.s, a6          ;Bit-Plane anfordern
move.l #1*40*200, d0          ;(Speicher)
move.l #10002, d1
jsr allocmem(a6)
move.l d0, Bpl1

move.l Bpl1, a1                ;Bit-Plane in a1
move.l #Test, a0              ;Textadresse in a0
jsr text                       ;und drucken

lea custom, a6
move.w #$00A0, dmacon(a6)     ;Playfield
move.w #$1200, bplcon0(a6)    ;initialisieren
move.w #$0000, bplcon1(a6)
move.l #$003800D0, dffstrt(a6)
move.l #$2C81F4C1, diwstrt(a6)
move.w #$0000, bpl1mod(a6)
move.w #$0000, color00(a6)
move.w #$0FFF, color01(a6)
move.w #$0020, intena(a6)
move.w #$8350, dmacon(a6)
wait:    move.w intreqr(a6), d4    ;Vertical-Blank
         btst #$5, d4
         beq wait                ;Nein, dann zu wait
         move.w #$0020, intreq(a6) ;Ja, stoppen
         move.l Bpl1, bpl1pth(a6) ;Bitplane-Adresse eintragen
         btst #6, $BFE001         ;Maustaste gedrückt
         beq end                 ;Ja, dann Ende
         bra wait
end:    move.w #$81F0, dmacon(a6)
         move.w #$8020, intena(a6)
         move.l execbase.s, a6
         move.l Bpl1, a1
         move.l #1*40*200, d0
         jsr freemem(a6)
         movem.l (a7)+, d0-d7/a0-a6
         rts

Bpl1:   dc.l 0
Test:   dc.b 'Hallo Assembler-Freaks!!!', 0
         even

;Text-Funktion mit dem Topaz.font im ROM
;a0: Adresse des auszudruckenden Textes
;a1: Adresse der Bitplane
text:
movem.l d1-d2/a0-a3, -(a7); Register retten
text1: move.b (a0)+, d1      ;Buchstabe holen
         beq textende       ;0-Byte, dann Ende
         sub.b #$20, d1      ;richtige Adresse
         move.l #$FC90F4, a2 ;für den Buchstaben
         add.l d1, a2        ;berechnen
         moveq #7, d2        ;ein Buchstabe = 8 Pixel hoch
         move.l a1, a3       ;Bitplane kopieren
text2:  move.b (a2), (a3)    ;Teil des Buchstaben kopieren
         add.l #40, a3       ;nächste Zeile
         add.l #$C0, a2      ;nächsten Teil
         dbf d2, text2      ;8 mal
         addq.l #1, a1       ;8 Pixel weiter
         bra text1         ;Schleife machen
textende: movem.l (a7)+, d1-d2/a0-a3; Register holen
         rts

```

DIE GEHEIMEN SCHÄTZE DES AMIGABASIC

(mit CTRL auf verschlungenen Pfaden)

Im folgenden soll gezeigt werden, wie ASCII-Darstellung, Hexadezimal- und Oktalumwandlungen im List-Fenster von AmigaBasic vorgenommen werden können.

Das wohl Bedeutendste, was ich auf der Suche nach geheimen Geheimnissen des AmigaBasic, die der Programmierer vergraben hatte, entdeckte, war wohl die Geschichte mit dem ASCII-Code.

Es ist sicher bekannt, daß man beim Programmieren öfter mal von diesem oder jenem Zeichen den ASCII-Code braucht. Dann geht es los: "Also der Kleinbuchstabe 'a' hat .. warte .. äh ..genau.. 97! Dann hat 'b' 98!".

Alles klar, aber was ist mit den Sonderzeichen, den Funktionstasten? Will man alle auswendig lernen oder die ASCII-Tabelle an den Bildschirm nageln? Natürlich nicht! Dies hat sich nämlich auch der oben angesprochene Programmierer gedacht, weshalb er sich in einer seiner kaffeekonsumierenden Nächte ein kleines Utility bastelte. Und da er den Müll (für den wir ihm dankbar sind), den er während seiner Kreativphasen am BASIC produzierte, im Programm ließ, können wir nun davon profitieren. Will man den ASCII-Code eines Zeichens auf den Bildschirm bringen, so gibt man im List-Window des AmigaBasic folgendes ein:

```
[ ALT ]+[ ä ] [ CTRL ]+[ / ] [ Z ]
```

Mit ALT & "ä" schreibt man ein Hochkomma, mit CTRL & "/" wird später das schlummernde Utility geweckt, und "Z" steht für das Zeichen, dessen ASCII-Code man sucht. Nun braucht man nur noch die Zeile zu verlassen, und wie ein Wunder steht der ASCII-Code des Zeichens an dessen Stelle.

Das Hochkomma ist nur am Anfang der Zeile notwendig und man kann nach ihm die Kombination CTRL & "/", Zeichen sooft wiederholen, bis der Zeichenpuffer voll ist (es ist allerdings ratsam, normale Kommas zwischen die Zeichen zu setzen, da es sonst zu einem Zahlenwirrwarr kommen würde, an dem wohl nur Graf Zahl seine Freude hätte).

Doch dies sollte nicht der letzte Schatz sein, der aus den Tiefen des Basics heraufgeholt wurde. So gibt es da z.B. ein Utility, das den ASCII-Code in hexadezimaler und octaler Schreibweise darstellt (und dabei muß wieder nur das entsprechende Zeichen angegeben werden!). Dies wird unten noch beschrieben.

Daß darüberhinaus noch viele kleine, unentdeckte Schätze im AmigaBasic verborgen liegen, macht eine Reise in die Tiefen der Tastenkombinationen umso spannender.

Ich für meinen Teil werde mir gleich wieder bis in die Nacht daran zu schaffen machen, die Geheimnisse des AMIGAS zu erkunden.

Charakter-Code darstellen:

```
[ALT]+[ä] [CTRL]+[/] [<Zeichen>]
```

Verläßt man nun die Zeile, so wird das Zeichen "Zeichen" in den ASCII-Code des Zeichens verwandelt.

Beispiel:

```
Alt&ä , Ctrl&/ , a = 97
ASCII-Code(a) = 97
```

Charakter-Code in Hex:

```
[ALT]+[ä] [CTRL]+[l] [<Zeichen>]
```

Wandelt den ASCII-Code des Zeichens "Zeichen" nach Verlassen der Zeile in die hexadezimale Schreibweise um.

Beispiel:

```
Alt&ä , Ctrl&l , a = &H6100
ASCII-Code(a) = 97. HEX(97) = 61
```

Charakter-Code in Oct:

```
[ALT]+[ä] [CTRL]+[k] [CTRL]+[l] [<Zeichen>]
```

Wandelt den ASCII-Code des "Zeichen" in die hexadezimale Schreibweise um, und dann in die oktale.

```
Alt&ä , Ctrl&k , Ctrl&l , a = &O6141
ASCII-Code(a) = 97. OCT(97) = 141
```

Möge die Macht mit Ihnen sein,
Oliver C. Neutert

ZUWEISUNGEN MIT ASSIGN OHNE CLI

Hier ein Tip, wie Zuweisungen mit ASSIGN ohne CLI möglich sind! Voraussetzung ist allerdings eine Workbench 1.3!

Viele User haben inzwischen eine ansehnliche Sammlung verschiedenster Fonts, doch leider alle wunderbar verstreut. Oder alle auf einer Diskette in einem übervollen Ordner "fonts", das gibt aber bei manchen Programmen - wie z.B. bei Notepad oder DPaint - Schwierigkeiten, da bei diesen Programmen im Menü nur eine begrenzte Anzahl Fonts aufgenommen werden können.

Abhilfe läßt sich, wie gleich gezeigt wird, leicht schaffen:

Zuerst legt man sich auf seiner Festplatte oder WB-Diskette mittels "makedir zusatzfonts", "makedir zusatzfonts/fonts1", "makedir zusatzfonts/fonts2" und "makedir zusatzfonts/fonts3" drei Ordner an, in die man weitere, vielleicht sogar nach Ordnungsprinzipien sortierte Fonts kopieren kann. Dabei ist zu beachten, daß zu einem speziellen Font ein Directory mit dem Namen des Fonts und ein File, das auf Font endet, gehören.

Schön und gut - nun ist zwar Ordnung, doch wie kann ich von Programmen aus diese Fonts nutzen, die sich nicht im Ordner "fonts" befinden, in dem aber fast alle Programme ausschließlich nachgucken?

Schon mal was von assign gehört? O weh! CLiii, stöhnt der Durchschnittsanwender bzw. Anfänger. Doch wozu gibt es eigentlich Icons?

Folgendes Batch-file läßt sich bei Verwendung der Workbench 1.3 von der Workbench aus starten. Dazu kopiert man sich ein Projekt-Icon (z.B. Textda-

tei-Icon vom Notepad) und benennt es mit RENAME in Fontauswahl um. Mittels INFO aus dem Workbench-Menü trägt man in die Info-Datei als DEFAULT TOOL

```
sys:c/iconx
```

ein und in TOOL TYPES

```
WINDOW=newcon:0/10/390/160/  
AssignFonts
```

und speichert diese dann ab.

Das Batchfile wird danach mit einem Editor (z.B. ed), der kein Icon erstellt, unter dem Namen Fontauswahl gespeichert.

Nach dem Doppelklick auf das Icon Fontauswahl öffnet sich ein Fenster, und die Zusatzfonts-Ordner werden aufgelistet. Geben Sie nun die Nummer des gewünschten Fonts-Ordners ein. Wenn Sie dann anschließend ein Anwenderprogramm laden, werden die Zeichensätze des von Ihnen ausgewählten Fonts-Ordners geladen!

(Georg Schuh)

```
.key " "  
.bra $  
.ket #  
assign from $$$#: " "  
echo "*Ec*E[4mAssign Zusatzfonts *N*N*E[0m verfügbare Zusatzfonts:*N"  
cd sys:Zusatzfonts  
; bei Verwendung einer Extradiskette statt sys: den Diskettenamen  
dir  
cd sys:  
echo "*N Bitte Zusatzfonts 1-3 auswählen"  
echo " Nummer eingeben!"  
echo " (0 für normale Fonts) (RETURN für Abbruch) " noline  
setenv >NIL: q ?  
IF $q eq " "  
  skip err  
ENDIF  
if $q eq "1"  
  assign fonts: zusatzfonts/fonts1  
  skip fo  
endif  
if $q eq "2"  
  assign fonts: zusatzfonts/fonts2  
  skip fo  
endif  
IF $q eq "3"  
  assign fonts: zusatzfonts/fonts3  
  skip fo  
ENDIF  
IF $q eq "0"  
  assign fonts: fonts:  
  echo "*N*NWieder normale Fonts!"  
  skip err  
ENDIF  
lab fo  
echo "*N*NAusgewählt wurden Fonts" noline  
getenv q  
endskip  
lab err  
endskip  
assign from$$$#:
```

EIN SCHRITT GESPART

Bei vielen Grafikanwendungen kann man mit einem kleinen Trick Zeit und vor allem Speicherplatz sparen.

Bei kleineren Assembler-Grafikprogrammen mit mehr als vier Farben sieht der Anfang meist so aus: Screen öffnen,

Window öffnen, aus dem Windowhandle die Adresse des Rastports holen, um die Grafikroutinen, die meist den Rastport benötigen, aufrufen zu können. Es geht aber auch einfacher: Das Window kann man sich nämlich auch sparen, da auch jeder Screen einen eigenen Rastport hat. In der Struktur, dessen Adresse man von der OpenScreen-Funktion erhält, steht nun aber nicht der Zeiger auf

den Rastport, sondern der Rastport selbst. Man muß also zu der Adresse, die man erhält, genau 84 (oder \$54) addieren, um den Zeiger auf den Rastport zu erhalten. Mit diesem Zeiger können nun alle Grafikoperationen, die mit dem Rastport arbeiten, direkt auf einem Screen ausgeführt werden.

(Claus Brunzema)

Jetzt für AMIGA

PICTURE-DISKS GRAFIKSAMMLUNG

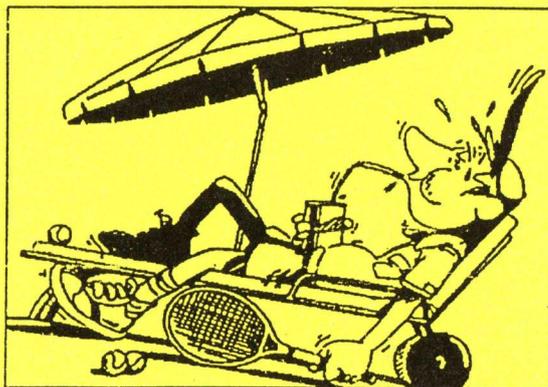
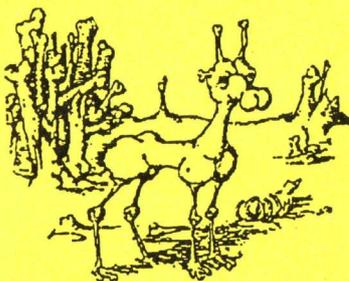
Die Picture-Disks-Serie ist die erste Graphik-Sammlung mit hochauflösenden Graphiken für Commodore Amiga Computer. Die Picture-Disks sind von der Qualität einzigartig und decken alle Themenbereiche ab.

1000 Graphiken DM 98,-

2000 Graphiken DM 189,-

Die hochauflösenden Graphiken im IFF-Format können ohne Probleme in allen gängigen Graphik-, Mal- oder DTP-Programme der AMIGA Welt übernommen und dort weiterverarbeitet werden. Ein schnelles Auffinden der einzelnen Graphiken garantiert ein mitgeliefertes Handbuch. Picture-Disks sind ein Muß für alle professionellen DTP-, Mal- und Textverarbeitungssysteme mit Graphikeinbindung.

Unverbindlich empfohlene Verkaufspreise



PICTURE DISKS erhalten Sie im guten Fachhandel oder direkt beim Heim-Verlag

Heim Verlag

Heidelberger Landstr. 194
6100 Darmstadt-Eberstadt
Telefon 0 61 51-56057

Schweiz: DataTrade AG
Langstr. 94
CH-8021 Zürich

BESTELL-COUPON

an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Ich bestelle 1000 AMIGA PICTURE-DISKS zum Preis von DM 98,-
 2000 AMIGA PICTURE-DISKS zum Preis von DM 189,-
zzgl. DM 5,- Versandkosten (unabhängig von bestellter Stückzahl)
 per Nachnahme Verrechnungsscheck liegt bei

Name, Vorname _____

Straße, Hausnr. _____

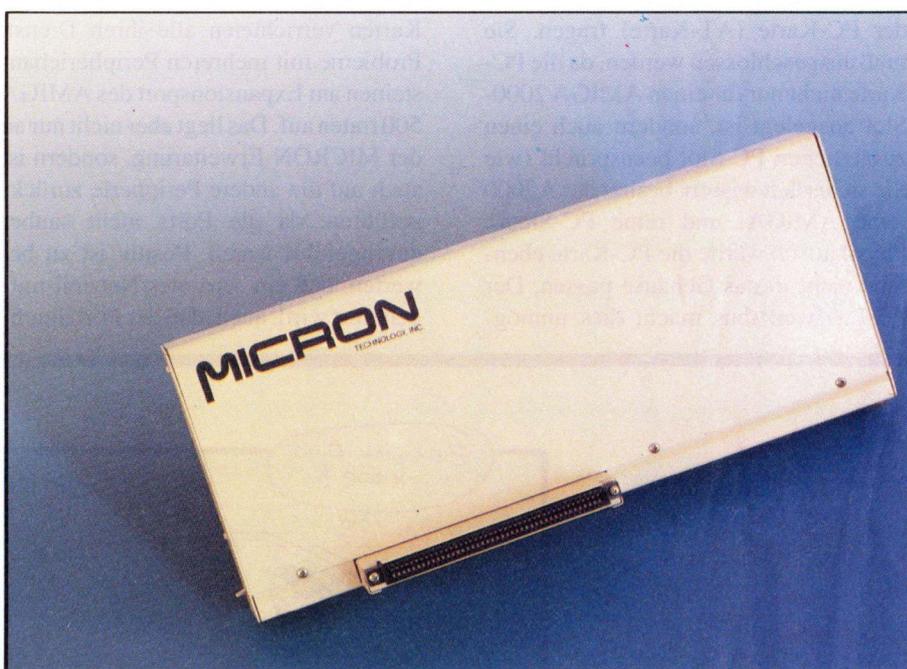
PLZ, Ort _____

Benutzen Sie auch die in KICKSTART vorhandene Bestellkarte.

AMIGA 2000

Steckplatz am AMIGA 500

Steckkarten für den AMIGA 2000 gibt es eine ganze Menge, ob es sich um eine RAM-, EPROM-Karte oder Filecard handelt. Die mitunter gegenüber den für den AMIGA 500 vergleichbar billigeren Erweiterungen blieben bisher AMIGA 2000-Besitzern vorbehalten.



Das AMIGA 500 Interconnect Board erlaubt, AMIGA 2000-Steckkarten am AMIGA 500 zu betreiben.

Die amerikanische FIRMA MICRON hat diesbezüglich Abhilfe geschaffen und eine Erweiterung für den AMIGA 500 herausgebracht, die dem A500 einen AMIGA 2000-Steckplatz zur Verfügung stellt, genaue Bezeichnung: MICRON AMIGA 500 Interconnect Board. Ob alle AMIGA 2000-Erweiterungskarten betrieben werden können, muß jedoch näher untersucht werden.

Im Lieferumfang befinden sich neben der Erweiterung eine kurze Dokumentation zur Installation und ein separates Netzteil. Die Erweiterung wird an den Expansionport des AMIGA 500 ge-

steckt. Da sie hochkant ist, sieht sie dort ein wenig deplaziert aus, überragt sie doch den Rechner um einiges. Das separate Netzteil wird über einen DIN-Stecker auf der Rückseite eingesteckt. Eine Leuchtdiode auf der Frontseite zeigt den Betriebszustand an. Ein Schalter, ebenfalls auf der Frontseite, dient zum Ein-/Ausschalten der Erweiterung. Das Gehäuse ist aus Metall und amigafarben gehalten.

Um eine AMIGA 2000-Karte einzustecken, muß das Gehäuse geöffnet werden, sechs Schrauben sind zu diesem Zweck zu entfernen. Zwei Führungsschienen geben der Karte im Gehäuse Halt.

Die Elektronik, die im Gehäuse steckt, ist nicht sehr aufwendig, neben einem TTL-Chip finden sich noch einige Array-Widerstände.

Der Expansionport des A500 ist durchgeführt, jedoch nicht gepuffert. Deshalb können Probleme mit anderweitiger Expansionport-Peripherie auftreten.

Kompatibilität

Kernpunkt der MICRON-Erweiterung ist mit Sicherheit die Kompatibilität. Zum Test haben wir einige AMIGA 2000-Karten herangezogen: die Commodore 2 MB-Erweiterungskarte, die Collcard-Eprom-Bank und die GVP-

Festplatte. Die Speichererweiterung macht keinerlei Probleme, ebenso verhielt es sich mit der Collicard, von der auch einwandfrei gebootet werden konnte. Die GVP-Festplatte verrichtete ihren Dienst ebenfalls ohne Probleme. Allerdings ist anzuraten, normale AMIGA 500-Peripherie, die am Expansionport betrieben wird, auf Verträglichkeit zu testen. Im Test haben wir eine weitere Festplatte und eine 2 MB-Speichererweiterung zusätzlich angeschlossen. Der AMIGA 500 erkannte beides an, jedoch alle drei Komponenten (MICRON-Erweiterung, A500-Festplatte, 2er MB-Speichererweiterung) vertrug der A500 nicht. Einige Leser werden sicher nach der PC-Karte (AT-Karte) fragen. Sie muß ausgeschlossen werden, da die PC-Karte nicht nur für einen AMIGA 2000-Slot ausgelegt ist, sondern auch einen zusätzlichen PC-Slot beansprucht (wie Sie sicherlich wissen, besitzt der A2000 reine AMIGA- und reine PC-Slots). Physikalisch würde die PC-Karte ebenfalls nicht in das Gehäuse passen. Der PC-Laufwerksbus macht dies unmög-

lich. Die Schnittstelle steht nämlich über und da die MICRON-Erweiterung vollkommen geschlossen ist, müßte die Sub-Din-Buchse entfernt werden. Da die PC-Karte nicht eingesetzt werden kann, gilt das natürlich auch für die AT-Karte.

Fazit

Die MICRON-Erweiterung schlägt mit 298.-DM zu Buche, ein Preis, der natürlich zu denken gibt. Ob sich der Kauf lohnt, muß sich jeder selber überlegen und seine Bedürfnisse genau analysieren. Unter Umständen lohnt sich der Kauf aber durchaus. Die getesteten Karten verrichteten alle ihren Dienst. Probleme mit mehreren Peripheriebausteinen am Expansionsport des AMIGA 500 traten auf. Das liegt aber nicht nur an der MICRON-Erweiterung, sondern ist auch auf die andere Peripherie zurückzuführen, da die Ports nicht sauber durchgeführt waren. Positiv ist zu bewerten, daß ein separates Netzteil mitgeliefert wird, auch, daß der Port durch-

geführt ist. Der Schalter zum Ein-/Ausschalten kann ebenfalls sehr nützlich sein. Negativ fiel lediglich auf, daß das Gehäuse vollkommen geschlossen ist. Verbindungen nach außen können somit nicht gezogen werden.

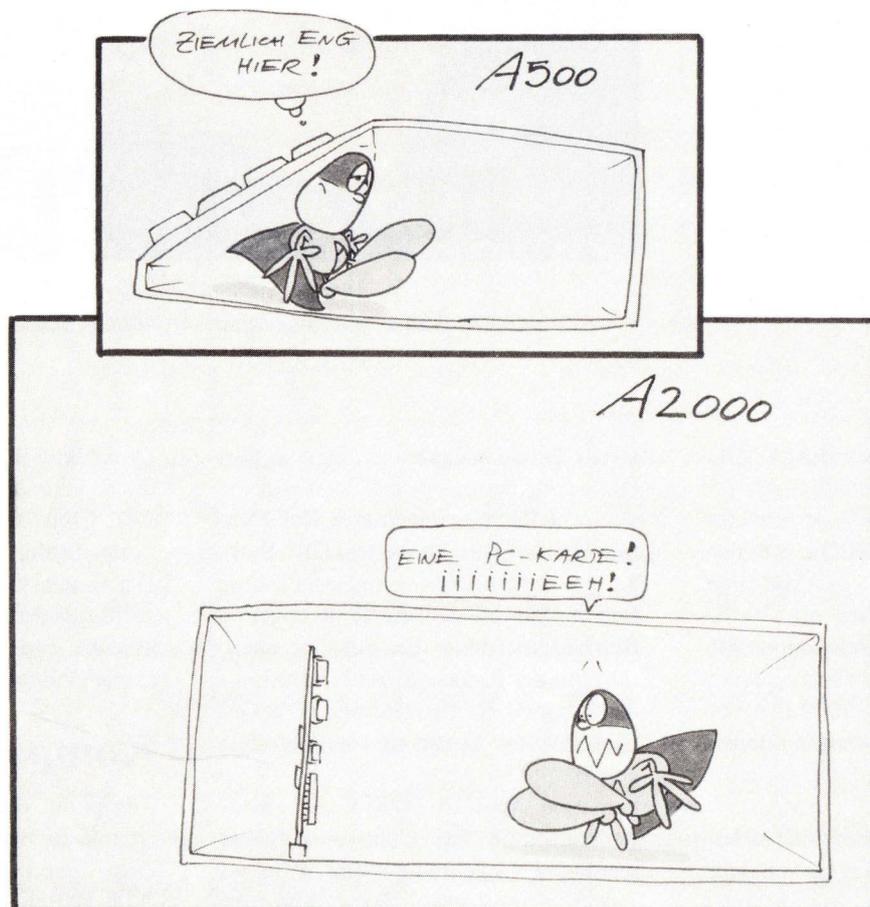
MICRON AMIGA 500 Interconnect Board

- + separates Netzteil
- + Ein-/Ausschalter
- + Expansionport durchgeführt
- Gehäuse vollkommen geschlossen

Anbieter:

CSS
Auf der Warte 46
6367 Karben 1
Tel. (06039) 5776

Preis: 298.- DM



TAP 13/89 © MAXON

Einkaufsführer

Hier finden Sie Ihren

Commodore/Amiga Fachhändler

1000 Berlin

HD AMIGA Hardware Spezialist

Computertechnik

Sämtliche verfügbaren KICKSTART-PD's vorrätig!
Und über 1500 weitere PD's für den AMIGA!

1000 Berlin 65 · Pankstr. 42
Tel. 030/465 70 28-29

COMPUTER-STUDIO

Schlichting

... die etwas andere Computerei

ATARI-Fachmarkt
MS-DOS Fachmarkt · NEC-Fachhandel

Katzbachstraße 8 · 1000 Berlin 61
☎ 030/7864340

SOFTPOWER

Das Software-Paradies im Norden Berlins!

1000 Berlin 65
Schwedenstr. 18c
Tel. 030/492 20 56

Mo-Fr. 11-18.30
Sa. 10-14
Daily News !!!

2000 Hamburg

Computer

Hardware · Software · Zubehör

Lilienstraße 32
(beim Mönckebergbrunnen)
2000 Hamburg 1
Tel. (040) 33 67 08

SYSTEMSHOP

GMA

Systemhändler
Wandsbeker Chaussee 58
2000 Hamburg 76

2000 Hamburg

**BRINKMANN'S
Computer Ranch**

SPEZIALISTEN für

AMIGA und PC - SYSTEME

Mühlendamm 2
2000 HAMBURG 76
Commodore Tel.: 040/252557

2160 Stade

BERGHAU

Büromaschinen · EDV-Systeme
Neue Straße 5 · 2160 Stade
Telefon (04141) 2364 u. 2384

2300 Kiel

Hardware
Software
Service

**Home
Computer
Laden**

Spezialisiert auf
Public Domain
Immer die neueste Software auf Lager

Gutenbergstraße 5 · 2300 Kiel · Tel. (0431) 55 55 55

2900 Oldenburg

**GOLDT
Computerhaus**

Donnerschweer Straße 127-129
(gegenüber Weser Ems Halle)
2900 Oldenburg
Telefon (0441) 88 47 06

2940 Wilhelmshaven

Radio Tiemann

Commodore-Systemfachhändler

Marktstraße 52
2940 Wilhelmshaven
Telefon (04421) 26145

3000 Hannover

COM DATA

Am Schiffgraben 19 · 3000 Hannover 1
Telefon (0511) 326736

Für Action

CSJ

Für Action

Hannover's Softwarethek Nr. 1

DIE AUSWAHL an Software für
C 16, C 64 / 128, Amiga,
Atari ST u. IBM PC.

An der Tiefenriede 27 · 3000 Hannover 1
Tel. 05 11/88 63 83

3500 Kassel

Hermann Fischer GmbH

Commodore-Systemfachhändler

Rudolf-Schwander-Str. 5-13
3500 Kassel
Telefon (0561) 70 00 00

4500 Osnabrück

**B.O.S.S.
Computerhaus**

Am Berliner Platz
Goethering 3
4500 Osnabrück
Telefon (0541) 26570

4650 Gelsenkirchen-Horst

MENTIS GmbH

Hard- und Software, Literatur
Bauteile, Service, Versand
Groß- und Einzelhandel

Poststraße 15 · 4650 Gelsenkirchen-Horst
Telefon (02 09) 5 25 72

6200 Wiesbaden

Poststraße 25
Luisenstraße 47
6200 Wiesbaden
Telefon (061 21) 56 20 91



Werbung & EDV GmbH

AUTORISIERTER
COMMODORE
SYSTEM-HÄNDLER



Commodore

COMPUTER TREFF

Computerbedarf, PD und
Software für

ATARI, AMIGA, PC

Nettelbeckstraße 12
6200 Wiesbaden
Tel. (061 21) 40 43 02

6380 Bad-Homburg

PDC GmbH

Produkte u. Details Computerverband

Luisenstraße 115
Ladenpassage Alter Bahnhof
6380 Bad-Homburg
Telefon (0 61 72) 2 47 48



Autorisierter Commodore-Systemhändler!

6457 Maintal

Landolt-Computer

Beratung · Service · Verkauf · Leasing

Autorisierter Commodore-Händler

Wingertstr. 112 · 6457 Maintal/Dörnigheim
Telefon (061 81) 4 52 93

Bei uns werben bringt

GEWINN



Sprechen Sie mit uns.
Heim Verlag 0 61 51 / 56057

6551 Fürfeld



HARD
+
SOFT

Rathausstraße 2
6551 Fürfeld
Telefon (0 67 09) 7 78

6800 Mannheim



Computersysteme + Textsysteme

6800 Mannheim 24

Casterfeldstraße 74-76

Telefon (06 21) 85 00 40 · Teletex 6 211 912

7000 Stuttgart

»If AMIGA, go to Schreiber«
Stuttgart's starker Computer-Laden.

**SCHREIBER
COMPUTER**

Rotebühlplatz 10
7000 Stuttgart-1
Tel. 0711/227099

Neu · Neu · Neu · Neu
Im SUBWAY
BREUNINGER City

7800 Freiburg



Comp.Z

Pochgasse 31
7800 Freiburg
T. 0761/554280



AUTORISIERTER
COMMODORE
SYSTEM-HÄNDLER

7890 Waldshut-Tiengen

hettler-data

service gmbh

Lenzburger Straße 4
7890 Waldshut-Tiengen
Telefon (07751) 30 94

8000 München

AMIGA

Hard- und Software

PD Fish — 199
Kickstart — 160

MODL FOTO · VIDEO · AUDIO · FERNSEH
COMPUTER · GELEGENHEITSMARKT
REPARATUREN · FOTOKOPIEN
FARBKOPPIEN · PASSBILDER
MAX-WEBER-PLATZ · U 4/5 · ☎ 4 80 16 50

8400 Regensburg

**Zimmermann
elektroland**

8400 Regensburg
Dr.-Gessler-Str. 8
☎ 09 41/ 9 50 85

8390 Passau
Kohlbruck 2a
☎ 08 51/ 5 20 07

8700 Würzburg

**SCHILL
BÜROTEAM**

Hardware · Software
Service · Schulung

computer center

am Dominikanerplatz
Ruf (09 31) 3 08 08 - 0

Schweiz

Computer Trend

Ihr Computer Spezialist

5000 Aarau, Bahnhofstrasse 86,
Tel. 064/22 78 40

4102 Basel-Binningen, Kronenplatz,
Tel. 061/47 88 64

5430 Wettingen, Zentralstrasse 93,
Tel. 056/27 16 60

8400 Winterthur, St. Gallerstrasse 41,
Tel. 052/27 96 96

8021 Zürich, Langstrasse 31,
Tel. 01/241 73 73

Grösste Auswahl an
Peripherie, Software, Literatur
und Zubehör.

ATARI

Commodore

ÖSTERREICH

A-1180 Wien

Ihr Amiga-Händler in Wien

COMPUTING

Tel. (0222) 48 52 56
A-1180 Wien - Schulgasse 63

KICKS

FÜR INSIDER

Heia Safaria

Während einige unserer Leser sich in fernen Ländern dem Urlaub hingeben, wollen wir den in der Heimat verweilenden AMIGA-Usern noch etwas Programmierstoff geben - wo doch ein Trip durch die Wildnis des Betriebssystems wesentlich erlebnisreicher ist, als die fernste Abenteuersafari. Hier gibt es mehr Unentdecktes als im tiefsten Dschungel, mehr Abstürze als in den Alpen, kürzere Wartelisten als bei Stand-By-Flügen, weniger geklaute Programme als gemopste Handtaschen und deutlich mehr Platz im RAM als am Strand. Kurzum - warum bleiben Sie nicht gleich zuhause? Wir haben jedenfalls diesmal viele KICKS zusammengestellt, die Ihnen über den Sommer helfen. Auch unser Freund scheint von seinem Urlaubsanimateur nicht sonderlich animiert und sehnt sich im Geiste schon wieder nach seinem AMIGA.

Wir wünschen Ihnen jedenfalls einen schönen Sommer und melden uns am 18.8. zurück.

INHALT

Krypt (C)	84
New Debugger (Assembler)	88
Vektorcheck (C)	92
Gib Gas (Assembler)	94
3D Echtzeit (C) (Assembler)	96

VON THOMAS BECKER

KRYPT

Verschlüsselung von Daten



„KRYPT“ kann jede Art von AMIGA-Datei sowohl ver- als auch entschlüsseln. Somit ist dieses kleine Programm ein wichtiges Utility, um (geheime) Daten vor unbefugten Zugriffen zu schützen. Dabei werden z.B. Textdateien bis zur Unkenntlichkeit verstümmelt, und Programmdateien sind (anscheinend) nicht mehr ausführbar.

Die verschlüsselten Dateien sind für einen Nichteingeweihten so gut wie nicht mehr zu restaurieren. Dafür sorgt „KRYPT“ mit einem sicheren Schlüsselsystem. „KRYPT“ kann entweder mit oder ohne Passwortschutz verwendet werden. Entfällt der Passwortschutz, so ist die Zahlenreihe in Zeile 30 der Schlüssel. Obwohl diese Zahlen auch später immer noch variiert werden können, könnte es ja vorkommen, daß eine „unautorisierte“ Person in den Besitz von „KRYPT“ kommt und somit die codierten Dateien wieder decodieren könnte. Und dies verhindert der (optionale) Passwortschutz. Aus

dem Passwort wird programmintern ein neuer Schlüssel errechnet. Jedes Passwort bedeutet somit einen neuen Schlüssel und eine andere Kodierungsart. Kennt die "unautorisierte" Person nicht das Passwort, hat sie auch fast keine Chance, etwas korrekt zu entschlüsseln. Ein sehr sicheres System, zumal Dateien auch doppelt oder dreifach verschlüsselt werden können und somit mit der Hilfe von "KRYPT" vielfältige Arten des Datenschutzes realisiert werden können.

Aufgerufen wird das Programm durch

```
KRYPT Quelldatei[.k] [Zieldatei|directory[.k]]
```

Ohne Angabe der Endung (ist in diesem Fall '.k', kann aber auch in Zeile 37 abgeändert werden) wird die entsprechende Datei verschlüsselt, im anderen Falle entschlüsselt. Wird "KRYPT" allein bzw. "KRYPT?" eingegeben, erscheint ein kurzer Hilfstext. Bei der Parameterangabe verhält sich der "KRYPT"-Befehl ebenso wie z.B. der "copy"-Befehl, hält sich also bei den Dateibezeichnungen an Amiga-DOS-Konventionen. Nur bei der Quelldatei darf kein Verzeichnisname angegeben werden, sonst sind komplette Pfade mit Geräten etc. erlaubt. Geschieht eine Falscheingabe, reagiert das Programm mit einer Fehlermeldung. "KRYPT" ist als CLI-Befehl am besten im C-Verzeichnis aufgehoben.

Einige Beispiele für Parameterangaben:

```
- Krypt df0:s/startup-sequence prt:
verschlüsselte startup-sequence-Datei auf
Drucker
```

```
- Krypt ram:c/list:
list wird verschlüsselt und als list.k im
Basisverzeichnis abgelegt
```

```
- Krypt sys:preferences nil:
preferences wird erst verschlüsselt und
dann ins Datengrab (nil:) befördert
```

```
- Krypt c:/list.k ram:
clist.k wird entschlüsselt und als list
in ram:c abgelegt
```

```
- Krypt work-disk:c/dir
```

```
- Krypt df1:datei.k *
Datei in Klartext auf dem Bildschirm
```

```
- Krypt datei ser:
Datei wird verschlüsselt über ser:
geschickt
```

```
1: /* PRG: KRYPT
2:   Autor: Thomas Becker
3:   (c) MAXON Computer GmbH 1989
4:   KICKSTART 1989
5:
6:   Aufruf: KRYPT quelledatei[.k] [ziel[.k]]
7:   !!! Dateien mit [.k] werden codiert!!!
8:   !!! Dateien ohne [.k] decodiert !!!
9:   Options: cc krypt.c -s +l
10:  ln krypt.o -lc32 -lm
11: */
12:
13: #include <exec/types.h>   /* benötigte Headerfiles */
14: #include <stdio.h>
15: #include <exec/memory.h>
16: #include <libraries/dosexten.h>
17: #include <libraries/dos.h>
18:
19: /*#define PASSWORD */
20: /* wenn definiert (ohne Kommentar) dann Passwortabfrage */
21:
22: #define TYPE eintrag->fib_DirEntryType
23: #define NAME eintrag->fib_FileName
24: #define SIZE eintrag->fib_Size
25: #define MAX_BUFF_SIZE 200*512L /* Max-Buffer-Konstante */
26: #define KEYS sizeof(key)/sizeof(UBYTE)
27: #define CODIERE 1
28: #define DECODIERE 0
29:
30: static struct FileInfoBlock *eintrag;
31: UBYTE key [] = {9,3,0,8,5,7,2,4,1,9,8,6}; /* veränderbar */
32: /* Schlüssel zum Vertauschen, gültig wenn undef PASSWORD */
33: BOOL merke=FALSE;
34:
35: VOID hilfe (), make_it ();
36: BYTE endezeichen [3] = ".k"; /* veränderbar */
37:
38: main (argc,argv)
39: int argc;
40: char *argv [];
41: {
42:   BYTE ziel [232];
43:   BOOL flag=TRUE;
44:   LONG buff_size;
45:
46:   if ((argc<2) || (argc>3) || (argv[1][0]=='?')) hilfe ();
47:   ziel[0]='\0';
48:
49:   if (argc==3)
50:   {
51:     strcpy (ziel,argv[2]);
52:     if ((check_datei(ziel))&&(TYPE>0))
53:     {
54:       if (((ziel[strlen(ziel)-1])!=':')&&
55:           ((ziel[strlen(ziel)-1]!='/') strcat (ziel, "/"));
56:     }
57:     else flag=FALSE; /*Dateiname*/
58:   }
59:
60:   if (check_datei(argv[1]))
61:   {
62:     if (!(strcmp("nil:",ziel))&&(strcmp("ser:",ziel))&&
63:         (strcmp("par:",ziel))&&(strcmp("prt:",ziel))&&
64:         (strcmp("*.k",ziel)))) merke=TRUE;
65:     else if (flag) strcat (ziel,NAME); /*wenn Verzeichnis,
66:         nicht bei ser: par: prt: etc */
67:     make_it (argv[1],ziel,SIZE); /* zum (De)Codieren */
68:   }
69:   else printf ("%s nicht gefunden!\n",argv[1]);
70:   FreeMem (eintrag,sizeof (struct FileInfoBlock));
71: } /* end of main */
72:
73: int check_datei (name) /* überprüft Dateiparameter */
74: char *name;
75: {
76:   struct FileLock *lock;
77:   eintrag=AllocMem (sizeof(struct FileInfoBlock),0);
78:   if (!eintrag)
79:   {
80:     printf ("Brauche CHIP-Memory für FileInfoBlock!\n");
81:     return (FALSE);
82:   }
83:   if (!(lock=Lock (name,ACCESS_READ)))
84:     return (FALSE);
```

Das letzte Beispiel schickt eine Datei verschlüsselt über die serielle Schnittstelle nach draußen. KRYPT ist somit auch eine interessante Variante für eine sichere DFÜ. Durch diese Art der online-Verschlüsselung werden eigene Daten vor allzu gierigen Hackern unkenntlich gemacht (z.B. in Mailboxen recht praktisch). Somit können "Sie" völlig gefahrlos Ihre persönlichen Daten durch die Netze schicken oder eigene Programme, die für andere als die Bestimmungsperson nutzlos sind. Wichtige Daten werden ohnehin fast immer verschlüsselt über die Leitungen geschickt, so machen es die Bundeswehr mit ihren Fernschreibern, die Banken mit ihren Überweisungen, warum also nicht auch "Sie" mit Ihrem AMIGA?

Bei der Passwordeingabe muß auch Groß- und Kleinschreibung korrekt sein. Falsche Passwörter decodieren die entsprechende Datei ebenfalls, bringen aber nicht das gewünschte Ergebnis. Dateien können auch mit mehreren Passwörtern mehrfach durch mehrere Aufrufe von KRYPT verschlüsselt werden. Solche Dateien werden durch die umgekehrte Reihenfolge derselben Passwörter wieder entschlüsselt.

Zur Funktionsweise des Programmes:

Die Funktion *main()* kontrolliert die Parametereingaben. Dabei wird auch bereits die Dateigröße der zu (de)codierenden Datei für einen dynamischen Diskpuffer ermittelt. In *make_it()* geschieht der wichtigste Teil von KRYPT. Quell- und Zieldatei werden vorbereitet und der Diskpuffer abhängig von der Quelldateigröße eingerichtet. Sobald die Quelldatei größer als 100 kB sein sollte, bleibt die Puffergröße bei 100 kB, um Speicher zu sparen. Man kann diese Pufferkonstante auch verändern, abhängig davon, wieviel Speicher zur Verfügung steht. Wird diese Konstante sehr niedrig gewählt, dauert auch das (De-)Codieren länger. 100kB Maxbuffer ist nicht besonders viel, wenn man bedenkt, daß sich z.B. auch der copy-Befehl immer soviel Buffer nimmt. Programmdateien über 100 kB sind sowieso selten. Trotzdem geht das Ver-/Entschlüsseln ohne viel Zeitverlust vonstatten.

Anhand der Endung der Quelldatei wird erkannt, ob ent- oder verschlüsselt wer-

```

85:
86:  if (!(Examine(lock,eintrag)))
87:  {
88:      UnLock(lock);
89:      return (FALSE);
90:  }
91:
92:  UnLock(lock);
93:  return (TRUE);
94: } /* end of check_datei */
95:
96: VOID hilfe () /* bei unzulässigen Parametern */
97: {
98:     printf ("\n\033[4mUsage:\033[0m\t%s quelldatei[%s] [ziel %s]
... \n", "KRYPT",endezeichen,endezeichen);
99:     printf ("\tDateien mit '%s' werden dekodiert\n",endezeichen);
100:    printf ("\tDateien ohne '%s' werden kodiert\n",endezeichen);
101:    exit ();
102: } /* end of hilfe */
103:
104: VOID make_it(quelldatei,zieldatei,buff_size) /*(De)Codierung*/
105: char *quelldatei,*zieldatei;
106: LONG buff_size;
107: {
108:  UBYTE *buffer,password[81];
109:  VOID kill_endung(),umrechnen(),codiere(),decodiere();
110:  BOOL verschluesseln;
111:  LONG number,handle1,handle2;
112:
113:  if ((handle1 = open (quelldatei,0,0))== -1)
114:  {
115:      printf ("%s kann nicht geöffnet werden!\n",quelldatei);
116:      goto WEITER;
117:  }
118:  if (buff_size > MAX_BUFF_SIZE) buff_size = MAX_BUFF_SIZE;
119:  if (!(buffer =(UBYTE *)AllocMem (buff_size,0L)))
120:  {
121:      printf ("Ich brauche mehr MEMORY\n");
122:      close (handle1);
123:      goto WEITER;
124:  }
125:  if (!(endung(quelldatei))) /*Test auf Codieren bzw Decodieren*/
126:  {
127:      if (!merke) strcat (zieldatei,endezeichen);
/*nicht bei prt: par: etc */
128:      verschluesseln = CODIERE;
129:  }
130:  else
131:  {
132:      if (endung(zieldatei)) kill_endung (zieldatei);
133:      verschluesseln = DECODIERE;
134:  }
135:  if ((handle2 = creat (zieldatei,0))== -1)
136:  if ((handle2 = open (zieldatei,1,0))== -1)
137:  {
138:      printf ("%s kann nicht geöffnet werden!\n",zieldatei);
139:      FreeMem (buffer,buff_size);
140:      close (handle1);
141:      goto WEITER;
142:  }
143:  #ifdef PASSWORD /* wird nur bei Zeile 20 compiliert */
144:  printf ("\n\033[1mEnter Password: \033[0m");
145:  scanf ("%80s",password);
146:  umrechnen (password); /* Passwort wird verarbeitet */
147:  #endif PASSWORD
148:  printf ("\n");
149:  if (!verschluesseln) printf ("DE"); /*Kontrollausgabe*/
150:  printf ("CODIERE \033[3m%s\033[0m to \033[3m%s\033[0m\n",
    quelldatei,zieldatei);
151:  do
152:  {
153:      number =read (handle1,buffer,buff_size);
154:      if (verschluesseln) codiere (buffer,number);
155:      else decodiere (buffer,number);
156:      write (handle2,buffer,number);
157:      } while (number==buff_size);
158:  FreeMem (buffer,buff_size);
159:  close (handle1);
160:  close (handle2);
161:  WEITER;;
162:  } /* end of make_it */
163:
164:
165:

```

→

den muß. Das Endezeichen habe ich mit '.k' ausgewählt, es kann aber auch verändert werden. Bei Passwortabfrage wird in der Funktion 'umrechnen' aus dem Passwort der Schlüssel errechnet, wobei das Passwort bis zu 80 Zeichen lang sein darf. Dabei wird folgendermaßen vorgegangen: Zuerst werden alle ASCII-Werte der einzelnen Buchstaben des Passwortes addiert. Der ASCII-Wert des letzten Buchstabens des Passwortes minus 1 (somit eignen sich auch Passwörter bestehend aus nur einem Buchstaben bestens) wird dann durch den addierten Wert dividiert. Man bekommt eine double-Zahl zwischen 0 und 1. Die ersten 12 Hinterkommazahlen dienen dann als Schlüssel, die von den Funktionen Codiere und Decodiere benutzt werden. Dabei wird immer je ein Block von 10 Zeichen der Quelldatei verändert. Die 10 Zeichen werden untereinander ausgetauscht und zwar elfmal, wobei die 12 Schlüsselzahlen als Feldindex fungieren, z.B.:

```
G u t e n   T a g
.
0 1 2 3 4 5 6 7 8 9   vertausche 3 und 9
G u . e n   T a g e   etc.
e u . G n   T a g e
g u . G n   T a e e
g u . G n e T a e
g u . G n a T e e
g u e G n a T . e
g u n G e a T . e
. u n G e a T g e
. u n G e a g T e
. u e G n a g T e
. e u G n a g T e
```

So, fertig dies ist die verschlüsselte Version von "Guten Tag." und entspricht der Funktionsweise von *Codiere()*. *Decodiere()* beschreibt genau den umgekehrten Weg. Verzichtet man auf die Passwordeingabe, so dienen die Zahlen in Zeile 32 als Schlüssel. Man kann auch den Algorithmus der Verschlüsselung noch weiter komplizieren, indem man z.B. den Schlüssel für jeden Zehnerblock laufend verändert, oder einen Pseudozufallsgenerator einsetzt. Hier kann man seine eigenen Ideen umsetzen und hat unendlich viele Möglichkeiten den Verschlüsselungsvorgang noch unberechenbarer zu gestalten. Krypt bietet da sicherlich noch viel Platz für Experimente.

Schließlich werden die Dateioperationen ausgeführt. Dabei ist der Dateipuffer so groß wie das zu verschlüsselnde Programm selber, bis maximal 100 kB (kann auch verändert werden). Programme über 100 kB werden dann in

```
166:  endung (s) /* Mal 2 eigene Routinchen */
167:  register char *s;
168:  {
169:  while (*s) s++;
170:  if (((*(--s)) == endezeichen[1]) && ((*(--s)) == endezeichen[0]))
      return (1);
171:  return (0);
172:  }
173:  /* end of endung */
174:
175:  VOID kill_endung (s)
176:  register char *s;
177:  {
178:  while (*s) s++;
179:  *(s-2) = 0;
180:  } /* end of kill_endung */
181:
182:  VOID decodiere (reihe,number)
183:  UBYTE reihe [];
184:  LONG number;
185:  {
186:  register UBYTE zeichen;
187:  register long i,j;
188:
189:  for (j=0;j<number-9;j+=10)
190:  {
191:    for (i=KEYS-2;i>=0;i--)
192:    {
193:      zeichen = reihe [key[i]+j];
194:      reihe [key[i]+j] = reihe [key[i+1]+j];
195:      reihe [key[i+1]+j] = zeichen;
196:    }
197:  }
198:  } /* end of decodiere */
199:
200:  VOID codiere (reihe,number)
201:  UBYTE reihe [];
202:  long number;
203:  {
204:  register UBYTE zeichen;
205:  register long i,j;
206:
207:  for (j=0;j<number-9;j+=10)
208:  {
209:    for (i=0;i<KEYS-1;i++)
210:    {
211:      zeichen = reihe [key[i]+j];
212:      reihe [key[i]+j] = reihe [key[i+1]+j];
213:      reihe [key[i+1]+j] = zeichen;
214:    }
215:  }
216:  } /* end of codiere */
217:
218:  #ifdef PASSWORD /* wird nur bei Zeile 20 compiliert */
219:
220:  VOID umrechnen (password) /* Passwort als Schlüssel */
221:  UBYTE password [];
222:  {
223:  unsigned int i,zahl=0;
224:  double kommazahl;
225:
226:  for (i=0;i<strlen(password);i++) zahl += password [i];
227:  kommazahl = (password [i-1]-1) / (double)zahl;
228:  for (i=0;i<KEYS;i++)
229:  {
230:    key [i] = (UBYTE) (kommazahl*=10);
231:    kommazahl -= key[i];
232:  }
233:  } /* end of umrechnen
234:  n */
235:  #endif PASSWORD
236:  /* ENDE LISTING */
```

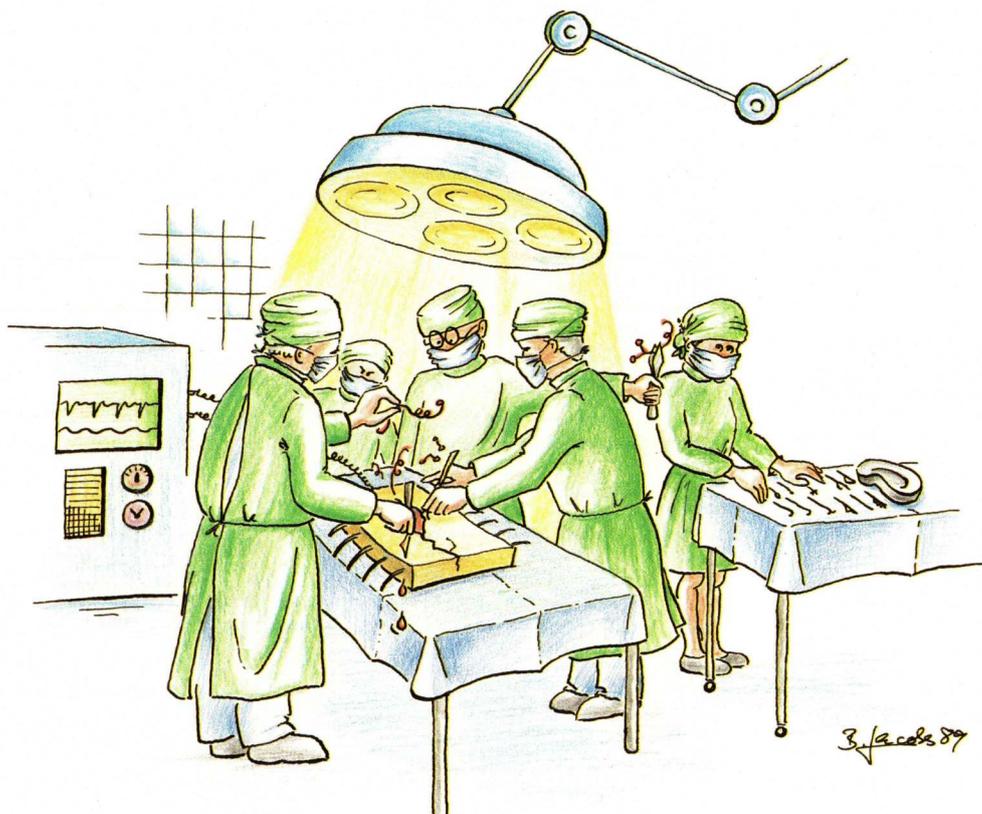
Blöcken von 100kB geladen, (de)codiert und wieder abgespeichert. Bei kleineren Programmen wird dies in einem Durchgang durchgeführt, wobei der

Puffer dynamisch angepaßt ist. Zuallerletzt werden noch alle Ressourcen geschlossen und das Programm beendet.

VON DANIEL GEMBRIS

Prozessordoktor

Ein neuer Debugger
für den AMIGA



Bekanntlich ist in der KICKSTART ein Debugger namens ROM-Wack integriert. Der Debugger ist sehr nützlich - vorausgesetzt, man nennt ein 9600-Baud-Terminal sein eigen. Fragt sich nur, wer ein solches besitzt, zumal der C64 und die meisten 8 Bit-Computer dafür nicht in Frage kommen, weil sie die geforderte Geschwindigkeit nicht schaffen.

A bhilfe schafft das Programm "NewDebugger", welches in Assembler geschrieben und resetfest ist. Es ersetzt den internen Debugger durch einen neuen, der beim Aufruf die Inhalte der Daten- und Adreßregister sowie des ConditionCode-(Status-) Registers und des UserStackpointers in einem Alert ausgibt. Der Debugger kann, wenn die linke Maustaste gedrückt wird, mit einem RTS beendet werden oder mit einem Reset, wenn die rechte Maustaste gedrückt wird. Das Komplizierteste an NewDebugger ist nicht der Debugger selbst, sondern die

```

NewDebugger (w) by Daniel Gembris
D0-D7 :
$ 00123456 00654321 00ABCDEF 00FEDCBA 00123456 00654321 00ABCDEF 00FEDCBA
A0-A6 :
$ 00123456 00654321 00ABCDEF 00FEDCBA 00123456 00654321 00ABCDEF
USP : $ 00C7FFFA      CCR : $ 00000000

RTS                                RESET

```

```

BarDisk 1:
Trashcan (dir)
Audiomaster (dir)
DPaint_III (dir)
.info
Create_A_SA
Disk.info
Fantavision.info
GoAniga! Date1.info
Jeane.info
Pal-DP.info
PPage.info
Audiomaster.info
Create_A_Shape.info
DPaint_III.info
getkick
Hochzeit.pic
PageStream.info
Pixnate.info
Sculpt-3D.info

```

Die Registerinhalte werden in einem Alert ausgegeben.

Routine, die dem Betriebssystem "verclickert", daß ein anderer Debugger aufgerufen werden soll. Zum einen wird mit der Exec-Library-Funktion "SetFunction" die Debug-Funktion, ebenfalls aus der Exec-Library, so modifiziert, daß beim Aufruf dieser Funktion der neue Debugger aufgerufen wird. Zum anderen wird noch ein Vektor namens "DebugEntry", in der Exec-Base-Structure, auf den neuen Debugger umgebogen. Leider läßt sich aber mit diesen beiden Methoden nicht erreichen, daß bei einer Gurumeditation der neue Debugger aufgerufen werden kann. Der Prozessor springt nämlich, wenn man bei einer Gurumeditation die rechte Maustaste drückt, direkt, d.h. ohne über einen Vektor im RAM, in den ROM-Wack. Das macht einen Griff in

die Trickkiste erforderlich, wenn man nicht seine Kickstart umändern will, wovon ich aber keinen abhalten möchte.

So werden von "NewDebugger" die Prozessor-Exception-Vektoren auf den neuen Debugger umgebogen. Das führt dazu, daß, sobald der Prozessor in eine Guru-Routine springen "will", der Debugger aktiv wird. Wenn dann die linke Maustaste gedrückt wird, wird die ursprüngliche Exception-Routine aufgerufen. Wenn man sich den Anblick einer Gurumeditation ersparen will, kann man die rechte Maustaste drücken - der AMIGA wird zurückgesetzt.

Dieser Trick klappt selbstverständlich nur bei Gurumeditationen, die durch eine Prozessor-Exception hervorgeru-

fen wurden. Von dieser Art sind sehr viele, wenn nicht sogar die meisten Abstürze, besonders wenn man selber in Assembler oder C programmiert. Da die Prozessor-Exception-Vektoren nur einmal umgebogen werden sollten, wenn man noch in den Genuß eines Absturzes kommen möchte, wird, wenn versucht wird, das Programm zu starten, obwohl die Vektoren schon umgebogen sind, das Programm vorzeitig abgebrochen, nachdem eine entsprechende Meldung in Form eines Alerts ausgegeben wurde.

In der KICKSTART 4/89 wurde ausführlichst der Aufbau von resetfesten Programmen besprochen; daher soll hier auf eine Erklärung der für die Resetfestigkeit verantwortlichen Programmteile verzichtet werden.

Zum Schluß sei noch erwähnt, daß das Programm absolut Turbo-AMIGA (MC 68020+881) freundlich ist. So wird das Status-Register mit der Exec-Funktion "GetCC" ausgelesen und der Exception-Vektor für Privilegsverletzungen nicht umgebogen, weil dort schon ein Programm eingetragen sein könnte, daß den "move sr,ea"- durch den "move ccr,ea"- Befehl ersetzt.

Das Programm "NEWDEBUGGER" kann entweder mit KICKASS, oder SEKA eingegeben werden. Allerdings fehlt die "end-Endung", die bei diesem Assembler am Ende des Listings angefügt werden muß.

END of KICK

```

1: ; Prg: NewDebugger
2: ; Autor: Daniel Gembris
3: ; (c) MAXON Computer GmbH
4: ; KICKSTART 1989
5:
6: execbase = 4
7: AllocMem = -$c6
8: AllocAbs = -204
9: SetFunction = -420
10: Debug = -114
11: OpenLib = -408
12: CloseLib = -414
13: GetCC = -528
14: require = 2
15: CoolCapture = 46
16: ColdCapture = 42
17: DebugEntry = 66
18: DisplayAlert = -90
19: toffset0 = buserror-ndebuga
20: toffset1 = addresserror-ndebuga
21: toffset2 = illegalop-ndebuga
22: toffset3 = divisionbyz-ndebuga
23: toffset5 = aopex-ndebuga

```

```

24: toffset6 = fopex-ndebuga
25: toffset7 = uninitinter-ndebuga
26: toffset8 = unpermittedinter-ndebuga
27:
28: move.l $8,d0 ;testen, ob NewDebugger
29: and.l #$0F00000,d0 ;bereits installiert
30: bne notyet ;noch nicht
31: move.l execbase,a6 ;Intuition-Lib. öffnen
32: lea i_name,a1
33: move.l #0,d0
34: jsr OpenLib(a6)
35: move.l d0,i_base
36: move.l d0,a6
37: beq Ende0
38: move.l #0,d0 ;Alert ausgeben
39: lea String2,a0
40: move.l #48,d1
41: jsr DisplayAlert(a6)
42: move.l execbase,a6 ;Library schließen
43: move.l i_base,a1
44: jsr CloseLib(a6)
45: Ende0: rts
46: notyet:

```

```

47:  move.l  execbase,a6 ;Installationsroutine (IR)
48:  move.l  #Ende-Anfang,d0 ;für neuen Debugger
                                resetfest machen
49:  move.l  #require,d1
50:  jsr  AllocMem(a6) ;Speicher für IRanfordern
51:  tst.l  d0
52:  beq  fehler ;hat nicht geklappt
53:  move.l  d0,a1
54:  move.l  a1,CoolCapture(a6) ;CoolCapture-Vektor
                                auf IR umbiegen
55:  clr.l  ColdCapture(a6) ;ColdCapture-Vektor
                                wird nicht benötigt
56:  move.w  #Ende-Anfang,d0 ;IR kopieren
57:  lea  Anfang,a0
58:  loop1: move.b  (a0)+,(a1)+
59:  dbra  d0,loop1
60:  jsr  Prüfsumme ;Prüfsumme neu berechnen
61:  fehler: movem.l  d0-d7/a0-a6,-(sp)
                                ;Register retten
62:  jmp  weiter
63:
64:  Anfang: ;====Installationsroutine====
65:  movem.l  d0-d7/a0-a6,-(sp) ;Register retten
66:  move.l  execbase,a6 ;IR vor Löschung schützen
67:  move.l  #Ende-Anfang,d0
68:  lea  Anfang(PC),a1
69:  jsr  AllocAbs(a6)
70:  lea  Anfang(PC),a1 ;CoolCapture-Vektor neu
71:  move.l  a1,CoolCapture(a6)
72:  clr.l  ColdCapture(a6)
73:  weiter:
74:  move.l  #ndebugE-ndebugA,d0 ;Speicher für
                                neuen Debugger besorgen
75:  move.l  execbase,a6
76:  move.l  #require,d1
77:  jsr  AllocMem(a6)
78:  lea  adresse(PC),a0 ;Adresse merken
79:  move.l  d0,(a0)
80:  beq  error ;kein Speicher
                                verfügbar
81:  move.l  d0,DebugEntry(a6) ;Debugger-Vektor
                                umbiegen
82:  jsr  Prüfsumme(PC) ;Prüfsumme neu
                                berechnen
83:  lea  adresse(PC),a2 ;Trap-Vektoren
                                auf Debugger umbiegen
84:  move.l  (a2),a2
85:  lea  TrapData(PC),a1 ;ab TrapData werden
                                die Adressen, auf
86:  ;die die Trap-Pointer
                                zeigen, abgelegt
87:  lea  Traps(PC),a3 ;ab Traps stehen
                                die Trap-Adressen
88:  lea  TrapOffsets(PC),a4 ;ab TrapOffsets
                                stehen die Offsets -
89:  move.w  #7,d0 ;sie werden zu ndebugA
                                addiert - für die
90:
91:  loop5: move.l  (a3)+,a5 ;neue Trap-Routinen
92:  move.l  (a5),(a1)+
93:  move.l  (a4)+,d1
94:  lea  0(a2,d1),a0
95:  move.l  a0,(a5)
96:  dbra  d0,loop5
97:
98:  lea  adresse(PC),a1 ;Routine kopieren und ...
99:  move.l  (a1),a1
100:  move.l  #ndebugE-ndebugA-1,d1
101:  lea  ndebugA(PC),a0
102:  loop3: move.b  (a0)+,(a1)+
103:  dbra  d1,loop3
104:  move.l  a6,a1 ;...als neuen Debugger
                                deklarieren
105:  move.l  #Debug,a0
106:  lea  adresse(PC),a2
107:  move.l  (a2),d0
108:  jsr  SetFunction(a6)
109:  error: movem.l  (sp)+,d0-d7/a0-a6 ;Register mit
                                alten Werten
                                beschreiben
110:  rts
111:  adresse: dc.l  0
112:
113:  ndebugA: ;der neue Debugger:
114:  movem.l  d0-d7/a0-a7,-(a7) ;Register auf Stack
115:  move.l  #$fffffff,d0 ;Aufruf erfolgte
                                nicht durch Trap
116:  dbug:
117:  lea  tnumber(PC),a0 ;Trap-Nummer merken
118:  move.l  d0,(a0)
119:  move.l  execbase,a6 ;Status-Register
                                mit Exec-Funktion
120:  jsr  GetCC(a6) ;"GetCC" auslesen
121:  lea  ret3(PC),a0 ;und Inhalt merken
122:  move.l  d0,(a0)
123:  move.w  #15,d0 ;Register-Daten
                                vom Stack
124:  lea  ret(PC),a0 ;in den Speicher
                                kopieren
125:  loop0: move.l  (a7)+,(a0)+
126:  dbra  d0,loop0
127:  move.w  #15,d0 ;..und wieder auf
                                den Stack
128:  loop4: move.l  -(a0),-(sp)
129:  dbra  d0,loop4
130:  lea  PufferA(PC),a1 ;Inhalt der
                                Daten-Register
131:  lea  ret(PC),a0 ;nach ASCII konv
                                ertieren und
132:  move.w  #7,d4 ;in die Alert-
                                Datenliste eintragen
133:  jsr  hex(PC) ;D4: Anzahl der
                                umzuwandelnden LWs
134:  ;A0: Adresse ab der die umzuwandelnden
135:  ; Longwords stehen
136:  ;A1: Adresse ab der die umgewandelten
137:  ; Longwords abgelegt werden sollen
138:  lea  PufferB(PC),a1 ;..nun den Inhalt
                                der Adressregister
139:  lea  ret1(PC),a0
140:  move.w  #6,d4
141:  jsr  hex(PC)
142:  lea  PufferC(PC),a1 ;..und jetzt den
                                Inhalt des USP
143:  lea  ret2(PC),a0
144:  move.w  #0,d4
145:  jsr  hex(PC)
146:  lea  PufferD(PC),a1 ;..und schließlich
                                den Inhalt des CCR
147:  lea  ret3(PC),a0
148:  move.w  #0,d4
149:  jsr  hex(PC)
150:  move.l  execbase,a6 ;Intuition-Library
                                öffnen
151:  lea  i_name(PC),a1
152:  move.l  #0,d0
153:  jsr  OpenLib(a6)
154:  lea  i_base(PC),a0
155:  move.l  d0,(a0)
156:  move.l  d0,a6
157:  beq  error2
158:
159:  move.l  #0,d0 ;Alert-Nummer
160:  lea  String(PC),a0 ;Anfangsadresse
                                der Alert-Datenliste
161:  move.l  #120,d1 ;Höhe des Alerts
162:  jsr  DisplayAlert(a6) ;Alert anzeigen
163:  lea  save0(PC),a0 ;Rückgabewert retten
164:  move.l  d0,(a0)
165:  lea  i_base(PC),a1
166:  move.l  (a1),a1 ;Intuition-Library
                                schließen
167:  move.l  execbase,a6
168:  jsr  CloseLib(a6)
169:  lea  save0(PC),a0 ;Rückgabewert
                                auswerten
170:  move.l  (a0),d0
171:  beq  rset ;rechte Maustaste
                                betätigt
172:  error2: lea  tnumber(PC),a0 ;in eine Trap-
                                Routine springen?
173:  cmp.l  #$fffffff,(a0)
174:  beq  Finito ;nein!
175:
176:  lea  TrapData(PC),a0 ;Rücksprungadresse
                                ermitteln →

```

```

177: lea tnumber(PC),a1
178: move.l (a1),d0
179: lea radr(PC),a2 ;in radr wird
Rücksprungadresse gespeichert
180: move.l 0(a0,d0),(a2)
181: movem.l (sp)+,d0-d7/a0-a7
182: move.l radr(PC),-(sp) ;in Trap-Routine
springen

183: rts
184:
185: Finito: movem.l (sp)+,d0-d7/a0-a7
186: rts ;Programm mit RTS
beenden
187: rset: jmp $fc0000 ;Programm durch
Reset beenden

188:
189: hex: move.w #7,d3 ;Zähler der
Nibbles auf 8-1
190: move.l (a0)+,d1 ;ein Longword aus
Speicher holen

191: loopx:
192: rol.l #4,d1 ;oberes Nibble
nach unten,
193: move.w d1,d2 ;in D2 schreiben
194: and.w #$0f,d2 ;oberes Nibble
ausblenden
195: add.w #$30,d2 ;$30 addieren
196: cmp.w #$3a,d2 ;wenn Ziffer:
197: bcs we2 ;fertig
198: add.w #7,d2 ;sonst 7 addieren
199: we2: move.b d2,(a1)+ ;Zeichen in Alert-
Datenliste ablegen
200: dbra d3,loopx ;das ganze 8 mal
201: move.b #$20,(a1)+ ;umgewandeltes LW
mit Space abschließen
202: dbra d4,hex ;evt. nächstes
Langwort umwandeln

203: rts
204:
205: Prüfsumme: ;Prüfsumme neu
206: move.l execbase,a6
207: clr.l d1
208: lea 34(a6),a0
209: move.w #$16,d0
210: loop2: add.w (a0)+,d1
211: dbra d0,loop2
212: not.w d1
213: move.w d1,82(a6)
214: rts
215:
216: buserror: movem.l d0-d7/a0-a7,-(sp)
217: move.l #$0,d0
218: jmp dbug(PC)
219: addresserror: movem.l d0-d7/a0-a7,-(sp)
220: move.l #4,d0
221: jmp dbug(PC)
222: illegalop: movem.l d0-d7/a0-a7,-(sp)
223: move.l #8,d0
224: jmp dbug(PC)
225: divisionbyz: movem.l d0-d7/a0-a7,-(sp)
226: move.l #12,d0
227: jmp dbug(PC)
228: aopex: movem.l d0-d7/a0-a7,-(sp)

```

```

229: move.l #16,d0
230: jmp dbug(PC)
231: fopex: movem.l d0-d7/a0-a7,-(sp)
232: move.l #20,d0
233: jmp dbug(PC)
234: uninitinter: movem.l d0-d7/a0-a7,-(sp)
235: move.l #24,d0
236: jmp dbug(PC)
237: unpermittedinter: movem.l d0-d7/a0-a7,-(sp)
238: move.l #28,d0
239: jmp dbug(PC)
240: radr: dc.l 0
241: tnumber: dc.l $fffffff
242: save0: dc.l 0
243:
244: string:
245: dc.b 0,16,16," NewDebugger by Daniel Gembris "
246: dc.b 0,255,0,16,26,"D0-D7: ",0,255,0,16,36,"$"
247: PufferA:
248: blk.b 72
249: dc.b 0,255
250: dc.b 0,16,46,"A0-A6: ",0,255,0,16,56,"$"
251: PufferB:
252: blk.b 63
253: dc.b 0,255,0,16,66,"USP: $"
254: PufferC:
255: blk.b 9
256: dc.b " CCR: $"
257: PufferD:
258: blk.b 9
259: dc.b 0,255
260: dc.b 0,50,86," RTS RESET "
261: dc.b 0,0
262: even
263:
264: string2:
265: dc.b 0,46,16," NewDebugger is already
installed!!!",0,255
266: dc.b 0,46,36," OK OK",0,0
267: even
268:
269: ret: blk.l 8
270: ret1: blk.l 7
271: ret2: blk.l 1
272: ret3: blk.l 1
273:
274: i_base: dc.l 0
275: i_name: dc.b "intuition.library",0,0
276: even
277:
278: Traps:
279: dc.l $000008,$00000c,$000010,$000014
280: dc.l $000028,$00002c,$00003c,$000060
281:
282: TrapData:
283: blk.l 8
284:
285: TrapOffsets:
286: dc.l toffset0,toffset1,toffset2,toffset3,
287: dc.l toffset5,toffset6,toffset7,toffset8
288:
289: ndebugE:
290: Ende:

```

VEKTORCHECK

Dem Betriebssystem auf die
Finger geschaut



Durch diese wunderbare Ergänzung können Sie leicht die Übersicht über alle Interrupts, Captures und Kickvektoren behalten. Bei Veränderungen dieser Werte brauchen Sie den Computer zwar nicht aus dem Fenster zu werfen, aber es ist eine Sache, die Sie wenigstens ernst stimmen sollte.

In erster Linie dient das Programm natürlich zum Überprüfen der Betriebssystem-Vektoren. Denkbar ist im Zuge der Zeit z.B. die Routine in eigene Programme einzubauen, um entscheidende Veränderungen festzustellen. So kann man unter anderem leicht erkennen, ob sich ein Virus im System befindet.

Sie können sich auch folgende Situation vorstellen: Jemand schreibt einen Virus, der ohne Angaben jeden Bootblock befällt. Auch bei näherem Hinsehen auf die Diskette ist keine Art von Virus zu

sehen. Dieses Programm müßte allerdings jede Art von Virus auf dem AMIGA aufdecken, da Virenprogramme nur diese Vektoren benutzen können/dürfen.

Der EXEC-Befehl hat keine besondere Syntax. Die Vektoren werden in tabellarischer, übersichtlicher Form wiedergegeben. Angezeigt werden die ersten 10 Interrupt-Vektoren, *Cold-*, *Cool-*, und *Warm-Capture* sowie *KickCheckSum*, *KickMemPointer* und *KickTagPtr* (von Byte Bandit benutzt).

Das Programm wurde mit dem Aztec-C-Compiler geschrieben. Der Compiler wird folgendermaßen aufgerufen:

```
cc exec.c +l
ln exec.o -lc32
```

In der Hoffnung, daß der Virus den KICKSTART-Lesern nun sein letztes Schnippchen geschlagen hat, ist dieses Programm wohl eine 'Muß-Ergänzung' für jeden Virusvernichter.

```
My Tools:
2> exec
Interrupts:
Nr.   Type   Priority   Data      ScType   PrType   Code
0:    61      64      fc07f8    2f       2f       fc07f8
1:    f0      00      fc07f8    2f       ba       fc0c52
2:    f0      00      fc0cd8    2f       ba       fc0d30
3:    61      62      fc0e04    2f       2f       fc0e4a
4:    61      5c      fc07fc    2f       2f       fc07fe
5:    61      56      fc0802    2f       2f       fc0804
6:    61      50      fc0808    2f       2f       fc080a
7:    0       7c      fc080e    2f       2f       fc0810
8:    81      00      fc0814    2f       aa       fc0816
Caputres:
ColdCapture   CoolCapture   WarmCapture
fc07ec        fc07ee        fc07f0
Kick-Pointers:
KickMemPtr    KickTagPtr    KickCheckSum
0             0             0
2>
```

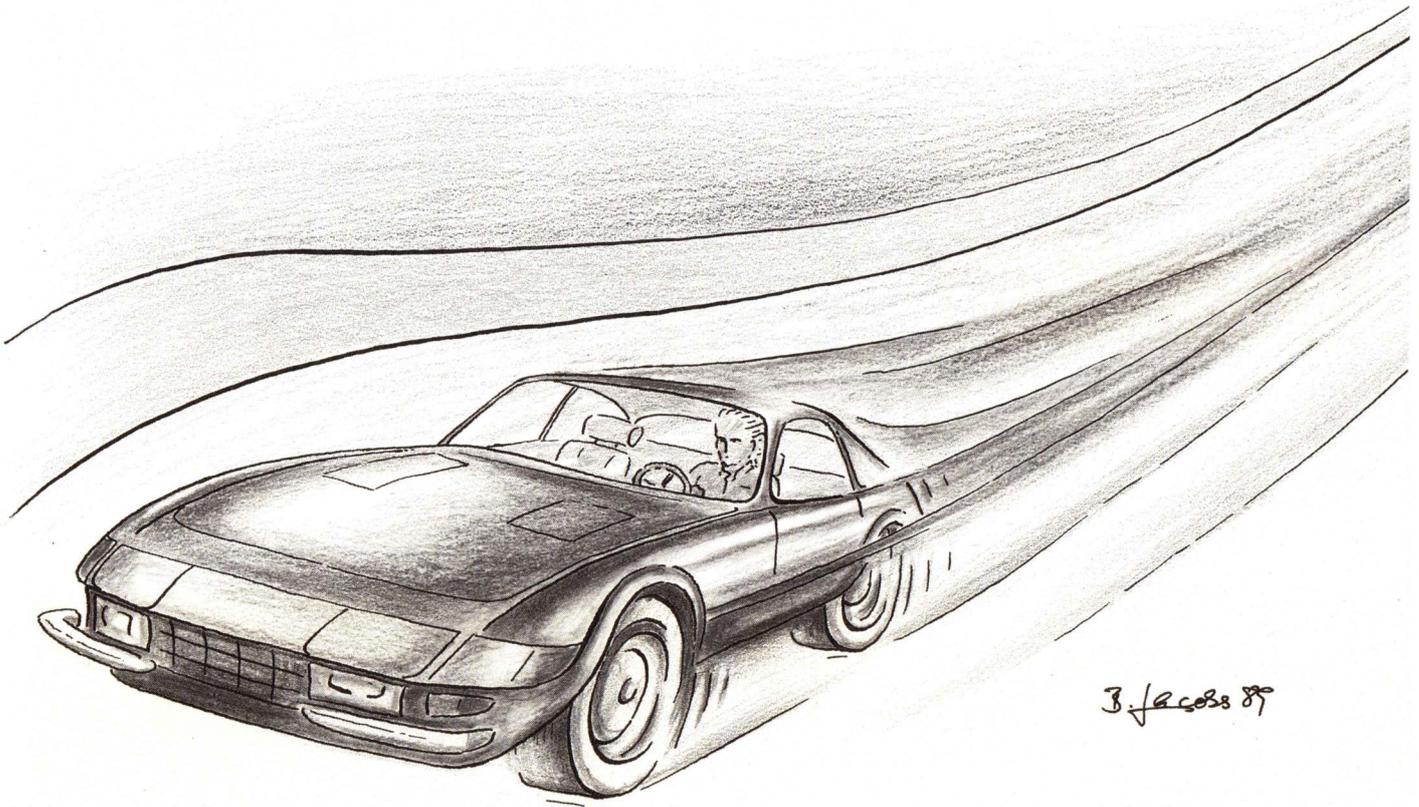
Das Programm EXEC gibt alle Betriebssystem-Vektoren übersichtlich aus.

```
1:  /* Autor: Thomas Wacker
2:    (c) MAXON Computer GmbH
3:    KICKSTART 1989
4:  */
5:
6:  #include <exec/execbase.h>
7:  #include <exec/nodes.h>
8:  #include <exec/interrupts.h>
9:  #define INTER ExecBase->IntVects
10:
11: struct ExecBase *ExecBase;
12: char c;
13:
14: main()
15: {
16: long i;
17: ULONG p1,p2,p3;
18: printf("Interrupts:\n");
19: printf("Nr.   Type   Priority   Data      ScType   PrType   Code\n");
20: printf("      PrType   Code\n");
21: for(i=0;i<17;i++)
22: {
23: if(INTER[i].iv_Data!=NULL)
24: printf("%2x:   %2x      %02x      %8x      %2x
          %2x      %8x\n",i,
25: (UBYTE) INTER[i].iv_Node->ln_Type,
26: (UBYTE) INTER[i].iv_Node->ln_Pri,
27: INTER[i].iv_Data,
28: (UBYTE) INTER[i].iv_Node->ln_Succ->ln_Type,
29: (UBYTE) INTER[i].iv_Node->ln_Pred->ln_Type,
30: INTER[i].iv_Code);
31: }
32:
```

```
33: Taste();
34:
35: printf("\033[ACaputres:          \n");
36: p1=ATOC(ExecBase->ColdCapture);
37: p2=ATOC(ExecBase->CoolCapture);
38: p3=ATOC(ExecBase->WarmCapture);
39: printf("ColdCapture   CoolCapture
          WarmCapture\n");
40: printf(" %8x      %8x      %8x\n",p1,p2,p3);
41:
42: Taste();
43:
44: printf("\033[AKick-Pointers:      \n");
45: printf("KickMemPtr    KickTagPtr
          KickCheckSum\n");
46: printf(" %8x      %8x      %8x\n\n",
          ExecBase->KickMemPtr,
          ExecBase->KickTagPtr,
          ExecBase->KickCheckSum);
47:
48: }
49:
50: Taste()
51: {
52: printf("Press Return!");
53: c=getchar();
54: }
55:
56: #asm
57: public _ATOC
58: _ATOC:
59: move.l 4(a7),d0
60: swap d0
61: rts
62: #endasm
```

VON OLIVER WAGNER

Gib Gas!



Der AMIGA ist nicht gerade einer der langsamsten Computer. Trotzdem wünscht man sich hin und wieder etwas mehr Geschwindigkeit, besonders bei Programmen, wo es auf reine Rechengeschwindigkeit ankommt und nicht auf Speicherschiebereien, die der Blitter souverän erledigt. Hierzu zählen z.B. Raytracing-Programme wie A-Render, C-Light, DBW-Render usw.

Möglich wäre jetzt eine Hardwarelösung wie z.B. ein 68020-Mathe-Prozessor oder ähnliches. Für Leute mit dickem Geldbeutel sicher kein Problem. Für den Normalanwender bleiben solche Sachen (im Moment) noch unerreichbar (hat man auch mal über die 16k-Erweiterung für den ZX-81 gesagt). Also suchen wir nach einer Softwarelösung.

Ein Ansatzpunkt ist der Buszugriff. Dazu muß man wissen, daß sowohl Prozessor als auch Blitter, Copper, Denise und Paula (nette Familie) über denselben Datenbus auf den Speicher zugreifen. Dies gilt auch für A500/2000-Besitzer mit 1 Mega-RAM. Dieses sog. Fast-RAM ist genauso langsam wie das Chip-RAM, da es auch von Agnus mitverwaltet wird und auf demselben Bus liegt. Wenn man nun den gesamten Coprozessor-Kram abschaltet, hat der Prozessor öfter die Möglichkeit, auf den Bus zuzugreifen und ist so logischerweise schneller (wer nicht glaubt, wieviele Zyklen allein die Grafikdarstellung frißt, sollte mal die Rechengeschwindigkeit testen bei einem Screen 640x400, 4 Bitplanes. Der AMIGA ist dann etwa 60% langsamer!!!). Man erkaufte diesen Vorteil damit, daß man nun kein Bild, keinen Ton usw. mehr hat; aber ist es wirklich

sor-Kram abschaltet, hat der Prozessor öfter die Möglichkeit, auf den Bus zuzugreifen und ist so logischerweise schneller (wer nicht glaubt, wieviele Zyklen allein die Grafikdarstellung frißt, sollte mal die Rechengeschwindigkeit testen bei einem Screen 640x400, 4 Bitplanes. Der AMIGA ist dann etwa 60% langsamer!!!). Man erkaufte diesen Vorteil damit, daß man nun kein Bild, keinen Ton usw. mehr hat; aber ist es wirklich

wichtig, beim Arbeiten mit DBW-Render magisch auf den blauen Bildschirm zu starren? Dafür dient das Programmchen "TURBO". Es ist in Assembler geschrieben und kann mit dem PD-Assembler A68k assembliert werden. Aufruf:

```
"a68k turbo.asm"
"blink turbo.o,small.lib to turbo ND SC SD".
```

Der Assembler findet sich auf KICK-START PD 56, auf der sich die *small.lib* befindet, die auch auf Fish 92 vorliegt.

Das Programm wird dann vom CLI mit "RUN TURBO" oder "RUNBACK TURBO" aufgerufen, von der Workbench einfach mit Doppelklick. Es wird dann ein kleines Window geöffnet. Um Turbo zu aktivieren, geht man folgendermaßen vor: erst das Fenster mit links anklicken (linke Maustaste, nicht linke Hand!), dann rechte Maustaste drücken. Sofort wird jegliche DMA abgeschaltet, und der Prozessor kann ungestört seiner Byte-Schaufelei fröhnen. Ein weiterer Druck auf die linke Maustaste oder ein

Tastendruck schaltet die DMA und damit den Bildschirm wieder ein. Ein Klick auf das Close-Gadget entfernt "TURBO" aus dem Speicher. Der Bildschirm wird übrigens auch dann wieder aktiv, wenn ein Requester erscheint. Also keine Angst, daß man einen "Disk-Full" verpaßt! Das Listing ist gut dokumentiert, so daß ich auf eine nähere Erklärung des Programms an dieser Stelle verzichten möchte.

END of KICK

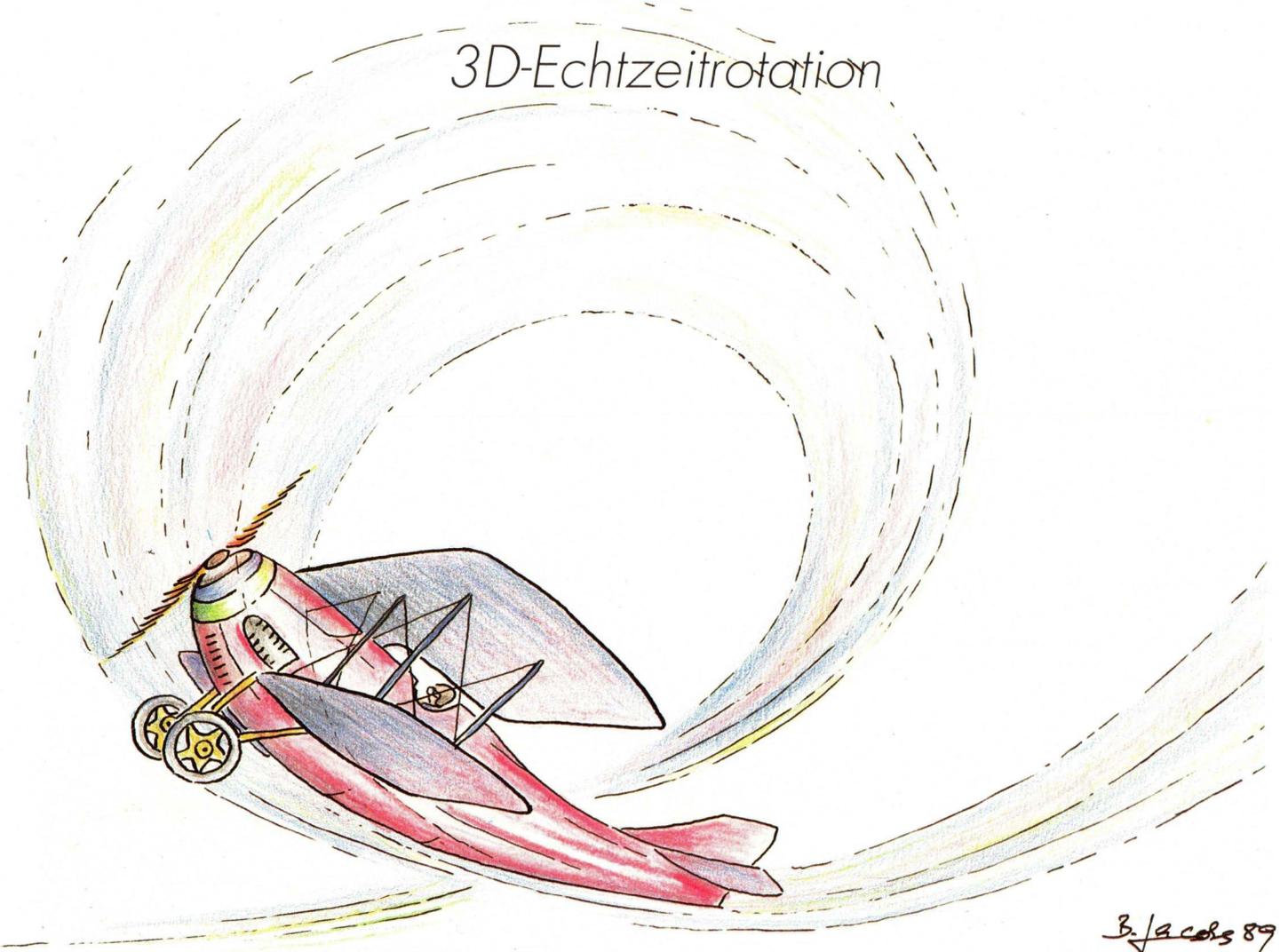
```
1: * TURBO.asm
2: * Autor: O. Wagner
3: *
4: * Schaltet sämtliche DMA ab, um Rechenvorgänge zu
5: * beschleunigen.
6: * Aktivierung: Fenster anklicken, rechts drücken.
7: * Jeder weitere Klick schaltet DMA wieder an.
8: *
9: * V2.0 für A68k,BLINK und SMALL.LIB
10: *
11: * a68k turbo.asm
12: * blink turbo.o,small.lib to turbo sc sd nd
13: call MACRO
14:     xref    _LVO\1
15:     jsr    _LVO\1(a6)
16:     ENDM
17:
18:     CODE    entry
19: * einsprung
20:     move.l 4,a6
21: * kommen wir von der WB?
22:     moveq #0,d6 * Exec Message
23:     move.l d6,a1
24:     call   FindTask
25:     move.l d0,a1
26:     tst.l  $ac(a1)
27: * nö
28:     bne.s  nowb
29: * message holen
30:     lea   $5c(a1),a0
31:     call  WaitPort
32:     call  GetMsg
33:     move.l d0,d6
34: nowb:
35: * intuition öffnen
36:     lea   intname(pc),a1
37:     call  OldOpenLibrary
38:     move.l d0,d7
39: * fenster öffnen
40:     lea   newwind(pc),a0
41:     move.l d7,a6
42:     call  OpenWindow
43:     move.l d0,a5
44:     tst.l  d0
45:     beq   nowind * nix Fenster
46: * linker unterer rand
47:     move.l 46(a5),a0 * screen
48:     move.w 12(a0),d0 * width
49:     move.w 14(a0),d1 * height
50:     sub.w #10,d1
51:     sub.w #126,d0
52:     move.l a5,a0
53: * fenster verschieben
54:     call  MoveWindow
55:     move.l 4,a6
56:     move.l 86(a5),a3
57: loop:
58: * auf nachricht warten
59:     move.l a3,a0
60:     call  WaitPort
61:     move.l a3,a0
62:     call  GetMsg
63:     move.l d0,a1
64:     move.l 20(a1),d5
```

```
65:     move.w 24(a1),d4
66: * zurück an Absender
67:     call  ReplyMsg
68:     cmp.w #$200,d5 * CloseWindow?
69:     beq.s quit
70:     cmp.w #$8,d5
71:     bne.s loop
72:     cmp.w #$69,d4
73:     bne.s loop
74: * DMA aus
75:     move.w #$0000000110101111,$dff096
76: * zweimal nachricht abwarten:
77: * 1) MOUSE_UP (rechts)
78: * 2) sonstige...
79:     move.l a3,a0
80:     call  WaitPort
81:     move.l a3,a0
82:     call  GetMsg
83:     move.l d0,a1
84:     call  ReplyMsg
85:     move.l a3,a0
86:     call  WaitPort
87:     move.l a3,a0
88:     call  GetMsg
89:     move.l d0,a1
90:     call  ReplyMsg
91: * DMA wieder ein
92:     move.w #$1000000110100000,$dff096
93:     bra.s loop
94: quit:
95:     move.l d7,a6
96:     move.l a5,a0
97: * Fenster zu!
98:     call  CloseWindow
99: nowind:
100:    move.l 4,a6
101:    tst.l  d6
102:    beq.s nomsg
103: * wb-msg beantworten
104:    call  Forbid
105:    move.l d6,a1
106:    call  ReplyMsg
107: nomsg:
108:    moveq #0,d0
109:    rts
110:
111: intname:
112:    dc.b  'intuition.library',0
113: newwind:
114:    dc.w  0,0,126,10
115:    dc.b  2,3
116:    dc.l  $4+$8+$200+$400+$4000+$8000+
117:          $10000+$40000+$80000
118:    dc.l  0,0
119:    dc.l  wint * titel
120:    dc.l  0,0
121: * Achtung: gehört noch zur Fenster-struktur
122: * (spart 6 byte)
123: wint:  dc.b  'TURBO',0,0,0
124:    dc.w  $1 * auf WB
125:
126: END
```

VON THOMAS GLOBISCH

Realtime

3D-Echtzeitrotation



Dieses kurze Demoprogramm richtet sich an alle C-Programmierer, die schon einmal versucht haben, dreidimensionale Objekte in Echtzeit (d.h. ruck- und flimmerfrei) drehen zu lassen. Dieses Programm zeigt anschaulich, wie man auch größere Objekte (solche mit mehr als 50 Linien) schnell berechnen und zeichnen kann, ohne auf Assembler zurückzugreifen.

So verwendet diese Demo keine Assembler-routinen und ist trotzdem schneller als viele Assemblerprogramme. Das Programm verwendet eine "echte" 3D-Darstellung, d.h. alle Linien und Flächen, die im Vordergrund liegen, werden automatisch größer gezeichnet, weiter hinten liegende Flächen kleiner.

Eine Beschreibung der Programmfunktionen erübrigt sich, mit der linken Maustaste kann man die Demo verlassen.

Das Programm arbeitet nach folgenden Prinzipien:

1. Es werden 2 Bitplanes angelegt, eine, die ständig angezeigt und eine zweite, auf der gezeichnet wird. Ist eine Zeichnung abgeschlossen, wird auf den Elektronenstrahl gewartet. Per Blitter werden die Bitplanes umkopiert (an dieser Stelle hätte man auch das Display auf die zweite Bitplane umschalten können und nun auf der ersten zeichnen, aber der Zeitgewinn ist so minimal, daß der Aufwand dafür zu groß ist). So verhindert man auch bei größeren Objekten ein Flackern.

2. Das Programm legt zu Anfang 3 Tabellen an, und zwar vom Datentyp *int*. Die erste Tabelle enthält alle Sinuswerte von 0 - 360 Grad multipliziert mit 256, die zweite Tabelle alle Cosinuswerte * 256, und die dritte Tabelle enthält die Perspektivewerte

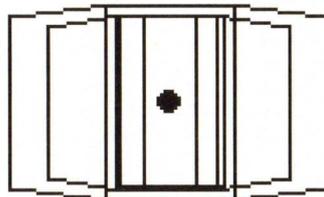
$$1,015^{\text{Tiefe}} * 256$$

$$\text{da } a^b = \exp(b * \ln(a))$$

Hier liegt auch schon der ganze Trick der enormen Geschwindigkeit - man bleibt immer im ganzzahligen Bereich und kann somit sehr schnell rechnen. Bei der anschließenden Berechnung der neuen Positionen muß man jetzt nur dafür sorgen, daß man zwischendurch wieder durch 256 teilt, um ein vernünftiges Ergebnis zu erhalten. Da aber 256 gerade 2^8 ist, kann man jetzt mit (Variable $\gg 8$) erreichen, daß alle Bits um 8 Stellen nach rechts verschoben werden, was einer Ganzzahldivision durch 256 entspricht. Das Verschieben von Bits geht übrigens wesentlich schneller als das direkte Teilen durch eine Zahl.

3. Es wird nicht der PolyDraw-Befehl verwendet, sodaß eine Beschränkung auf Objekte, die aus einer durchgezogenen Linie bestehen, entfällt. Wer mit der Perspektive nicht ganz zufrieden ist, kann durch leichtes (mit leicht meine ich die 3. Stelle nach dem Komma) Abändern des Wertes 1.015 in dem init-Unterprogramm [bei $\log_{10}(1.015)$] eine Verstärkung oder Abschwächung der Perspektive bewirken. Die Eingabe der Punkte geschieht in den Variablen *PaX*, *PaY*, *PaZ*. In *PaX* steht die X-, in *PaY* die Y- und in *PaZ* die Z-Koordinate eines Punktes. Der Punkt (0|0|0) liegt in der Mitte des Bildschirms, d.h. es können auch negative Werte eingegeben werden. Weiterhin muß eine Verbindungsvorschrift eingegeben werden.

3D - Echtzeit - Rotationsdemo



von Thomas Globisch

Das abgebildete Objekt wird in Echtzeit animiert.

In der Variablen *verba* steht immer der Anfangspunkt einer Linie und in *verbb* der Endpunkt. Ist *verba[5]=4* und *verbb[5]=9*, wird vom 4. zum 9. Punkt eine Linie gezeichnet. Einzelne Punkte ohne Verbindungsvorschrift werden nicht gezeichnet. ACHTUNG: Der erste berechnete Punkt steht in den Variablen *PaX[1]*, *PaY[1]* und *PaZ[1]*!! D.h. die Werte *PaX(Y,Z)[0]* sollte man auf 0 set-

zen, das gleiche gilt für die Verbindungsvorschrift!!

Programmiersprache:

Aztec C V3.6a

Compilieren:

cc 3D-Echtzeit.c

Linken:

ln 3D-Echtzeit.o -lm -lc

END of KICK

```

1:  /* 3 - D - Rotationsdemo      */
2:  /* benötigt mathtrans.library */
3:  /* Autor: Thomas Globisch    */
4:  /* (c) MAXON Computer GmbH 89 */
5:  /* KICKSTART 1989            */
6:
7:  #include <math.h>
8:  #include <libraries/mathffp.h>
9:  #include <graphics/gfx.h>
10: #include <graphics/gfxbase.h>
11: #include <graphics/rastport.h>
12: #include <graphics/view.h>
13: #include <graphics/display.h>
14: #include <functions.h>
15:
16: #define eckenzahl 40
17: #define linienzahl 52
18: #define XAchse 160
19: #define YAchse 100
20:
21: struct GfxBase *GfxBase;
22: struct RastPort rp, rp2;
23: struct View view, *oldview;
24: struct ViewPort VP;
25: struct BitMap Bitmap, Bitmap2;
26: struct RasInfo ri;
27:
28: int PaX[41] = { 0, -40, -55, -55, -40, -40, -55, -55, -40, 40, 55,
29:                55, 40, 40, 55, 55, 40, -55, -55, -55, -55, 55,
30:                55, 55, 55, 10, 20, 20, 10, -10, -20, -20, -10,
31:                10, 20, 20, 10, -10, -20, -20, -10 };
32: int PaY[41] = { 0, 40, 15, -15, -40, 40, 15, -15, -40, 40, 15,
33:                -15, -40, 40, 15, -15, -40, 2, 2, -2, -2, 2,
34:                2, -2, -2, 20, 10, -10, -20, -20, -10, 10, 20,
35:                20, 10, -10, -20, -20, -10, 10, 20 };

```

```

36: int PaZ[41] = { 0,20,20,20,20,-20,-20,20,20,
                20,20,20,20,-20,-20,-20,-20,2,
37:                -2,2,-2,2,-2,2,-2,25,25,25,25,
38:                25,25,25,25,-25,-25,-25,-25,
39:                -25,-25,-25,-25 };
40: int PbX[41],PbY[41],PbZ[41];
41: int PcX[41],PcY[41],PcZ[41];
42: short verba[53] = {0,1,2,3,4,5,6,7,5,2,3,9,10,
43:                  11,12,13,14,15,9,10,11,17,
44:                  18,19,20,21,22,23,24,25,26,
45:                  27,28,29,30,31,32,33,34,35,
46:                  36,37,38,39,36,37,38,39,40,
47:                  25,26,27,28,29,30,31,32};
48: short verbb[53] = {0,2,3,4,8,6,7,8,1,6,7,10,
49:                  11,12,16,14,15,16,13,14,15,
50:                  18,19,20,17,22,23,24,21,26,
51:                  27,28,29,30,31,32,25,34,35,
52:                  36,37,38,39,40,33,33,34,35,
53:                  36,37,38,39,40};
54: int sinus[361],cosinus[361];
55: int perspektive[301];
56: int xgrad = 0, ygrad = 0, zgrad = 0;
57: main()
58: {
59:     static short map[2] = { 0x0000, 0x0FFF };
60:     GfxBase = (struct GfxBase *)
61:     OpenLibrary("graphics.library",0L);
62:     oldview = GfxBase -> ActiView;
63:     InitView(&view);
64:     InitVPort(&VP);
65:     view.ViewPort = &VP;
66:     InitBitMap(&Bitmap,1L,320L,200L);
67:     InitBitMap(&Bitmap2,1L,320L,200L);
68:     InitRastPort(&rp);
69:     InitRastPort(&rp2);
70:     rp.BitMap = &Bitmap;
71:     rp2.BitMap = &Bitmap2;
72:     if((Bitmap.Planes[0] = AllocRaster(320L,200L))
73:     == NULL) exit();
74:     BltClear (Bitmap.Planes[0],
75:              (long)RASSIZE(320L,200L),0L);
76:     if((Bitmap2.Planes[0] =
77:     AllocRaster(320L,200L)) == NULL) exit();
78:     BltClear(Bitmap2.Planes[0],
79:              (long)RASSIZE(320L,200L),0L);
80:     ri.BitMap = &Bitmap;
81:     ri.RxOffset = 0L;
82:     ri.RyOffset = 0L;
83:     ri.Next = NULL;
84:     VP.RasInfo = &ri;
85:     VP.DWidth = 320L;
86:     VP.DHeight = 200L;
87:     VP.ColorMap = GetColorMap(2L);
88:     VP.Next = NULL;
89:     LoadRGB4 (&VP, &map, 2L);
90:     MakeVPort (&view, &VP);
91:     MrgCop (&view);
92:     LoadView (&view);
93:     SetAPen (&rp, 1L);
94:     SetAPen (&rp2, 1L);
95:     Move (&rp, 38L, 20L);
96:     Text (&rp, "3D - Echtzeit -Rotationsdemo", 29L);
97:     Move (&rp, 98L, 180L);
98:     Text (&rp, "Thomas Globisch", 15L);
99:     init();
100:    rotieren();
101: }
102: init()
103: {
104:     short lauf1;
105:     for (lauf1 = 0; lauf1 < 361; lauf1++)
106:     {
107:         sinus[lauf1] = (int) (sin(lauf1*PI/180)*256);
108:         cosinus[lauf1] = (int)
109:         (cos(lauf1*PI/180)*256);
110:     }

```

```

110:     for(lauf1 = -150; lauf1 < 150; lauf1++)
111:         perspektive[lauf1+150] = (int)
112:         (exp(lauf1*log10(1.015))*256);
113:     }
114:     closestuff()
115:     {
116:         FreeRaster(Bitmap.Planes[0],320L,200L);
117:         FreeRaster(Bitmap2.Planes[0],320L,200L);
118:         LoadView(oldview);
119:         CloseLibrary(GfxBase);
120:         exit();
121:     }
122:     rotieren()
123:     {
124:         register short lauf, lauf3;
125:         for(;;)
126:         {
127:             #asm
128:             btst    #6,$bfe001
129:             bne.s   ml
130:             jsr     _closestuff
131:             ml
132:             #endasm
133:             if((xgrad+=2)>360) xgrad -=360;
134:             if((ygrad+=5)>360) ygrad -=360;
135:             if((zgrad+=7)>360) zgrad -=360;
136:             for(lauf3 = 1; lauf3 <= eckenzahl; lauf3++)
137:             {
138:                 PbX[lauf3] = ((cosinus[xgrad] *
139:                 PaX[lauf3]) >> 8) - ((sinus[xgrad] *
140:                 PaY[lauf3]) >> 8);
141:                 PbY[lauf3] = ((cosinus[xgrad] *
142:                 PaY[lauf3]) >> 8) + ((sinus[xgrad] *
143:                 PaX[lauf3]) >> 8);
144:                 PcX[lauf3] = ((cosinus[ygrad] *
145:                 PbX[lauf3]) >> 8) - ((sinus[ygrad] *
146:                 PaZ[lauf3]) >> 8);
147:                 PbZ[lauf3] = ((cosinus[ygrad] *
148:                 PaZ[lauf3]) >> 8) + ((sinus[ygrad] *
149:                 PbX[lauf3]) >> 8);
150:                 PcY[lauf3] = ((cosinus[zgrad] *
151:                 PbY[lauf3]) >> 8) - ((sinus[zgrad] *
152:                 PbZ[lauf3]) >> 8);
153:                 PcZ[lauf3] = ((cosinus[zgrad] *
154:                 PbZ[lauf3]) >> 8)
155:                 + ((sinus[zgrad] *
156:                 PbY[lauf3]) >> 8);
157:                 PcX[lauf3] = (PcX[lauf3] * perspektive
158:                 [PcZ[lauf3] + 150] >> 8) + XAchse;
159:                 PcY[lauf3] = (PcY[lauf3] * perspektive
160:                 [PcZ[lauf3] + 150] >> 8) + YAchse;
161:             }
162:             ClipBlit(&rp2,50L,30L,&rp2,50L,30L,
163:                    220L,140L,(long) 0xc0);
164:             for(lauf3 = 1; lauf3 <= linienzahl; lauf3++)
165:             {
166:                 rp2.cp_x = PcX[verba[lauf3]];
167:                 rp2.cp_y = PcY[verba[lauf3]];
168:                 Draw(&rp2,(long)
169:                 PcX[verbb[lauf3]],(long)
170:                 PcY[verbb[lauf3]]);
171:             }
172:             WaitBOVP (&VP);
173:             ClipBlit (&rp2, 50L, 30L, &rp, 50L, 30L,
174:                      220L, 140L, (long) 0xc0);
175:         }
176:     } /* LISTING ENDE */

```


Absender
(Bitte deutlich schreiben)

Vorname/Name

Strasse/Nr.

PLZ/Ort

'TOP 12'
Mein Lieblingsspiel

Wettbewerbsbedingungen siehe
TOP 12 in diesem Heft

Bitte
freimachen

MAXON Computer
Redaktion KICKSTART
'TOP 12'
Industriestraße 26
6236 Eschborn

Kick
START
AMTIG ZITTSCHRIFFT

PD Bestellung

Absender
(Bitte deutlich schreiben)

Vorname/Name

Strasse/Nr.

PLZ/Ort

MAXON Computer
Redaktion KICKSTART
Industriestraße 26
6236 Eschborn

Postkarte

Bitte
freimachen

Kick
START
AMTIG ZITTSCHRIFFT

ABO

Absender
(Bitte deutlich schreiben)

Vorname/Name

Strasse/Nr.

PLZ/Ort

Heim Verlag
Heidelberger Landstr. 194
6100 Darmstadt 13

Bitte
freimachen

Kick
START
AMTIG ZITTSCHRIFFT

**Einzelheft- u.
Disketten Service**

Absender
(Bitte deutlich schreiben)

Vorname/Name

Strasse/Nr.

PLZ/Ort

Heim Verlag
Heidelberger Landstr. 194
6100 Darmstadt 13

Einzelheft- u.
Disketten Service

Die Floppy im Griff !!!

Teil 5:

Lese- und Schreibzugriffe

Heute geht es wie versprochen um Lese- und Schreibroutinen. Auch dafür stöbern wir ein wenig im Betriebssystem herum.

Die Synchronisationsmarkierung

Wenn wir uns nun dem Lesen widmen, sehen wir einem neuen Problem ins Auge: der Synchronisationsmarkierung. Wir haben zwei Möglichkeiten. Entweder wir lesen ohne Synchronisation ein, oder wir müssen die Synchronisationsmarke suchen. Im ersten Fall können die Daten von Diskette bitweise verschoben im Speicher stehen, wobei man über die Größe der Verschiebung keine Aussage machen kann. Im zweiten Fall müssen wir wissen, wie denn die Synchronisationsmarke aussieht, bevor wir danach suchen können. Das ist auch der Grund, warum einige Kopierprogramme nach dem Indexloch der Diskette synchronisieren, da so die Daten an der Stelle der Diskette abgelegt werden, wo sie gelesen wurden.

Aber wir wollen uns zuerst mal nur mit DOS-Disketten beschäftigen, die als Standardsynchronisation den Wert \$4489 haben. Dabei gibt es zwei Verfahren, die Synchronisationsmarke zu finden.

Zum einen kann man auf der Diskette selbst suchen und erst mit dem Lesen der Daten beginnen, wenn die Markierung gefunden wurde. Hierbei ist aber zu



beachten, daß die Sucherei nach Ablauf einer bestimmten Zeit abgebrochen wird, da sonst beim Nichtvorhandensein der Sync-Markierung ewig lange gesucht wird. Diese Methode kostet aber viel Zeit, und die Zeit ist etwas sehr Ernstzunehmendes bei Floppyzugriffen (finde ich jedenfalls). Also kann man auch direkt mit dem Einlesen der Daten beginnen und die Sync-Markierung im Speicher suchen. Doch das ist auch nicht ganz einfach, da (wie oben schon gesagt) die Daten im Speicher bitweise verschoben sein können.

Hier wendet man nun einen Trick an, indem man sich an den vier Nullbytes vor der Sync-Markierung orientiert. Wie auch immer die Daten verschoben sein mögen; dort steht entweder \$AAAA oder \$5555. Ist eine solche Markierung gefunden, wird danach die Sync-Marke erwartet. Sie wird gesucht, indem das Langwort hinter den Nullbytes mit den Werten einer Tabelle verglichen wird. Die Tabelle muß dabei alle möglichen Kombinationen der verschobenen Sync-Markierung beinhalten. Soll also nach einer anderen als der

Standard-Markierung (\$4489) gesucht werden, müssen die Daten der Tabelle entsprechend geändert werden. Die Tabellen finden wir in Listing 10.

Die eine Tabelle wird benutzt, wenn zuerst ein Taktbit gefunden wurde (\$5555). Die Sync-Markierung kann nun um 1, 3, 5, 7, 9, 11, 13 oder 15 Bits verschoben sein. Doch es handelt sich nicht um eine einfache Verschiebung, sondern die codierten Nullbytes werden mit verschoben, so daß auf der linken Seite keine Nullen eingefügt werden.

Die andere Tabelle wird benutzt, wenn zuerst ein Datenbit (\$AAAA) gefunden wurde. Hierbei kann die Sync-Markierung um 2, 4, 6, 8, 10, 12, oder 14 Bits verschoben sein. Listing 13 kann unter Angabe des Sync-Wortes diese Tabellen erstellen.

Doch nun zurück zur Methode, wie man die Sync-Markierung trotz Verschiebung sucht. Das Betriebssystem stellt eine recht komfortable Routine zur Verfügung, die diese Nullbytes findet und dann auch noch die verschobenen Daten im Speicher zurechtschiebt, so daß sie decodiert werden können. Diese Routinen finden wir in Listing 10. Auch hier handelt es sich nicht um ein selbständig laufendes Programm. Die Routine arbeitet nur mit fehlerfreien DOS-Disketten, da ich sämtliche Fehlerabfragen gestrichen habe. Die Routine wäre sonst um einiges länger.

Eine Anwendung aller bis jetzt besprochenen Routinen stellt Listing 11 dar, das einen Track einliest und decodiert. Listing 11 enthält jedoch noch nicht die Routinen zur Motorschaltung und zur Bewegung des Schreib-/Lesekopfes. In Teil 4 haben wir bereits den Motor ein und ausgeschaltet sowie den Kopf gesteppt. Damals mußten wir das Multitasking abschalten, damit mehrere gleichzeitige Zugriffe auf dasselbe Laufwerk ausgeschlossen waren. Diese Einschränkung wollen wir aber nun vermeiden.

Um ein Laufwerk dennoch für sich alleine benutzen zu können, gibt es im Betriebssystem eine Routine zum Anmelden eines Laufwerks. Hierbei wird der Wunsch (eine Message), auf die Hardware-Register zugreifen zu dürfen, an das Ende einer Liste gehängt. Der Task geht nun solange in Wartestellung, bis er eine Message erhält, daß die Register

```

1: ;Listing 10
2: readtrack:
3: ;Diese Routine liest einen Track und entfernt die Lücke.
4: ;A3 zeigt auf Laufwerks-Port
5: move.l a2,-(a7) ;A2 retten
6: move.l 78(a3),a2 ;Zeiger auf Load Puffer
7: moveq #1,d0 ;Wert für Motor an
8: bsr motorschalten ;Motor einschalten
9: moveq #0,d0 ;D0 löschen, da Wort-Operation
10: move.w 74(a3),d0 ;Tracknummer nach D0
11: bsr movehead ;Schreib-Lesekopf positionieren
12: lea 1668(a2),a0 ;Zeiger auf Anfang der Daten
13: move.w #14716,d0 ;Anzahl zu lesender Bytes
14: bsr read ;Track lesen
15: tst.l d0 ;Fehler beim Lesen?
16: beq noerror ;Nein, dann verzweigen
17: move.b d0,3(a2) ;sonst Fehlernummer übergeben
18: bra readexit ;und Routine verlassen
19: noerror:
20: bsr findsync ;Track richtigstellen und Lücke entfernen
21: move.b d0,3(a2) ;Sektornummer eintragen
22: readexit:
23: move.l (a7)+,a2 ;A2 zurückholen
24: rts ;Rücksprung
25:
26: read:
27: ;Einlesen eines Tracks von Diskette
28: ;A0 Zeiger auf richtige Position im Load Puffer
29: ;D0 enthält Anzahl zu lesender Bytes
30: lea setskllen,a1 ;Zeiger auf Routine, die DSKLEN setzt
31: movem.l a4/a2/d2,-(a7) ;Register retten
32: move.l a0,a2 ;Ladeadresse in A2
33: move.l d0,d2 ;Anzahl zu lesender Bytes
34: move.l a1,a4 ;Zeiger auf DSKLEN-Routine
35: bsr anmelden ;Laufwerk anmelden
36: move.b 65(a3),$bfd100 ;Motorbits setzen
37: move.w #$4000,$dff024 ;DSKLEN-Register vorbereiten
38: move.l #1000,d0 ;Wert für Warteschleife
39: bsr warten ;warten
40: lea $dff000,a1 ;Zeiger auf Hardware-Register
41: move.l a2,32(a1) ;Zeiger auf Puffer für gelesene Daten
42: move.w #$1002,156(a1) ;Sync-Interrupt sperren
43: move.w #$8002,154(a1) ;Disk-Block-Ready Interrupt freigeben
44: btst #2,$bfe001 ;Diskette im Laufwerk?
45: bne diskinserted ;Ja, dann verzweige
46: diskremoved:
47: moveq #29,d0 ;Fehlermeldung: keine Disk eingelegt
48: bra diskerror ;Routine verlassen
49: diskinserted:
50: lsr.w #1,d2 ;Anzahl Bytes in Anzahl Worte wandeln
51: ori.w #$8000,d2 ;DMA-Bit setzen
52: move.l d2,d0 ;Wert nach D0
53: jsr (a4) ;DSKLEN-Routine aufrufen
54: bsr waitmessage ;Auf Rückmeldung warten
55: lea $dff000,a1 ;Zeiger auf Hardware-Register
56: move.w #2,154(a1) ;Disk-Block-Ready Interrupt sperren
57: move.w #$4002,36(a1) ;Disk-DMA sperren
58: btst #2,$bfe001 ;Disk herausgenommen?
59: beq diskremoved ;Ja, Fehlermeldung
60: moveq #0,d2 ;Fehler-Flag auf OK
61: diskerror:
62: bsr abmelden ;Laufwerk abmelden
63: move.l d2,d0 ;Rückmeldung nach D0
64: movem.l (a7)+,a4/a2/d2 ;Register zurückholen
65: rts ;Rücksprung
66:
67: findsync:
68: ;Diese Routine sucht nach der Standard-Sync-Markierung
;($4489) und stellt
69: ;ggf. die Daten im Speicher richtig.
70: ;A3 zeigt auf den Laufwerks-Port
71: ;A6 zeigt auf die Trackdisk-Device-Struktur
72: movem.l a2/d2-d6,-(a7) ;Register retten
73: link a4,#-8 ;Eigener Stapel für Zwischenspeicherungen
74: move.l 78(a3),a2 ;Zeiger auf Load-Puffer (codierte Daten)
75: lea 1664(a2),a2 ;Zeiger auf Anfang der Daten
76: move.l a2,a0 ;Zeiger auf Daten in A0
77: addq.l #2,a0 ;Anfangsmarkierung überspringen
78: move.l #2748,d0 ;zuviel gelesene Bytes als Zähler
79: bsr findheader ;Block-Header suchen
80: cmpa.l #-1,a0 ;Header gefunden?
81: beq nosyncfound ;Nein, Fehler
82: move.l a0,d5 ;Zeiger auf Sync nach D5 →

```

freigegeben sind. Daneben gibt es natürlich auch eine Routine, die ein Laufwerk wieder abmeldet. Dann kann wieder ein anderer Task auf dieses Laufwerk zugreifen. Die Routinen zur Motorschaltung und zur Kopfbewegung laufen ebenfalls mit dem Multitasking, weshalb sie recht lang und kompliziert sind. Bei der Kopfbewegung gibt es ursprünglich eine Abfrage, die testet, ob der gewünschte Track negativ ist.

Da es bei uns keine negativen Tracknummern geben soll, habe ich diese Abfrage gestrichen. Bei normalen DOS-Disketten wird ein angegebener Track zwischen 0 und 160 nämlich immer gefunden. Für diejenigen, die die Routinen nicht mühevoll abtippen wollen, kann ich die absolute Adresse der Routinen im Betriebssystem angeben. Ich tue das erst jetzt, da man sich intensiv mit den Routinen beschäftigen sollte. Das geht besser, wenn man sie sich selbst verständlich macht, anstatt nur mit mehreren Sprungadressen zu arbeiten. Hier eine Tabelle der Adressen:

Name der Routine	absolute Adresse
blockcodierung	\$FEAADC
codlongword	\$FEAD46
coddatablock	\$FEAB4A
sendmessage	\$FEA6F2
waitmessage	\$FEA70A
psumme	\$FEADA4
decodeblock	\$FEACB2
decodelongword	\$FEAD8E
readtrack	\$FEA99E
read	\$FEA524
findsync	\$FEAFE2
findheader	\$FEAF4C
shiftblocks	\$FEB214
getrightlongword	\$FEB204
anmelden	\$FEADDC
abmelden	\$FEAE42
motorschalten	\$FEA462
warten	\$FEA4F0
movehead	\$FEA3DA
stepmotor	\$FEA4BE
writetrack	\$FEA5B4

Da es verschiedene Kickstart-Versionen gibt, muß man erst überprüfen, ob die Adressen auch wirklich stimmen. Die AMIGA 1000-Benutzer mit der HQC-Kickstart können obige Adressen benutzen.

Soviel zum Thema Lesen. Kommen wir nun zum Gegenteil: dem Schreiben.

Das Schreiben

Wie die Daten codiert sein müssen, sollte an dieser Stelle klar sein. Der Track sollte im Speicher bereits so aufgebaut sein, wie er auch auf der Diskette steht.

```

83:  move.l  d0,d2          ;Anzahl der zu verschiebenden Bits
84:  addq.l  #8,a0          ;Zeiger auf codiertes Header-Langwort
85:  tst.l   d2             ;müssen Bits verschoben werden?
86:  bne    shifthead     ;Ja
87:  move.l  (a0),-8(a4)   ;1. Langwort des codierten Headers
88:  move.l  4(a0),-4(a4) ;2. Langwort des codierten Headers
89:  bra    examinehead   ;Header untersuchen
90:  shifthead:
91:  bsr    getrightlongword ;Richtig geschobenen Header holen
92:  move.l  d0,-8(a4)    ;und speichern
93:  bsr    getrightlongword ;zweites Langwort des Headers holen
94:  move.l  d0,-4(a4)    ;und speichern
95:  examinehead:
96:  lea    -8(a4),a0      ;Zeiger auf gelesenen Header setzen
97:  bsr    decodelongword ;Header-Langwort decodieren
98:  move.l  d0,d3        ;decodierter Header nach D3
99:  moveq  #0,d0         ;D0 löschen, da Byte-Operation
100: move.b  d3,d0         ;Anzahl Blöcke bis zur Lücke
101: mulu   #1088,d0      ;Anzahl Bytes bis zur Lücke
102: move.l  d5,a0        ;Zeiger auf Sync-Markierung
103: move.l  a2,a1        ;zeigt auf Anfang der Daten im Load Puffer
104: move.l  d2,d1        ;Anzahl der zu verschiebenden Bits
105: move.l  d0,d4        ;Byte-Entfernung bis zur Lücke in D4
106: bsr    shiftblocks   ;Daten zurechtkopieren
107: moveq  #0,d2         ;D2 löschen, da Byte-Operation
108: move.b  d3,d2         ;Anzahl Blöcke bis zur Lücke
109: sub.l  #11,d2        ;maximal 11 Blöcke pro Track
110: neg.l  d2            ;Anzahl der Blöcke nach der Lücke
111: beq    nomoreblocks  ;dies ist der Block vor der Lücke
112: add.l  d4,d5         ;Adresse der Lücke bestimmen
113: move.l  #1160,d0     ;Anzahl der zu durchsuchenden Byte
114: move.l  d5,a0        ;Zeiger auf Lücke in A0
115: addq.l #2,a0         ;Anfangsmarkierung überspringen
116: bsr    findheader    ;Standard-Sync nach Lücke suchen
117: cmpa.l #-1,a0        ;Sync gefunden?
118: beq    nosyncfound  ;Nein, Fehler
119: move.l  d0,d1        ;Anzahl der zu verschiebenden Bits
120: move.l  a2,a1        ;zeigt auf Anfang der Daten im Load Puffer
121: adda.l d4,a1         ;Zieladresse für Daten
122: move.l  d2,d0        ;Anzahl der Blöcke nach Lücke
123: mulu   #1088,d0     ;Anzahl Bytes nach Lücke
124: bsr    shiftblocks   ;Daten zurechtkopieren
125: nomoreblocks:
126: move.l  a2,a0        ;Zeiger auf Daten im Load Puffer
127: adda.l  d4,a0        ;Adresse der neu kopierten Daten
128: bsr    connect      ;Ränder korrigieren
129: lea    11968(a2),a0 ;Zeiger auf Ende der Daten
130: move.w  #$aaa8,d0    ;Wert für Endmarkierung
131: btst   #0,-1(a0)    ;letztes Datenbit testen
132: beq    bitgelöscht  ;schon gelöscht
133: bclr   #15,d0       ;sonst Endmarkierungsbit löschen
134: bitgelöscht:
135: move.w  #$aaaa,(a2) ;Anfangsmarkierung eintragen
136: nosyncfound:
137: unlk   a4            ;Zwischenspeicher freigeben
138: movem.l (a7)+,a2/d2-d6 ;Register zurückholen
139: rts                ;Rücksprung
140:
141: findheader:
142: ;Sucht nach Standard-Sync-Markierung ($4489) im Block-Header
143: ;A0 zeigt auf die zu decodierenden Daten
144: ;D0 enthält Anzahl zu durchsuchender Bytes
145: movem.l a2/d2-d4,-(a7) ;Register retten
146: move.w  #$aaaa,d3    ;erstes Kontrollwort in D3
147: move.w  #$5555,d4    ;zweites Kontrollwort in D4
148: move.l  a0,a2        ;Zeiger auf Daten nach A2
149: adda.l  d0,a2        ;Zeiger auf Ende der Daten
150: findloop:
151: move.w  (a0)+,d2     ;aktuelles Wort in D2
152: cmp.w   d3,d2        ;Datenbit am Anfang?
153: beq    databitfound ;Ja
154: cmp.w   d4,d2        ;Taktbit am Anfang?
155: beq    taktbitfound  ;Ja
156: cmpa.l  a0,a2        ;Ende der Daten erreicht?
157: bhi    findloop     ;Nein, weitersuchen
158: notfound:
159: moveq  #-1,d0        ;Fehlermeldung in D0
160: move.l  d0,a0        ;Adresse auf Daten löschen
161: bra    nosync        ;keine Sync-Markierung gefunden
162: taktbitfound:
163: moveq  #15,d0        ;Zähler für Anzahl der verschobenen Bits
164: lea    takttable,a1 ;Zeiger auf Tabelle in A1
165: nullmarke:

```

Das heißt elf Sektoren jeweils mit Block-Header und Datenteil, die auch die richtigen Prüfsummen beinhalten. Nach den elf Sektoren folgt dann die Tracklücke, der sogenannte GAP. Der Schreibroutine wird also nur ein Zeiger auf den Speicherbereich übergeben, an dem der Track steht, falls ein gesamter Track geschrieben werden soll. Natürlich kann man auch eine variable Anzahl von Bytes schreiben, indem man eine Länge übergibt. Vom Aufbau her ist die Schreibroutine (Listing 14) ähnlich der Laderoutine.

Neu beim Schreiben ist, daß abhängig von der Position des Tracks auf der Diskette mit unterschiedlichen Schreibdichten geschrieben wird. Die Schreibdichte kann man in dem ADKCON-Register bestimmen über den sogenannten PRECOMP-Wert. Es gibt vier Möglichkeiten:

Bit 14	Bit 13	PRECOMP-Zeit
0	0	0 Nanosekunden
0	1	140 Nanosekunden
1	0	280 Nanosekunden
1	1	560 Nanosekunden

Das Betriebssystem nutzt nur zwei der vier Möglichkeiten aus. Von Zylinder Null bis 79 (einschließlich) wird die PRECOMP-Zeit auf Null gesetzt. Ab Zylinder 80 wird diese Zeit auf 140 Nanosekunden gesetzt. Beim Lesen wird in den Betriebssystemroutinen die PRECOMP-Zeit nie verändert, da die Routine zum Positionieren des Schreib-/Lesekopfes bei Tracknummern größer als 160 aussteigt (aussteigen, nicht abstürzen!). Die größte Tracknummer kann man als Wort in dem Laufwerks-Port festlegen (Offset 54). Bei einigen Experimenten habe ich festgestellt, daß die PRECOMP-Zeit auf das Lesen keinen Einfluß hat.

So, nun sind wir am Ende des Kurses gelangt. Als Anhang möchte ich noch zwei Strukturen erläutern, mit denen die Betriebssystemroutinen die ganze Zeit arbeiten. Jetzt dürfte so ziemlich alles verstanden werden, was die Strukturen beinhalten. Diese Information hätte zuvor wohl nur noch mehr Verwirrung gestiftet anstatt zum Verständnis beizutragen.

```

166: cmpa.l a0,a2 ;Ende der Daten erreicht?
167: bls notfound ;Ja, Fehler
168: move.w (a0)+,d1 ;nächstes Wort nach D1
169: cmp.w d2,d1 ;immer noch Kennung vor Sync-Markierung?
170: beq nullmarke ;Ja, nächstes Wort untersuchen
171: subq.l #2,a0 ;Zeigt nun auf vorheriges Wort
172: move.l (a0),d1 ;Langwort in D1
173: testsync:
174: cmp.l (a1)+,d1 ;Sync-Markierung in D1?
175: beq syncfound ;Ja, mögliche Sync ($4489) in D1
176: subq #2,d0 ;sonst Anzahl der verschobenen Bits
177: ;verringern
178: bge testsync ;solange D0>0, weitertesten
179: bra findloop ;dies war keine Sync, also weitersuchen
180: syncfound:
181: subq.l #4,a0 ;Zeigt auf Sync-Markierung
182: nosync:
183: movem.l (a7)+,a2/d2-d4 ;Register zurückholen
184: rts ;Rücksprung
185: databitfound:
186: moveq #14,d0 ;Zähler für Anzahl der verschobenen Bits
187: lea datatabelle,a1 ;Zeiger auf Tabelle setzen
188: bra nullmarke ;Sync-Markierung suchen
189:
190: shiftblocks:
191: ;Diese Routine kopiert die Daten an die Stelle im Load Puffer,
; an der die
192: ;sinnvollen Daten beginnen (Offset 1664). Falls die Daten
; verschoben sein
193: ;sollten, werden sie zurechtgerückt.
194: ;A0 zeigt auf Syncadresse eines Blocks
195: ;A1 zeigt auf Ziel für die richtigen Daten (Offset 1664 im
; Load Puffer)
196: ;A3 zeigt auf den Laufwerks-Port
197: ;A6 zeigt auf Trackdisk-Device-Struktur
198: ;D0 enthält die Anzahl der zu bearbeitenden Bytes
199: ;D1 enthält die Anzahl Bits, um die verschoben werden muß
200: link a2,#-30 ;Platz für QBLIT-Struktur
201: move.b d1,-8(a2) ;Anzahl der Verschiebungsbits eintragen
202: tst.l d1 ;müssen Bits verschoben werden?
203: beq noshift ;Nein
204: addq.l #2,d0 ;sonst ein Wort von Ende einschieben
205: noshift:
206: move.l d0,d1 ;Wert für
207: add.l #63,d1 ;BLTSIZE berechnen
208: and.w #$ffc0,d1 ;Breite des Blitterfensters auf Null
209: or.w #32,d1 ;Breite setzen
210: move.w d1,-10(a2) ;BLTSIZE in QBLIT-Struktur eintragen
211: movem.l d0/a0/a1,-22(a2) ;Werte in Struktur eintragen
212: move.l #shiftroutine1,-26(a2) ;Zeiger auf Funktion setzen
213: move.l a3,-4(a2) ;Laufwerks-Port eintragen
214: lea -30(a2),a1 ;Zeiger auf Anfang der Struktur
215: move.l a6,-(a7) ;A6 retten
216: move.l 56(a6),a6 ;Zeiger auf GfxBase
217: jsr qblit(a6) ;QBLIT-Funktion aufrufen
218: move.l (a7)+,a6 ;A6 zurückholen
219: bsr waitmessage ;Auf Reply-Message warten
220: unlk a2 ;Platz für QBLIT-Struktur freigeben
221: rts ;Rücksprung
222:
223: shiftroutine1:
224: ;evtl. lea $dff000,a0

```

a) Die Trackdisk-Device-Struktur:

Offset	Größe	Beschreibung
00	—	herkömmliche Library-Struktur
34	dc.w	Wort, um auf Langwort-Adresse zu kommen
36	dc.l	Zeiger auf Message-Port für Laufwerk 0
40	dc.l	Zeiger auf Message-Port für Laufwerk 1 (kein df1: => NULL)
44	dc.l	Zeiger auf Message-Port für Laufwerk 2 (kein df2: => NULL)
48	dc.l	Zeiger auf Message-Port für Laufwerk 3 (kein df3: => NULL)
52	dc.l	Zeiger auf ExecBase
56	dc.l	Zeiger auf GfxBase
60	dc.l	Zeiger auf Disk-Resource
78	dc.l	Zeiger auf Timer-Device
94	dc.l	Zeiger auf CIAB-Resource

■ Diese Struktur ist nicht vollständig, die wichtigen Einträge sind jedoch enthalten.

DAS GFA-BASIC

3.0

**KLAUS SCHNEIDER,
OLIVER STEINMEIER,
PETER FRITZEN**

BUCH

Lieferbar!

Als optimale Ergänzung zum Handbuch des neuen GFA-BASIC 3.0-Interpreters bietet sich dieses Buch an. In zwei Hauptteilen wird zunächst eine systematische Einführung in die Programmierung von BASIC unter Berücksichtigung der besonderen Fähigkeiten von GFA-BASIC auf dem Amiga gegeben. Hier werden dem Neuling vom ersten Einzeiler bis zu abstrakten Datentypen alle Möglichkeiten der strukturierten Programmierung mit zahlreichen, durch Flußdiagramme transparenter gemachten Beispielen nahegebracht. Doch auch BASIC-erfahrene Programmierer lernen hier die neuen Strukturen kennen, die sich doch sehr von denen anderer Dialekte unterscheiden.

Der zweite Teil baut auf dem ersten auf und vermittelt weitere Kenntnisse der Programmierung, anhand von Programmen, die wiederum ausführlich beschrieben und erklärt sind. Hier seien ein leistungsfähiges Grafikprogramm sowie zahlreiche Beispiele zur Betriebssystemprogrammierung genannt.



Die Benutzung der verschiedenen Libraries und Intuition-Funktionen wird detailliert erklärt, so daß die Verwendung dieser Möglichkeiten in eigenen Programmen keine Schwierigkeiten bereitet.

Durch zahlreiche Anhänge – neben vielen Tabellen finden Sie auch ein sehr ausführliches Stichwortverzeichnis – wird das Buch optimal ergänzt und kann problemlos zum Nachschlagen von Details benutzt werden.

AUS DEM INHALT:

Erklärung der Schleifen- und Programmstrukturen

- ▶ FOR-NEXT, WHILE-WEND, REPEAT-UNTIL, DO-LOOP
- ▶ Prozeduren, Funktionsunterprogramme und Verzweigungen
- ▶ Rekursion
- ▶ Beispielprogramme

Variablentypen und Arrays

- ▶ numerische und Zeichenkettenvariablen
- ▶ Arrays zur Aufnahme großer Datenmengen

Multitasking in GFA-BASIC

- ▶ Reagieren auf Ereignisse
- ▶ Zeitabhängige Prozeduraufrufe

Programmentwicklung

- ▶ Programmplanung und -entwurf
- ▶ strukturierte Programmierung
- ▶ TOP-DOWN-Prinzip
- ▶ Fehlersuche
- ▶ Debugging-Möglichkeiten

Dateiverwaltung

- ▶ sequentielle Dateien
- ▶ Random-Access-Dateien
- ▶ Funktionen und Befehle zur Diskettenverwaltung

Grafikprogrammierung

- ▶ Die vielseitigen Möglichkeiten
- ▶ Turtlegrafik, Spriteprogrammierung
- ▶ Arbeiten mit mehreren Bildschirmen
- ▶ HAM-Modus
- ▶ Entwicklung eines vielseitigen Grafikprogramms

Sound- und Sprachprogrammierung

- ▶ Die Möglichkeiten der Befehle
- ▶ Verwendung der Anweisung in eigenen Programmen

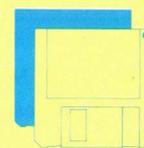
Abstrakte Datentypen

- ▶ Verkettete Listen
- ▶ Bäume

Betriebssystemprogrammierung

- ▶ Pulldownmenü-Steuerung
- ▶ Requester-Einsatz in Programmen
- ▶ Intuition und Graphics-Library
- ▶ Aufruf von Systemroutinen

Für Einsteiger ★ Fortgeschrittene ★ und Profis



ÜBER 500 Seiten
EINSCHLIESSLICH
PROGRAMMDISKETTE

Dem Buch liegt eine Programm-Diskette bei mit über 150 Übungs- und Beispielprogrammen

BUCH & DISKETTE
KOMPLETT **59,-**
HARDCOVER

Preise sind unverbindlich empfohlene Verkaufspreise

Heim Verlag

Heidelberger Landstr. 194
6100 Darmstadt-Eberstadt
Telefon 0 61 51-56057

BESTELLCOUPON

AMIGA GFA 3.0 BUCH

Bitte senden Sie mir _____ St. DAS GFA-BASIC 3.0 BUCH einschließlich Programm-Diskette für DM 59,-

zzgl. DM 5,- Versandkosten (unabhängig von bestellter Stückzahl)

per Nachnahme

Verrechnungsscheck liegt bei

Name, Vorname _____

Straße, Hausnr. _____

PLZ/Ort _____

Benutzen Sie auch die in KICKSTART vorhandene Bestellkarte.

b) Die Laufwerks-Port-Struktur

Offset	Größe	Beschreibung
00	—	herkömmliche Message-Port-Struktur
34	dc.b	Flagbits zur Verwaltung des Devices (Bit 0 = 1 => Device in Arbeit)
35	dc.b	Byte, um auf gerade Adresse zu kommen
36	dc.w	Zähler für Anzahl Tasks, die auf das Device zugreifen
38	dc.w	PRECOMP-Erhöhung, wenn Tracknummer größer ist als der hier angegebene Wert (normal 80)
40	dc.w	PRECOMP-Erhöhung, wenn Tracknummer größer ist als der hier angegebene Wert [wird normal nicht verwendet (\$FFFF)]
42	dc.w	PRECOMP-Erhöhung, wenn Tracknummer größer ist als der hier angegebene Wert [wird normal nicht verwendet (\$FFFF)]
44	dc.l	Wert für Wartezeit beim Steppen (normal 3000)
48	dc.l	Wert für Wartezeit, nachdem gewünschter Track erreicht wurde (normal 6000)
52	dc.b	Anzahl der erlaubten Fehler beim Diskzugriff (normal 10)
53	dc.b	Typ des angeschlossenen Laufwerkes (1 für 3.5"-Floppy)
54	dc.w	Anzahl der erreichbaren Tracks (normal 160)
56	dc.l	größter Offset (normal \$DC000 => 160 Tracks)
64	dc.b	Bits für Verwaltung: Bit 1 = 1 => keine Diskette im Laufwerk Bit 2 = 1 => erweiterter Befehl Bit 3 = 1 => Device schließen Bit 4 = 1 => Diskette ist schreibgeschützt
65	dc.b	motorbezogene Laufwerksbits für Drive-Select-Register
66	dc.b	Zähler für Anzahl Fehler beim Diskettenzugriff
67	dc.b	Laufwerksnummer
68	dc.l	Zeiger auf übergebene IO-Request-Struktur
72	dc.w	Sektornummer, die gelesen bzw. geschrieben werden soll
74	dc.w	Tracknummer des Kopfes
76	dc.w	Tracknummer des Kopfes
78	dc.l	Zeiger auf Load Puffer, in dem die codierten Daten stehen
82	dc.l	Zeiger auf Schreibpuffer mit codierten Daten
90	dc.l	Zeiger auf Datenpuffer, wenn die 16 Leerbytes im Block-Header mit decodiert werden sollen, sonst NULL
94	—	IORequest-Struktur, die zum Warten ans Timer-Device gesandt wird
134	—	IORequest-Struktur, die zum Warten bei der Prüfung auf Disk im Laufwerk ans Timer-Device gesandt wird
174	—	Port, an den Messages gesendet werden, wenn ein bestimmter Prozeß zuende ist
208	—	Message, die an den Port (Offset 174) gesandt wird, um Ende eines Prozesses anzuzeigen
228	—	Struktur für DiskBlockReady-Interrupt (Routine bei \$fea6f2)
250	—	Struktur für WordSync-Interrupt
272	—	Struktur für Index-Interrupt (Routine bei \$feb38e)
298	dc.w	Zähler für Diskettenwechsel

Um die Laufwerke im Multitasking-Betrieb verwalten zu können, wurde die Disk-Resource benutzt (siehe Routinen "anmelden" und "abmelden"). Hierbei handelt es sich wie bei einer Library um eine Ansammlung von Funktionen, die mit negativen Sprungadressen aufgerufen werden können. Hier eine Beschreibung der Funktionen:

Offset -6

Funktion: Bit für Laufwerk setzen
Parameter: D0 = Laufwerksnummer, die als vorhanden eingetragen wird
Rückgabe: D0 = \$FF Laufwerk war noch nicht vorhanden.
D0 = 0 Laufwerk war vorhanden.

Offset -12

Funktion: Bit für Laufwerk löschen
Parameter: D0 = Laufwerksnummer

Offset -18

Funktion: Laufwerk anmelden
Parameter: A1 = Zeiger auf einzufügende Message
Rückgabe: D0 <= 0 bedeutet, daß nicht auf Anmeldung gewartet werden muß.
Der Zugriff kann sofort beginnen.

Offset -24

Funktion: Laufwerk abmelden
Parameter: keine

Offset -30

Funktion: Testen, ob Laufwerk vorhanden
Parameter: D0 = Laufwerksnummer
Rückgabe: D0 = 0 Laufwerk vorhanden
D0 = \$FFFF Laufwerk nicht vorhanden

```

225: move.l a5, -(a7) ;A5 retten
226: move.l a1, a5 ;Zeiger auf QBLIT-Struktur
227: bsr setmodulo ;Blitter-Parameter setzen
228: move.b 22(a5), d0 ;Anzahl der Verschiebungs-
bits
229: moveq #12, d1 ;Verschiebungswert ins oberste
Nibble
230: lsl.w d1, d0 ;Wert nun in Bits 15-12
231: move.w #$5cc, 64(a0) ;BLTC0N0 beschreiben
232: ;keine Verschiebung für A
233: ;DMA-Bits für B und D
234: ;ABC+ABc+aBc+aBc als Minterms (D=B)
235: move.w d0, 66(a0) ;BLTC0N1 beschreiben
(Verschiebung von B)
236: movem.l 8(a5), d0/d1/a1 ;Werte aus Struktur
holen
237: move.l #shiftroutine2, 4(a5) ;Zeiger auf
nächste Funktion
238: tst.b 22(a5) ;Verschiebung notwendig?
239: beq keineverschiebung ;Nein, dann verzweige
240: sub.w.l #2, a1 ;Einschub am Ende unnötig
241: move.l #shiftroutine3, 4(a5) ;Zeiger auf
nächste Funktion
242: keineverschiebung:
243: move.l a1, 84(a0) ;Zeiger auf Destination D
244: move.w (a1), 24(a5) ;erstes Wort in QBLIT-
Struktur speichern
245: move.w 20(a5), 88(a0) ;BLT0SIZE beschreiben
(Blitter starten)
246: move.l (a7)+, a5 ;A5 zurückholen
247: rts ;Rücksprung
248:
249: shiftroutine3:
250: move.l 16(a1), a0 ;Zeiger auf Ziel in A0
251: move.w 24(a1), -2(a0) ;Zwischengespeichertes
Wort zurückschreiben
252: shiftroutine2:
253: moveq #0, d0 ;D0 löschen
254: move.l d0, 4(a1) ;Funktionszeiger löschen
255: move.l 26(a1), a1 ;Zeiger auf Laufwerks-
Port in A1
256: bsr sendmessage ;Message für Ende senden
257: moveq #0, d0 ;Flag für Ende
258: rts ;Rücksprung
259:
260: decodelongword:
261: ;Decodiert ein Langword aus zwei hintereinander
stehenden MFM-Langworten
262: ;A0 zeigt auf erstes MFM-Langword
263: ;D0 gibt decodiertes Langword zurück
264: move.l (a0)+, d0 ;Erstes Langword holen
265: move.l (a0)+, d1 ;Zweites Langword holen
266: and.l #$55555555, d0 ;Taktbits entfernen
267: and.l #$55555555, d1 ;Taktbits entfernen
268: lsl.l #1, d0 ;Bits richtigstellen
269: or.l d1, d0 ;und verknüpfen
270: rts ;Rücksprung
271:
272: getrightlongword:
273: ;Diese Routine schiebt ein codiertes Langword
richtig
274: ;A0 zeigt auf auf das Langword
275: ;D2 enthält Anzahl Bits für die Verschiebung
276: ;D0 gibt geschobenes Langword zurück
277: move.l (a0)+, d0 ;Langword holen
278: move.w (a0), d1 ;nachfolgendes Wort auch,
da maximale
;Verschiebung 15 Bit beträgt
279:
280: moveq #16, d3 ;max. Verschiebung + 1
281: sub.l d2, d3 ;Verschiebung für Langword
282: lsl.l d3, d0 ;Langword richtig verschieben
283: lsr.w d2, d1 ;Rest andersrum verschieben
284: or.w d1, d0 ;ergibt richtiges Langword
285: rts ;Rücksprung
286:
287: takttabelle:
288: dc.l $2244a244, $48912891, $52244a24, $54891289
289: dc.l $552244a2, $55489128, $5552244a, $55548912
290: datatabelle:
291: dc.l $91225122, $a4489448, $a9122512, $aa448944
292: dc.l $aa912251, $aaa44894, $aaa91225, $44894489

```

```

1: ;Listing 11
2: execbase = 4
3: putmsg = -366
4: getmsg = -372
5: findname = -276
6: wait = -318
7: qblit = -276
8: header = $ff150802
9:
10: run:
11: move.l execbase,a6 ;Zeiger auf Execbase
12: lea trdname,a1 ;Zeiger auf Device-Name
13: lea 350(a6),a0 ;Zeiger auf Device-List
14: jsr findname(a6) ;Device finden
15: tst.l d0 ;gefunden?
16: beq notfound ;Nein, ende
17: move.l 276(a6),a0 ;Zeiger auf eigenen Task
18: move.l d0,a6 ;Device-Struktur in A6
19: move.l 36(a6),a3 ;Zeiger auf Laufwerk-
;Port (df0:)
20: lea 174(a3),a1 ;Adresse des Reply-Ports
21: move.l 16(a1),-(a7) ;Zeiger auf Empfänger-
;Task retten
22: move.l a1,-(a7) ;Reply-Port retten
23: move.l a0,16(a1) ;eigenen Task als
;Empfänger eintragen
24: bset #0,34(a3) ;Trackdisk-Task auf
;Warteposition setzen
25:
26: move.l #track,d0 ;Tracknummer in D0
27: move.w d0,74(a3) ;in Laufwerks-Port
;eintragen
28: move.l 78(a3),a2 ;Zeiger auf Load Puffer
29: move.w d0,(a2) ;Tracknummer eintragen
30: bclr #0,2(a2) ;Bit 0 des nächsten
;Wortes löschen
31: clr.b 66(a3) ;Fehleranzahl auf Null
32: jsr readtrack ;Track lesen
33: clr.l d0 ;Wert für Motor aus
34: jsr motorschalten ;Motor ausschalten
35: move.b 3(a2),d0 ;erste Blocknummer in D0
36: cmp.b #11,d0 ;Blocknummer größer als 11?
37: bcc ende ;Ja, Fehler
38: move.b #11,d6 ;Sektoranzahl
39: clr.l d0 ;erster Block = Null
40: sub.b 3(a2),d0 ;erste Blocknummer abziehen
41: bpl nummerpositiv ;Ist Blocknummer
;positiv?
42: add.b #11,d0 ;Ja, dann positiv machen
43: nummerpositiv:
44: mulu #1088,d0 ;Anzahl Bytes pro Block
45: lea 1664(a2),a4 ;Zeiger auf Datenanfang
46: adda.l d0,a4 ;Zeiger auf Block 0
47: lea ziel,a5 ;Zeiger auf Ziel
48: clr.l d7 ;Anfang bei Sektor 0
49: blockloop:
50: lea 64(a4),a1 ;Zeiger auf Datenblock
51: move.l a5,a0 ;Ziel nach A0
52: move.l #512,d0 ;Anzahl zu decodierender Bytes
53: jsr decodeblock ;Daten eines Blocks
;decodieren
54: adda.l #512,a5 ;Zeiger auf Ziel erhöhen
55: sub.b #1,d6 ;Anzahl der Blöcke verringern
56: beq ende ;Alle 11 Sektoren
57: add.b #1,d7 ;Nächster Block
58: cmp.b 3(a2),d7 ;am Pufferanfang weitermachen?
59: bne normal ;Nein, normal weitermachen
60: lea 1664(a2),a4 ;sonst neuer Zeiger auf
;Pufferanfang
61: bra blockloop ;restlichen Blöcke decodieren
62: normal:
63: add.l #1088,a4 ;Zeigt auf nächsten Block
64: bra blockloop ;restlichen Blöcke decodieren
65: ende:
66: bclr #0,34(a3) ;Trackdisk-Task wieder
;freigeben
67: move.l (a7)+,a1 ;Reply-Port zurückholen
68: move.l (a7)+,16(a1) ;alten Empfängertask
;wieder eintragen
69: fehler:
70: rts ;Programmende

```

```

1: ;Listing 12
2: anmelden:
3: ;Diese Routine setzt den Task auf Wartestellung,
;bis das Laufwerk frei ist
4: movem.l a2/a4,-(a7) ;Register retten
5: move.l a6,a2 ;Zeiger auf Trackdisk-
;Device-Struktur
6: move.l 52(a2),a4 ;Zeiger auf ExecBase in A4
7: move.l 60(a2),a6 ;Zeiger auf Disk-Resource
8: anmeldung:
9: lea 208(a3),a1 ;Zeiger auf Message für
;Anmeldung
10: jsr -18(a6) ;Laufwerk anmelden
11: tst.l d0 ;muß noch gewartet werden?
12: bne nowait ;Nein, Zugriff kann sofort
;beginnen
13: exg a6,a4 ;Zeiger auf ExecBase in A6
14: getmessage:
15: move.l #1024,d0 ;Signalmaske für Wait-Funktion
16: jsr wait(a6) ;Warten, bis Signal kommt
17: lea 174(a3),a0 ;Zeiger auf Reply-Port
18: jsr getmsg(a6) ;Message holen
19: tst.l d0 ;Message angekommen?
20: beq getmessage ;Nein, auf Message warten
21: exg a6,a4 ;A4 und A6 wieder vertauschen
22: bra anmeldung ;Laufwerk nochmals anmelden
23: nowait:
24: exg a6,a2 ;Zeiger auf Trackdisk-Device-
;Struktur
25: move.b 65(a3),d1 ;Motorbits nach D1
26: or.b #$7f,d1 ;Bits 0-6 setzen
27: move.b d1,$bfd100 ;Leitungen auf 1 setzen,
;da Low-aktiv
28: lea 208(a3),a0 ;Zeiger auf Message für
;Anmeldung
29: cmp.l d0,a0 ;Messages gleich?
30: beq gleich ;Ja, dann verzweige
31: lea $dff000,a0 ;Zeiger auf Hardware-
;Register
32: move.w #$600,158(a0) ;ADKCON-Register
;(Sync sperren)
33: move.w #$9100,158(a0) ;MFM-Format / FAST-Mode
34: gleich:
35: movem.l (a7)+,a2/a4 ;Register zurückholen
36: rts ;Rücksprung
37:
38: abmelden:
39: ;Ein vorher angemeldetes Laufwerk wird abge-
;meldet, so daß ein anderer Task
40: ;auf das Laufwerk zugreifen kann.
41: move.l a6,-(a7) ;A6 retten
42: move.l 60(a6),a6 ;Zeiger auf Disk-Resource
43: jsr -24(a6) ;Laufwerk abmelden
44: move.l (a7)+,a6 ;A6 zurückholen
45: rts ;Rücksprung
46:
47: motorschalten:
48: ;Schaltet den Motor ein bzw. aus
49: ;D0 enthält 1 für Ausschalten/1 für Einschalten
50: move.l d2,-(a7) ;D2 retten
51: tst.l d0 ;Ein- oder Ausschalten?
52: seq d1 ;D0=0 => D1=$FF sonst D1=0
53: and.b #128,d1 ;Bits 0-6 löschen
54: cmp.b d1,d2 ;Stimmen Werte überein?
55: beq motormarke ;Ja, Motor hat bereits
;gewünschten Zustand
56: bclr #7,65(a3) ;Bit 7 der Motorbits
;löschen
57: or.b d1,65(a3) ;Bit aus D1 eintragen
58: bsr anmelden ;Laufwerk anmelden
59: moveq #255,d1 ;Leitungen von $BFD100
; sind Low-aktiv
60: eor.b d2,d1 ;gewünschte Bits selectieren
61: move.b d1,$bfd100 ;Leitungen in Hi-Zustand
62: move.b 65(a3),$bfd100 ;Es werden nur 1-0
;Sprünge registriert
63: bsr abmelden ;Laufwerk abmelden
64: btst #7,65(a3) ;Ein- oder Ausschalten?
65: bne motormarke ;Einschalten, also verzweige
66: move.l #500000,d0 ;Wert für Wartezeit
67: bsr warten ;warten, bis Motor ausgelaufen
68: motormarke:
69: moveq #0,d0 ;D0 löschen

```

```

70:  btst    #7,d2      ;ursprünglicher Zustand des
                        Motors
71:  seq     d0         ;Bit7(D2)=0 => D0=$FF sonst D0=0
72:  move.l  (a7)+,d2   ;D2 zurückholen
73:  rts                                     ;Rücksprung
74:
75:  warten:
76:  ;D0 enthält den Wert der Warteschleife
77:  cmp.l   #300,d0    ;Wert für Wartezeit testen
78:  blt    kleiner    ;Warteschleife, sonst zum
                        Timer-Device
79:  movem.l a0/a1,-(a7) ;Register retten
80:  lea    94(a3),a1   ;Zeiger auf IOREquest
                        für Timer-Device
81:  clr.l  32(a1)      ;Actual-Longwort löschen
82:  move.l d0,36(a1)   ;Länge eintragen
83:  move.l a6,-(a7)    ;A6 retten
84:  move.l execbase,a6 ;Zeiger auf ExecBase
85:  jsr   doio(a6)     ;Warten
86:  move.l (a7)+,a6    ;A6 zurückholen
87:  bra   gewartet    ;Fertig
88:  kleiner:
89:  tst.l  (a3)        ;Aktion innerhalb der
                        Warteschleife
90:  dbf   d0,kleiner  ;Warteschleife
91:  gewartet:
92:  rts               ;Rücksprung
93:
94:  movehead:
95:  ;Bewegt den Schreib- Lesekopf auf einen
      gewünschten Track. Die Tracknummer
96:  ;muß unbedingt gültig sein, da die Fehlerabfrage
      entfernt ist!
97:  ;D0 enthält die Tracknummer
98:  movem.l d2/d3,-(a7) ;Register retten
99:  move.w  76(a3),d2   ;Tracknummer in D2
100: ;Hier steht die Fehlerabfrage:
101: ; bpl   trackok
102: ; move.l d0,d2
103: ; bsr   error
104: ; tst.l d0           ;Tracknummer berichtigt?
105: ; bne   movefertig  ;Register zurückholen,
                        Routine verlassen
106: ; exg   d0,d2
107: ;trackok:
108: cmp.w  d0,d2        ;Tracknummern gleich?
109: beq   trackgleich  ;Ja, steppen nicht mehr
                        nötig
110: move.l d0,d3        ;gewünschter Track in D3
111: bsr   anmelden     ;Laufwerk anmelden
112: move.b 65(a3),d1    ;Motorbits nach D1
113: bset  #2,d1        ;untere Seite anwählen
114: btst  #0,d3        ;obere oder untere Seite?
115: beq   geradertrack ;gerade Tracknummer
                        (untere Seite)
116: bclr  #2,d1        ;obere Seite anwählen
117: geradertrack:
118: move.b d1,65(a3)    ;Motorbits in Laufwerks-
                        Port eintragen
119: move.b d1,$bfd100  ;Side-Select eintragen
120: move.w d3,76(a3)   ;Tracknummer eintragen
121: lsr.w  #1,d2        ;Cylindernummer
122: lsr.w  #1,d3        ;Cylindernummer
123: sub.w  d3,d2        ;An gewünschte Position
                        angekommen?
124: bcs   nachinnen   ;Nein, noch zu weit außen
125: bhi   nachausen   ;Nein, noch zu weit innen
126: bsr   abmelden    ;Ja, dann Laufwerk abmelden
127: bra   trackgleich ;und Routine verlassen
128: nachinnen:
129: bclr  #1,d1        ;Direction-Bit: nach innen
130: move.b d1,65(a3)   ;Motorbits in Laufwerks-Port
131: neg.w  d2          ;Cylindernummer positiv machen
132: bra   stepagain   ;Kopf um einen Track bewegen
133: nachausen:
134: bset  #1,d1        ;Direction-Bit: nach außen
135: move.b d1,65(a3)   ;Motorbits in Laufwerks-Port
136: bra   stepagain   ;Kopf um einen Track bewegen
137: step:
138: bsr   stepmotor   ;Kopf um einen Track bewegen
139: stepagain:
140: dbf   d2,step     ;bis gewünschte Position
                        erreicht ist

```

```

141:  bsr    abmelden   ;Laufwerk abmelden
142:  move.l 48(a3),d0  ;Wert für Wartezeit in D0
143:  bsr    warten     ;warten
144:  trackgleich:
145:  moveq  #0,d0      ;Flag für kein Fehler
146:  movem.l (a7)+,d2/d3 ;Register zurückholen
147:  rts               ;Rücksprung
148:
149:  stepmotor:
150:  move.b 65(a3),d0  ;Motorbits nach D3
151:  move.b d0,d1      ;Motorbits nach D1
152:  bclr  #0,d0       ;Wert für Kopf steppen
153:  move.b d0,$bfd100 ;Kopf steppen
154:  nop
155:  nop               ;kurze Zeit warten
156:  move.b d1,$bfd100 ;Leitungen auf 1, da
                        Low-aktiv
157:  bsr    abmelden   ;Laufwerk abmelden
158:  move.l 44(a3),d0  ;Wartezeit zwischen
                        zwei steps
159:  bsr    warten     ;warten
160:  bsr    anmelden   ;Laufwerk wieder anmelden
161:  move.b 65(a3),$bfd100 ;Drive-Select-Register
                        beschreiben
162:  rts               ;Rücksprung

```

```

1:  ;Listing 13
2:  syncwort = $4489 ;Hier das gewünschte
                        Sync-Wort eintragen
3:  run:
4:  lea   takttable,a0 ;Zeiger auf Tabelle für
                        ungerade Verschiebung
5:  lea   datatable,a1 ;Zeiger auf Tabelle für
                        gerade Verschiebung
6:  lea   synccode1,a2 ;Erster Puffer
7:  lea   synccode2,a3 ;Zweiter Puffer
8:  move.w #$aaaa,(a2) ;Nullbytes eintragen
9:  move.w #$aaaa,(a3) ;Nullbytes eintragen
10: move.w #syncwort,2(a2)
11: move.w #syncwort,4(a2)
12: move.w #syncwort,6(a2)
13: move.w #syncwort,2(a3)
14: move.l 4(a2),4(a3) ;Sync-Wort eintragen
15: moveq #7,d0        ;Acht mögliche Verschiebungen
16: moveq #1,d1        ;Erste ungerade Verschiebung
                        um ein Bit
17: loop:
18: move.l (a2),d7     ;Wert aus Puffer
19: lsr.l  d1,d7       ;Wert verschieben
20: move.l d7,(a2)    ;Wert zurückschreiben
21: move.l 4(a2),d7
22: lsr.l  d1,d7
23: move.l d7,4(a2)
24: cmp.b  #2,d1      ;ab dem zweiten Durchlauf
                        konstante Verschiebung
25: beq   marke
26: addq.b #1,d1      ;sonst Verschiebung auf zwei
27: marke:
28: move.l (a3),d7
29: lsr.l  d1,d7
30: move.l d7,(a3)
31: move.l 4(a3),d7
32: lsr.l  d1,d7
33: move.l d7,4(a3)
34: move.w 2(a2),(a0)+ ;Verschobenen Wert in
                        Takttable eintragen
35: move.w 6(a2),(a0)+
36: move.w 2(a3),(a1)+ ;Verschobenen Wert in
                        Datatable eintragen
37: move.w 6(a3),(a1)+
38: dbf   d0,loop     ;das Ganze acht mal
39: move.w #syncwort,-4(a1) ;vorletztes Wort
                        korrigieren
40: rts               ;Fertig
41:
42: synccode1: dc.w 0,0,0,0
43: synccode2: dc.w 0,0,0,0
44: takttable: dc.l 0,0,0,0,0,0,0,0
45: datatable: dc.l 0,0,0,0,0,0,0,0

```

PROGRAMMIEREN AUF DEM AMIGA

Das richtige Buch für
Auf- und Umsteiger



VON
ERNST HEINZ

PROGRAMMIEREN AUF DEM AMIGA

WICHTIGE MERKMALE

- ▶ Dieses Buch weist Programmierern den Weg, wie Sie die fantastischen Fähigkeiten Ihres AMIGA auch von **AmigaBASIC** aus nutzen können. Es schließt die in den Handbüchern entstandenen Lücken, indem es die offengebliebenen Fragen in anschaulicher Weise beantwortet.
- ▶ Damit ist dieses Buch insbesondere für lernwillige Aufsteiger, d. h. mehr oder weniger erfahrene BASIC-Programmierer, sowie für Umsteiger, die bisher mit anderen Computersystemen gearbeitet haben, konzipiert.
- ▶ Besonderer Wert wurde auf guten Programmierstil gelegt. Anhand von über 40 sehr ausführlich dokumentierten Übungs- u. Beispielprogrammen kann der Leser trainieren, fremde Programme zu lesen und zu verstehen und richtige und gute Programmierung erlernen.
- ▶ Alle offenen Fragen und Probleme in Bezug auf AMIGA-spezifische Programmier-techniken und -befehle wird Ihnen dieses Buch zu beantworten versuchen.
- ▶ Besonders werden folgende Hauptthemen im Buch behandelt:
Fenster-Technik · Menue-Technik · Unterbrechung-Technik · Animation in Amiga-Basic · Grafik-Befehle · Normale Maussteuerung
- ▶ Zum Buch gibt es eine Programm-Diskette mit allen aufgeführten Übungs- und Beispielprogrammen

AUS DEM INHALT

- ▶ CLI und Möglichkeiten der **Execute**-Programmierung
- ▶ Erstellen einer eigenen **Startup**-Diskette
- ▶ Spezielle AmigaBASIC-Arbeitsdiskette erstellen
- ▶ Grafikprogrammierung (ohne Animation)
- ▶ Sound- und Sprachprogrammierung (Erzeugung menschlicher Sprache/Erzeugung von Tönen und mehrstimmigen Melodien/Tonerzeugung gemäß musikalischer Notennotierung)
- ▶ **Unterbrechungsfähigkeiten** von AmigaBASIC
- ▶ **Maussteuerung**
- ▶ **Menueprogrammierung** von Pull-Down-Menüs
- ▶ **Fenster- und Bildschirmtechnik**
- ▶ **Grafische Animation** mit **Sprites** und **Bobs**
- ▶ Ein **Potpurri** von AmigaBASIC-Programmen
- ▶ Kleine Einführung zur **Spezialhardware** des AMIGA

über 300 Seiten **DM 49,-***

PROGRAMMDISKETTE

AUS DEM INHALT

- ▶ Zur Unterstützung der praktischen Übung und Arbeit am Computer gibt es die Programmdiskette.
- ▶ An über 40 Übungs- und Beispielprogrammen können Sie guten Programmierstil nachvollziehen und trainieren.
- ▶ Damit hat die lästige Tipparbeit ein Ende; Diskette laden und los geht's...
- ▶ Die Diskette ist beim Verlag erhältlich und kostet nur **DM 29,-***

* Preise sind unverbindlich empfohlene Verkaufspreise

BESTELL-COUPON

an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Ich bestelle: _____ St. *Programmieren auf dem AMIGA* á DM 49,—
_____ St. *Programmdiskette* zum Buch á DM 29,—

zzgl. DM 5,- Versandkosten (unabhängig von bestellter Stückzahl)
 per Nachnahme Verrechnungsscheck liegt bei

Name, Vorname _____

Straße, Hausnr. _____

PLZ, Ort _____

Benutzen Sie auch die in KICKSTART vorhandene Bestellkarte.

HeimVerlag

Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 061 51-5 60 57

```

1: ;Listing 14
2: writetrack:
3: ;Routine schreibt Daten vom Speicher auf Disk
4: ;A0 zeigt auf Schreibpuffer, der die codierten
   und formatierten Daten
5: ;enthält
6: ;D0 enthält die Anzahl zu schreibender Bytes
7: move.l #setdsklen,a4 ;Zeiger auf DSKLEN-
   Routine
8: movem.l a4/a2/d2,-(a7) ;Register retten
9: move.l a0,a2 ;Zeiger auf Schreibpuffer
   nach A2
10: move.l d0,d2 ;Anzahl zu schreibender Bytes
11: bsr anmelden ;Laufwerk anmelden
12: move.b 65(a3),$bfd100 ;Motorbits setzen
13: move.w #$4000,$dff024 ;Disk-DMA sperren
14: move.l #1000,d0 ;Wert für Wartezeit
15: bsr warten ;warten (warum, weiß
   ich nicht!!!)
16: lea $dff000,a1 ;Zeiger auf Hardware-
   Register
17: move.l a2,32(a1) ;Zeiger auf Disk-Daten
   (DSKPT)
18: move.w #$1002,156(a1) ;DiskSync und
   DiskBlockReady Interrupts
19: ;sperren
20: move.w #$8002,154(a1) ;DiskBlockReady
   Interrupt erlauben
21: btst #2,$bfe001 ;Diskette im Laufwerk?
22: bne diskok ;verzweige, wenn OK
23: nodiskin:
24: moveq #29,d2 ;Fehlernummer in D0
25: bra writefailed ;Routine verlassen
26: diskok:
27: btst #3,$bfe001 ;Disk schreibgeschützt?
28: beq writeprotect ;verzweige, wenn
   schreibgeschützt
29: move.l execbase,a0 ;Zeiger auf ExecBase in A0
30: move.w #$4000,154(a1) ;Interrupts sperren
31: addq.b #1,294(a0) ;Disable-Funktion
   ausführen
32: btst #4,34(a6) ;Bit 4 in Statusbyte
   gesetzt?
33: beq bit4clear ;verzweige, wenn gelöscht
34: moveq #36,d2 ;sonst Fehlernummer in D2
35: move.l execbase,a0 ;Zeiger auf ExecBase in A0
36: subq.b #1,294(a0) ;Enable-Funktion ausführen
37: bge intfehler ;Interrupts trotz Enable
   noch gesperrt
38: move.w #$c000,154(a1) ;Interrupts erlauben
39: intfehler:
40: bra writefailed ;Routine verlassen
41: bit4clear:
42: bset #5,34(a6) ;Bit 5 in Statusbyte
   setzen
43: move.l execbase,a0 ;Zeiger auf ExecBase in A0
44: subq.b #1,294(a0) ;Enable-Funktion ausführen
45: bge nextintfehler ;Interrupts trotz
   Enable noch gesperrt
46: move.w #$c000,154(a1) ;Interrupts erlauben
47: nextintfehler:
48: move.w #$6000,158(a1) ;PRECOMP-Zeit auf Null
   setzen
49: move.w 76(a3),d0 ;Tracknummer in D0
50: move.w #$8000,d1 ;PRECOMP-Zeit Null
51: cmp.w 38(a3),d0 ;anderer PRECOMP-Wert
   erforderlich?
52: bls precompok ;Nein
53: move.w #$a000,d1 ;sonst 140 Nanosekunden
54: precompok:
55: move.w d1,158(a1) ;Schreibdichte eintragen
56: lsr.w #1,d2 ;Anzahl der zu schrei-
   benden Worte in D2
57: ori.w #$c000,d2 ;Bit für Schreiben setzen
58: move.l d2,d0 ;Wert nach D0
59: jsr (a4) ;DSKLEN beschreiben
   (Track schreiben)
60: bsr waitmessage ;Warten, bis Track
   geschrieben ist
61: lea $dff000,a1 ;Zeiger auf Hardware-
   Register
62: move.w #2,154(a1) ;DiskBlockReady Interrupt
   sperren

```

```

63: move.l #2000,d0 ;Wert für Wartezeit
64: bsr warten ;warten
65: move.w #$4000,36(a1) ;Disk-DMA sperren
66: bclr #5,34(a6) ;Bit 5 des Statusbyte löschen
67: btst #2,$bfe001 ;Diskette im Laufwerk?
68: beq nodiskin ;Nein, dann verzweigen
69: moveq #0,d2 ;Flag für kein Fehler
70: writefailed:
71: bsr abmelden ;Laufwerk abmelden
72: move.l d2,d0 ;Rückmeldung nach D0
73: movem.l (a7)+,a4/a2/d2 ;Register zurückholen
74: rts ;Rücksprung
75: writeprotect:
76: moveq #28,d2 ;Fehlermeldung in D2
77: bra writefailed ;Routine verlassen

```

```

1: ;Listing 15
2: execbase = 4
3: findname = -276
4: track = 20
5: adresse = $50000
6:
7: run:
8: move.l execbase,a6 ;Zeiger auf Execbase
9: lea trdname,a1 ;Zeiger auf Device-Name
10: lea 350(a6),a0 ;Zeiger auf Device-List
11: jsr findname(a6) ;Device finden
12: tst.l d0 ;gefunden?
13: beq notfound ;Nein, ende
14: move.l 276(a6),a0 ;Zeiger auf eigenen Task
15: move.l d0,a6 ;Device-Struktur in A6
16: move.l 36(a6),a3 ;Zeiger auf Laufwerk-Port
   (df0:)
17: lea 174(a3),a1 ;Adresse des Reply-Ports
18: move.l 16(a1),-(a7) ;Zeiger auf Empfänger-
   Task retten
19: move.l a1,-(a7) ;Reply-Port retten
20: move.l a0,16(a1) ;eigenen Task als
   Empfänger eintragen
21: bset #0,34(a3) ;Trackdisk-Task auf
   Warteposition setzen
22:
23: move.l #1,d0 ;Wert für Motor ein
24: jsr motorschalten ;Motor einschalten
25: move.l #track,d2 ;Tracknummer nach D2
26: move.l d2,d0 ;Tracknummer nach D0
27: jsr movehead ;Kopf positionieren
28: lea adresse,a5 ;Zeiger auf Startadresse
   der zu schreibenden
   Daten
29:
30: move.l 82(a3),a2 ;Zeiger auf Schreibpuffer
31: lea 4(a2),a2
32: move.w #$fff,d0 ;Zählwert
33: move.l #$aaaaaaa,d1 ;Wert für Löschen
34: loesche:
35: move.l d1,(a2)+ ;aktuelles Langwort löschen
36: dbra d0,loesche ;den ganzen Schreibpuffer
37: move.l 82(a3),a2 ;Zeiger auf Schreibpuffer
38: lea 1664(a2),a2 ;Zeiger auf Anfang der
   Daten
39: moveq #11,d4 ;Anzahl der Blöcke
40: moveq #0,d5 ;D5 als Blockzähler
41: makeblock:
42: move.l #$ff000000,d0 ;Format-Kennung des
   Headers
43: move.l d5,d1 ;Blocknummer nach D1
44: lsl.l #8,d1 ;Nummer auf richtige
   Position schieben
45: or.l d1,d0 ;in Header eintragen
46: or.l d4,d0 ;Anzahl der Blöcke bis zur
   Lücke eintragen
47: move.l d2,d1 ;Tracknummer nach D1
48: swap d1 ;Nummer in richtige Position
   bringen
49: or.l d1,d0 ;und in Header eintragen
50: move.l a2,a1 ;Schreibpuffer nach A1
51: move.l a5,a0 ;Zeiger auf Daten nach A0

```

```

52: jsr      blockcodierung ;Daten codieren und in
        Schreibpuffer schreiben
53: addq.l  #1,d5           ;Blockzähler erhöhen
54: adda.l  #1088,a2       ;Zeiger auf nächsten Block
        im Schreibpuffer
55: adda.l  #512,a5       ;Zeiger auf nächsten Datenteil
56: subq.l  #1,d4         ;Anzahl der Blöcke verringern
57: bne     makeblock     ;den ganzen Track durch
58: move.l  #82(a3),a0    ;Zeiger auf Schreibpuffer
59: lea     4(a0),a0
60: move.w  #13630,d0     ;Anzahl zu schreibender Bytes
61: jsr     writetrack    ;Track schreiben
62: moveq   #0,d0        ;Wert für Motor aus

```

```

63: jsr     motorschalten ;Motor ausschalten
64:
65: ende:
66: bclr   #0,34(a3)     ;Trackdisk-Task wieder
        freigeben
67: move.l (a7)+,a1     ;Reply-Port zurückholen
68: move.l (a7)+,16(a1) ;alten Empfängertask
        wieder eintragen
69: fehler:
70: rts                ;Programmende
71:
72: name: dc.b "trackdisk.device",0

```

Haben Sie eine gute Programm-
idee und wollen ein Buch
schreiben und mitgestalten.
Kennen Sie eine Menge
Tips und Tricks.
Möchten Sie Ihre
Erfahrungen
weitergeben.

Wir bieten Ihnen unsere Erfah-
rung und unterstützen Ihre
Ideen. Als leistungsstarker
Verlag freuen wir uns
bald von Ihnen zu
hören.

Wir suchen noch Autoren wie Sie.

Heim Verlag Kennwort: Autor Heidelberg Landstr. 194 6100 Da.-Eberstadt Tel.: 06151/56057

Commodore®-Ersatzteil-Service

- » Sie wollen Ihren Computer selbst reparieren, um Kosten und Zeit zu sparen ?
- » Sie suchen schon lange ein bestimmtes Ersatzteil, das es nicht "an jeder Ecke" gibt ?
- » Oder sind Sie selbst Händler oder bieten Reparaturen an und wollen nach Möglichkeit alles aus einer Hand möglichst günstig einkaufen ?

Wir liefern prompt und zuverlässig und beraten Sie gern, auch in schwierigen Fällen.

☎ Rufen Sie uns an: (02333-80202)

Von 8:00 bis 17:00 sind wir persönlich für Sie da. Nachts und am Wochenende zeichnet ein Tonband Ihre Wünsche auf.

↳ Oder schreiben Sie uns :



CIK-Computertechnik · Ingo Klepsch
Postfach 13 31 5828 Ennepetal 1
Tel. 0 23 33/ 8 02 02 Fax 0 23 33/ 7 03 45

3,5" NN DISKS 2 DD

100 Stück	DM 199,95
50 Stück	DM 104,95
10 Stück	DM 21,95
1 Stück	DM 2,25

Panorama Kickstart, Auge, Fish, RFD, Fough	Publik Domain
	1 Stk DM 4,50
	10 Stk DM 40,-
	50 Stk DM 187,50
	100 Stk DM 350,-

Amiga Publik Domain Handbuch
1-3 DM 49,-

video
LOFT

HARD & SOFT
ware GrnbH

Fiedlerstr. 22 - 32
D - 3500 Kassel
tel.: 0561 - 87 33 99 /
87 79 28

AMIGA-BUREAU

ein Programm nach Ihren Wünschen



EDOTRONIK®

D-8000 München 80, St.-Veit-Straße 70, ☎ 0 89 / 40 40 93

Mein Professor für theoretische Informatik glaubte, zum Thema der Schrifterkennung folgenden Satz verlautbaren zu müssen: "Schriftmustererkennung hat nur deshalb wahnsinnige Ähnlichkeit mit 'Desktop Publishing', weil sie genau umgekehrt und doch ganz anders funktioniert." Ganz ehrlich - ich habe den Sinn dieses Satzes nie so recht verstanden, obwohl mir die Bedeutung all dieser Worte geläufig ist, und vielleicht deshalb steht der Satz noch heute in meinem Vorlesungsskriptum.

Schrifterkennung theoretisch

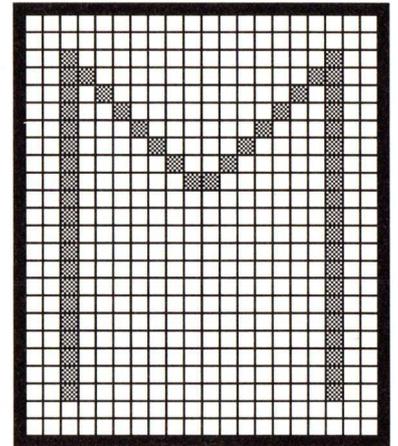
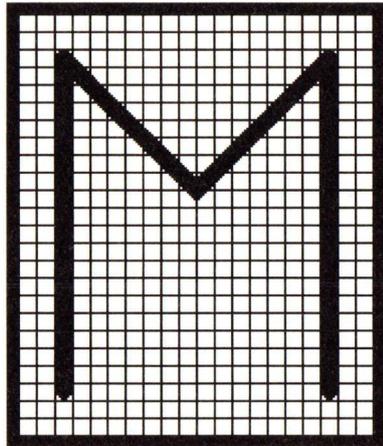


Bild 1: Das Vergleichsprinzip

Erst als dann verschiedene Wissenschaftssendungen von revolutionären Neuerungen in der Computertechnik zu berichten wußten und Worte wie 'Künstliche Intelligenz' und 'Neuro-Computer' die Runde machten, erst dann fiel mir dieser Satz wieder ein. Da gibt es Computer, die 'sehen' können, und bald kann ich mich mit dem Computer richtig unterhalten! Ganz unbestritten gehört die Schrifterkennung in den Zweig der 'Artifiziellen Intelligenz' (Abk.: "AI") - übrigens: von 'künstlicher Intelligenz' spricht man in deutschsprachigen Insiderkreisen längst nicht mehr (wegen der vielen Verwirrungen).

In der Computerbibel "Gödel, Escher, Bach" von Douglas R. Hofstadter ist auf Seite 641 eine übersichtliche Liste aller Disziplinen der AI aufgeführt. Hofstadter vermeidet zwar Hinweise auf sogenannte 'Expertensysteme' (auch wieder eine deutsche Wortschöpfung), räumt aber der visuellen AI viel Platz ein. Demnach gehört das "Erkennen indivi-

dueller handgeschriebener Zeichen" und das "Lesen von Texten in verschiedenen Schriftarten" in diesen Wissenschaftszweig und wird der Überschrift 'Mustererkennung' zugeordnet.

Was ist eine 'Schrift-(Muster)erkennung'?

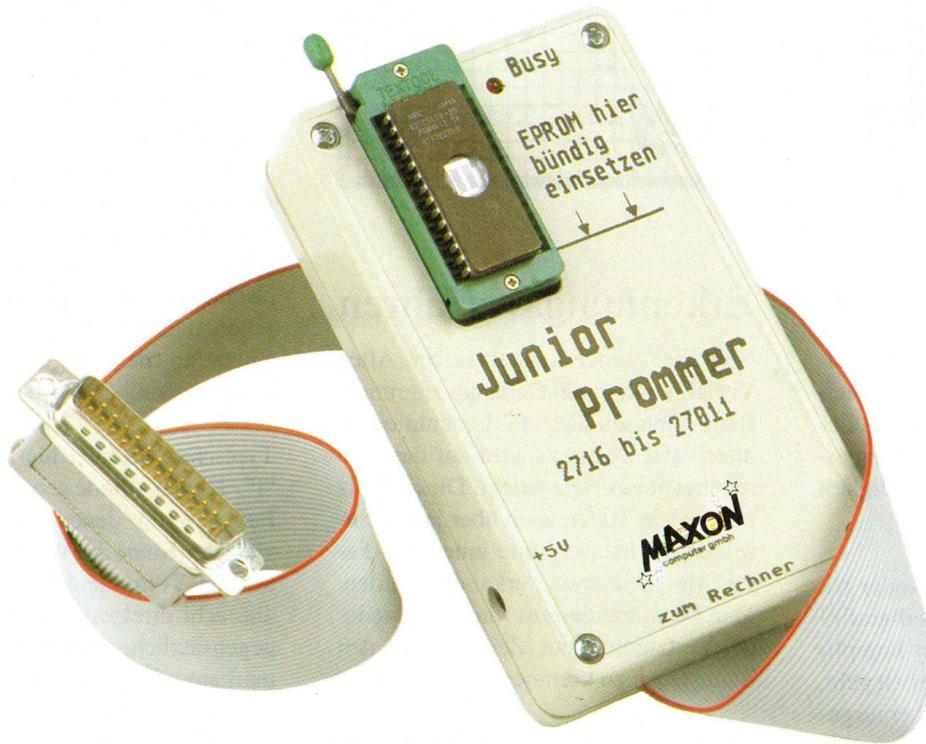
Es geht im Grunde nur darum, irgendwelche Grafik in computerverständliche Zeichen umzuwandeln. Daß es sich bei dieser 'Grafik' um Buchstaben und Texte handelt, kann der Computer noch nicht wissen. Durch ein spezielles Lesegerät (Scanner) oder per Videokamera werden Hell-/Dunkel-Impulse analog abgetastet, digital in den Rechner transportiert und dort in ASCII-Zeichen umgewandelt. Gerade dieser Umwandlungsprozeß aber hat es in sich!

Problem 1: Solange bei der Analog/Digital-Wandlung im Scanner nur zwischen Ganzhell (Weiß) und Ganzdunkel (Schwarz) unterschieden werden muß,

ist das Ganze kaum der Rede wert. Aber wenn mit mehreren Graustufen gearbeitet wird, oder sogar zwischen Farben unterschieden werden soll, muß der Scanner schon einiges leisten. Deswegen wird ein sogenannter Pendelwert vom Scanner berücksichtigt, der helleren Werten Weiß zuordnet und eher dunkleren Werten Schwarz.

Problem 2 - Die Auflösung: Die Anzahl der Lesepunkte einer Scannermatrix bestimmt die Genauigkeit des Lesevorganges. So waren bislang 200 dpi und 300 dpi recht grob. "dpi" = "dots per inch", zu deutsch: Bildpunkte pro Zoll. Gerade an den Grenzen zweier Buchstaben können dabei Verwaschungen oder Auslassungen auftreten. Das ist insbesondere dann der Fall, wenn diese Buchstaben zwischen zwei Lesepunkten lagen. Bei einem DIN-A4-Blatt lesen etwa 3500 solcher Matrixpunkte eine Scannerzeile ein (nicht verwechseln mit einer Textzeile). Geräte mit 400 dpi und 600 dpi erreichen gerade preislich interessante Tiefen.

Klein, kompakt und leistungsstark- der JuniorPrommer



Der JuniorPrommer programmiert alle gängigen EPROM-Typen, angefangen vom 2716 (2 KByte) bis zum modernen 27011 (1 MBit). Aber nicht nur EPROMS, sondern auch einige ROM- und EEPROM-Typen lassen sich lesen bzw. programmieren.

Zum Betrieb benötigt der JuniorPrommer nur +5V, die am Joystickport Ihres Amiga abgenommen werden; alle anderen Spannungen erzeugt die Elektronik des JuniorPrommers. Die sehr komfortable Software, natürlich Menü-unterstützt, erlaubt alle nur denkbaren-Manipulationen.

Fünf Programmieralgorithmen sorgen bei jedem EPROM-Typ für hohe Datensicherheit. Im eingebauten Hex-/ASCII-Monitor läßt sich der Inhalt eines EPROMS blitzschnell durchsuchen und auch ändern.

Alles dabei!

Bemerkenswert ist der Lieferumfang, so wird z.B. das Fertigerät komplett aufgebaut und geprüft im Gehäuse mit allen Kabeln anschlussfertig geliefert.

Auf der Diskette mit der Treiber-
software befindet sich noch ein Programm, das die Kickstart-Diskette

in vier EPROM-Dateien für 27512 (64 KByte EPROM) zerlegt, die dann mit dem JuniorPrommer gebrannt werden können. Ferner wird der Source-Code für die Lese- bzw. Programmerroutinen mitgeliefert. Und last but not least ist im Bedienungshandbuch (deutsch) der Schaltplan abgedruckt.



Bestellcoupon MAXON Computer GmbH Industriestraße 26 6236 Eschborn Tel.: 06196/481811

Hiermit bestelle ich:

- Fertigerät wie oben beschrieben DM 249,00
- Leerplatine (o. Bauteile) und Software DM 59,00
- Leergehäuse (gebohrt und bedruckt) DM 39,90

Versandkosten: Inland DM 7,50
Ausland DM 10,00

Auslandsbestellungen nur gegen Vorkasse
Nachnahme zuzüglich DM 3,80 Nachnahmege-
bühr

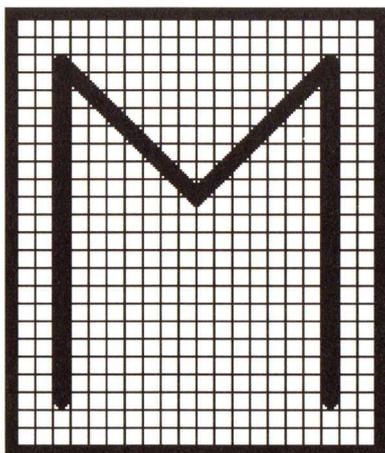
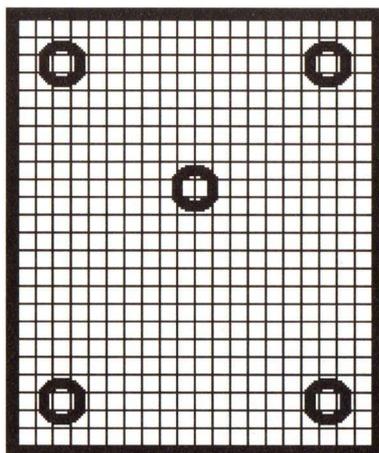


Bild 2: Das Markante-Punkte-Prinzip

Problem 3 - Die Vorlage: Wenn der Kontrast zwischen Papier und Schrift nicht sehr hoch ist, kann es beim Pendelwert oft zu 'entweder-oder-Entscheidungen' kommen und das für jeden Lesepunkt einzeln. Eine Verbesserung erreicht man, wenn die Scannerhelligkeit größer gewählt wird.

Gehen wir davon aus, der Scanner hat uns eine schöne 'Schwarzweißgrafik' geliefert. Bevor nun ein recht komplizierter Programmteil an die Arbeit geht, werden markante Grafikteile herausgesucht. Ganz wichtig sind durchgehende Linien. Weiße Linien, die 'unbeschadet' horizontal den Text durchlaufen, werden automatisch als Zeilenzwischenräume erkannt. Einige Programme prüfen auch in vertikaler Richtung und schließen dabei auf Block- bzw. Spaltengrenzen. Dann werden schwarze Linien in beiderlei Richtung aufgespürt und 'unsichtbar' gemacht. Dabei muß das Programm gewisse Toleranzen berücksichtigen. Vertikale Linien beispielsweise, die so groß sind wie der Abstand der Zeilenzwischenräume oder weniger, dürfen nicht unterdrückt werden, es könnten ja Buchstaben sein. Horizontale schwarze Linien dürfen in aller Regel unbedenklich getilgt werden.

Verschiedene Programme prüfen besonders bei schwarzen Linien die unmittelbare Nachbarschaft ab, um sicher zu gehen, daß es sich wirklich um grafische Teile handelt. Durch diese Linienauslese werden insbesondere Bilder und Fotos ausgeschaltet. Einige Prüfroutinen stellen dabei sicher, daß nicht zufällig doch einige Buchstaben unter den Tisch fallen.



Erkennungsverfahren

Nun aber zum Erkennen selber. Allen Verfahren liegt ein Gedanke zugrunde: Beim Einlesen über die Leuchtdiodenmatrix des Scanners wird auf das Bild regelrecht ein Netz gelegt. Dieses Netz legt sich in Teilen auch über jeden einzelnen Buchstaben, und innerhalb dieser Maschen kann 'erkannt' werden. Man kennt mindestens fünf verschiedene Verfahren, dem Zeichen auf den Buchstaben zu kommen.

Allereinfachst ist das **Vergleichsprinzip** bzw. der **'ASCII-Vergleich'** (Bild 1). Da muß vorher schon festgelegt sein, in welcher Schriftart und oftmals auch in welcher Größe der Text eingelesen wurde. In einer Bibliothek prüft das Programm eine streng näherungsweise Ähnlichkeit und nimmt bei Eindeutigkeit der Suche einen Treffer an. So haben früher die allerersten Programme gearbeitet. Das Verfahren ist unflexibel und nimmt keine Abweichungen hin.

Etwas komplizierter (Bild 2) wird die Suche nach den **'markanten Punkten'**. So werden für einen bestimmten Buchstaben Punkte festgelegt, in denen er in höherem Maße 'ausgeprägt' ist. Ausprägung heißt nichts anderes, als daß genau dort ein Kontrastunterschied, also eine gegenteilige Farbe gegenüber den direkten Nachbarpunkten vorliegen muß. Das Maß der Ausprägung bedeutet, daß in dem markanten Punkt ein Kontrastunterschied zu mehr als der Hälfte der Nachbarpunkte vorliegt. Je weniger unmittelbare Nachbarpunkte dieselbe Farbe haben, desto höher das Maß der Ausprägung und desto markanter der Punkt. Nachbarpunkte sind auch jene Maschenquadrate, die nur mit einer Eckspitze an den fraglichen markanten Punkt stoßen.

Typisches Beispiel hierzu: Das kleine "i". Es hat mindestens drei markante Punkte, von denen der i-Punkt (im wahrsten Sinne des Wortes) der markanteste ist. Vom i-Punkt aus gesehen haben alle acht direkten Nachbarpunkte eine gegensätzliche Farbe, also hat dieser markante Punkt das höchstmögliche Maß der Ausprägung. Die Wahl der markanten Punkte unterliegt wichtigen Gesetzmäßigkeiten. So nimmt die Treffergenauigkeit ab, wenn zu wenige markante Punkte definiert wurden (logisch), kurioserweise ebenso, wenn es zu viele sind. Auch ist die Lage, Entfernung und das Verhältnis der markanten Punkte zueinander zu beachten. Typische Punkte hoher Ausprägung sind Eckpunkte. Punkte innerhalb von Linien eignen sich nicht als markante Punkte.

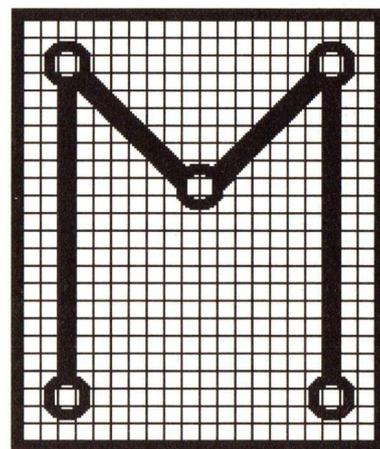
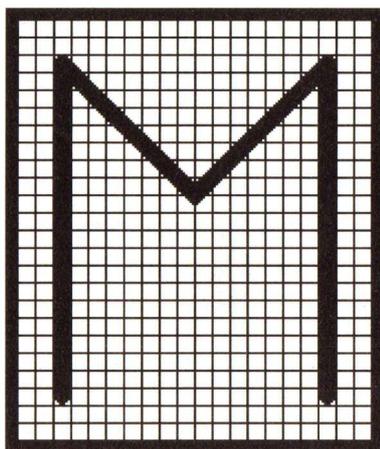


Bild 3: Das Drahtmodell-Prinzip

Ziemlich schnell hat man aus dem Prinzip der markanten Punkte das **'Drahtmodell'** entwickelt (Bild 3), aber nicht notwendigerweise, um dieses dadurch ablösen zu lassen. Dort wurde ganz einfach die Verbindung zwischen den Punkten sichtbar gemacht. Es genügt demgegenüber auch völlig, im markanten Punkt die Richtung des Linienverlaufes zu vermerken (Vektormodell), also die Linie selbst nicht zu ziehen. Vorteil dieser Verfeinerung: Es werden immer weniger markante Punkte zur Eindeutigkeit nötig.

Als Fortentwicklung dieses Modells zeigt sich das Einbeziehen typischer Muster wie Öffnungen auf bestimmten Seiten, bestimmte Winkelverhältnisse, Parallelitäten, Kreuzungen, Überlagerungen und ähnliches mehr.

Verfahren Nummer vier nennt man **'Eingrenzung'** und sagt nichts anderes, als daß man für ein bestimmtes Maschennetz eindeutig festgelegt hat, wo eine gegenteilige Farbe sein darf und wo nicht. Man nennt diese Eingrenzung auch noch 'Prinzip der verbotenen Zone' (Bild 4). So kann man dreierlei 'Zustände' festschreiben und zwar: 1., wo ein gegenteiliger Maschenfarbpunkt auf jeden Fall sein muß, 2., wo er möglicherweise sein kann und 3., wo er auf jeden Fall nicht sein darf. Je strenger man diese Bedingungen für die 3 Zustände nimmt, desto genauer wird die Trefferquote.

Eine Abwandlung ist das stark vereinfachte **'Maskenvergleichsverfahren'** (Bild 5). Da verfährt man mit den 3 obigen Zuständen folgendermaßen: 1. Man

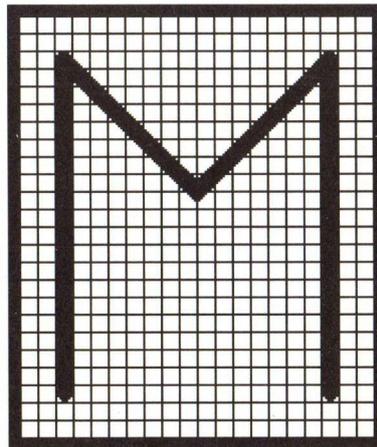


Bild 4: Das Eingrenzungsprinzip

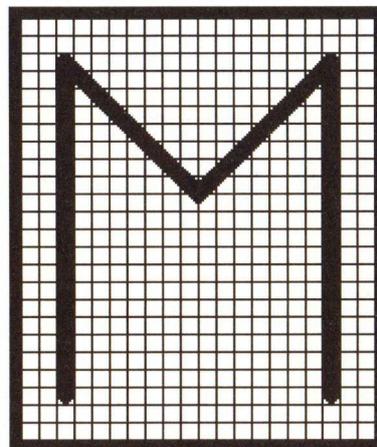
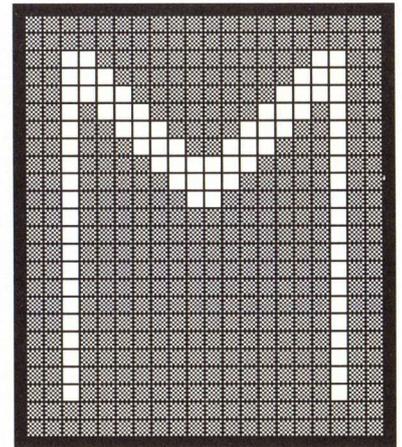


Bild 5: Das Maskenvergleichsverfahren

nimmt einen Treffer an, wenn das Zeichen die Maske überdeckt, und beachtet die Grenzbereiche nicht. 2. Man nimmt einen Treffer an, wenn die Maske überdeckt und keine Farbgegensätze außerhalb der Maske sind. Hauptnachteil: Es müssen auch hier die Zeichensätze an-



nähernd bekannt sein. Hauptvorteil: extrem schnell.

Zusätzlich können noch Programmteile in Aktion treten, die gewisse grammatikalische und orthografische Eigenheiten berücksichtigen. ■

über 50

TEXTURES

professionel mit studioequipment erstellt

endlich sind sie in der lage realistische bilder und traumhaft schöne animationen zu erstellen ein muss für alle grafiker und computeranimateure

FE GRAFIK + VIDEO

klosterkirchhof 18-20
2300 kiel 1
tel 0431/978989
fax 0431/97279

preis incl. mwst 69,-DM

Profi-Auflösung für alle Amiga !

Die HIGH RESOLUTION WORKBENCH ermöglicht es Ihnen, Ihren Amiga mit einer Bildschirmauflösung von bis zu 736 x 568 Punkten (Pal) zu betreiben. Dies entspricht einer Erhöhung von 28 % gegenüber dem üblichen Pal-Screen.

Die HIGH RESOLUTION WORKBENCH ist eine Befehlsweiterung für die Workbench oder das CLI, so daß sie einfach auf Ihre persönlichen Bedürfnisse angepasst werden kann. In einer Zeit, in der sich viele Software-Hersteller auf höhere Auflösungen, z.B. durch den neuen 204 Monitor, vorbereiten sind schon jetzt viele Programme in der Lage mit höheren Auflösungen zu arbeiten. Z.B. arbeitet Ed mit der HIGH RESOLUTION WORKBENCH mit 92 Zeichen pro Zeile (sonst 80). Auch Anwenderprogramme wie Professional Page, PageStream oder Deluxe Photolab nutzen die neue Auflösung, und es ist nun z.T. möglich auch ohne Interlace professionell zu arbeiten.

Grundsätzlich kann man sagen, daß fast alle Programme, die auf der Workbench-Oberfläche

arbeiten, also ohne eigenen Screen, die HIGH RESOLUTION WORKBENCH nutzen können.

Sie können die Auflösung mit der Sie arbeiten wollen frei wählen. Da im Overscan-Modus gearbeitet wird, ist die darstellbare Auflösung aber auch noch von Ihrem Monitor abhängig. Mit einem "normalen" Amiga Monitor sollte die horizontale Auflösung nicht größer als 732 Punkte gewählt werden, da das dargestellte Bild sonst am linken Rand verzerrt wird. Bei Multisync- oder Monochrom-Monitoren sind die vollen 736 Punkte darstellbar.

Die Daten:

- Für alle Amiga (ab Dos1.2)
- Bis zu 736 x 568 Punkte Workbench
- Bis zu 92x71 Zeichen pro Bildschirmseite
- Resetfest
- Bis zu 16 Workbench-Farbe
- Kompatibel zu allen Flicker-Fixer-Karten
- Beliebiges aus- und einschalten

Erhältlich ist das Programm zum Preis von DM 19,80 zuzüglich DM 2,- Versandkosten bei Olaf Penugaow, Kaiser-Friedrich-Str. 17, 1000 Berlin 10, Tel.: 030/88 33 505. Der Betrag sollte der Bestellung als Scheck oder in bar beiliegen.

Kurz & Klar

AMIGA GFA-BASIC

3.0

PETER FRITZEN
KLAUS SCHNEIDER
OLIVER STEINMEIER

Der GFA-BASIC 3.0-Interpreter stellt zur Zeit eines der leistungsfähigsten Entwicklungssysteme für den Commodore Amiga dar. Aufgrund seines sehr großen Befehlsatzes (mehr als 300 Befehle und Funktionen) kann man nahezu alle Probleme auf einfache Art und Weise lösen. Allerdings ist es fast unmöglich, die Syntax und exakte Funktionsweise sämtlicher Anweisungen auswendig zu kennen. „Kurz & Klar – Amiga GFA-BASIC 3.0“ soll hier helfen. In kurzer, prägnanter Form werden die Kommandos des Interpreters (incl. der im Handbuch fehlenden Anweisungen wie die Bob-Steuerung) in alphabetischer Reihenfolge erklärt. Dabei ist jeweils die genaue Syntax angegeben. Kurze Beispiele beseitigen letzte Unklarheiten. Das Buch enthält weiterhin eine nach Sachgruppen geordnete Befehlsreferenz, einen ausführlichen Index sowie zahlreiche nützliche Tabellen und Übersichten, die bei der Programmierung in GFA-BASIC auf dem Amiga benötigt werden. Kurz gesagt enthält das „Kurz & Klar – Amiga GFA-BASIC 3.0“ alles, was man zur Entwicklung eigener Programme benötigt, wenn grundlegende Kenntnisse der Sprache selbst bereits vorhanden sind.



AUS DEM INHALT:

- ▶ umfangreiche alphabetische Befehlsübersicht mit genauer Syntaxangabe, Befehlsklärung und kurzen Beispielen
- ▶ Sachgruppen-Übersicht aller Anweisungen des GFA-BASICs
- ▶ zahlreiche nützliche Tabellen und Übersichten (Füllmuster, ASCII, Scan-Codes, Schriftarten, CSI-Codes, etc.)
- ▶ umfangreiches Schlagwortregister

Das übersichtliche Nachschlagewerk

ca. 250 Seiten,
Hardcover **DM 29,-**

Preise sind unverbindlich
empfohlene Verkaufspreise

Heim Verlag

Heidelberger Landstr. 194
6100 Darmstadt-Eberstadt
Telefon 0 61 51-56057

BESTELLCOUPON Kurz & Klar – Amiga GFA-BASIC 3.0

Bitte senden Sie mir _____ St. **Kurz & Klar – AMIGA GFA-BASIC 3.0**
zzgl. DM 5,- Versandkosten (unabhängig von bestellter Stückzahl)
 per Nachnahme Verrechnungsscheck liegt bei

Name, Vorname _____

Straße, Hausnr. _____ PLZ/Ort _____

Benutzen Sie auch die in KICKSTART vorhandene Bestellkarte.

DAS NEUESTE VOM SPIELEMARKT

von ddf, cpl und ak

RUNNING MAN - GRAND SLAM

Möglicherweise haben Sie den Film RUNNING MAN mit Arnold Schwarzenegger gesehen, in dem er einen Polizisten aus dem Jahr 2019 spielt, der unschuldig eines Verbrechens angeklagt wird. GRAND SLAM hat die Chance ergriffen und bietet das "Spiel zum Film" jetzt an. Arnold wird gejagt von Freaks wie Subzero, Buzzsaw (man beachte die Aussprache!) und Dynamo. Die Suche nach Liebe, Gerechtigkeit und der entscheidenden Waffe führen in die Tiefen der Spielzone, wobei es sich als recht schwierig



Die Umsetzung vom Film bietet GRANDS SLAM an.

erweist, die Spielfigur zu steuern. Recht schwierig ist es auch, die Aufgabe als solche zu verstehen, und an manchen Stellen wird es zum Mysterium. Woran es fehlt, sind Ideen und Motivation. Im Gegensatz dazu ist die Aufmachung des Spieles gut gelungen. Besonders der Vorspann ist sehenswert, flüssige Animation und gelungener Sound. Nach dem beeindruckenden Start fällt das Spiel allerdings stark ab, und die mangelnde Steuerung gibt ihm den Rest. ■

AIRBALL



AIRBALL, seit langem angekündigt, jetzt endlich fertig.

Lange hat es gedauert, bis die kleine aufblasbare Kugel ihren Weg vom ATARI ST zum AMIGA gemacht hat. Nun rollt sie auf der Suche nach dem magischen Buch auch durch die Register unserer Rechner. Bereits vor zwei Jahren hat uns John Symes die erste Version gezeigt, und beinahe hätten wir nicht mehr an das Erscheinen des fertigen Spiels geglaubt. Die Frage, ob sich das Warten gelohnt hat, kann man allerdings mit einem entschiedenen Jain beantworten.

Einerseits ist hier ein faszinierendes dreidimensionales Labyrinthspiel, mit Einsammeln von Gegenständen, "Lufttankstationen",

um die Kugel aufzupumpen, Geschicklichkeit und Timing erfordernden Parcours etc., etc. Andererseits ist die Steuerung der - zwei Jahre alten und graphisch identischen - ATARI ST-Version wesentlich genauer. Nun ist aber die Steuerung bei einem Geschicklichkeitsspiel gerade das, worauf es ankommt. Deshalb ist unsere Freude etwas gedämpft. Es wird also ein ganz kleines Weilchen dauern, bis wir mit AIRBALL uneingeschränkt zufrieden sind - so lange eben, bis wir damit uneingeschränkt zurecht kommen. Trotzdem nimmt es unter den Neuerscheinungen mit Recht einen der besten Plätze ein. ■

FRIGHT NIGHT

MICRODEAL, einst bekannt für Software bester Qualität, scheint zur Zeit kaum etwas veröffentlichen zu können, ohne daß das Publikum darüber stöhnt. FRIGHT NIGHT, verknüpft mit dem bekannten Film und geschrie-

die Sie dafür über den Ladentisch geschoben haben. Okay, wir haben ein exzellentes Titelbild, daß unsere Hände vor Gier zum Schwitzen bringt. Auch die Soundeffekte liegen jenseits jeglichen Vergleichs - aber das ganze



Der Film ist glänzend, aber das Spiel ist schrecklich.

ben von Englands bekanntestem und produktivstem Programmierer Stave Bak, besitzt die Voraussetzungen für ein hervorragendes Programm. Das Ergebnis wird Ihnen manche schlaflose Nacht bereiten, besonders wenn Sie an die beinahe siebzig Mark denken,

Spiel besteht nur aus dem Herumlaufen in einem Haus, dem Vermeiden von Poltergeistern und dem Einsammeln von Opfern. Die Graphik ist fein, die Musik großartig, der Film ist glänzend, aber das Spiel ist schrecklich AAHHHHH!!! ■

LORDS OF THE RISING SUN



THE LORDS OF THE RISING SUN versetzt Sie in das 12. Jahrhundert.

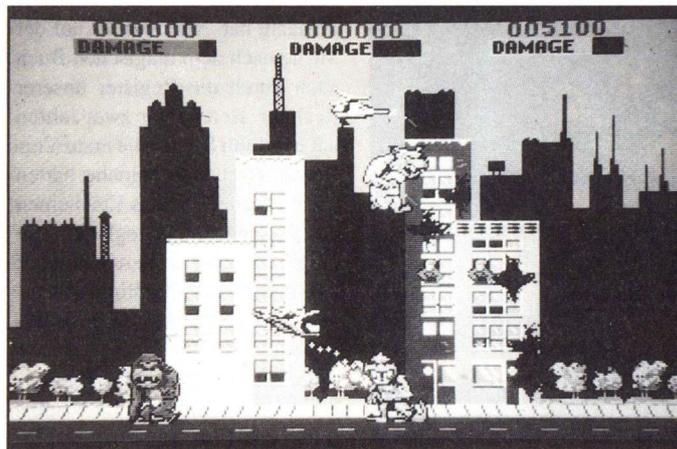
Willkommen im Japan des zwölften Jahrhunderts! Ihre bitteren Rivalen vom Stamm der Taira haben Ihren Vater ermordet und den Kaiser gestürzt. In einem Feldzug versuchen Sie sich zum Shogun zu machen und die Ehre der Familie zu retten. Sie führen Ihre Samurai-Krieger in den Kampf, belagern mächtige Festungen, handeln Bündnisse aus und stellen sich Ihren Feinden im Zweikampf.

Mit LORDS OF THE RISING SUN hat CINEMAWARE ein weiteres Spiel des bewährten Strickmusters auf den Markt ge-

bracht. Actionsequenzen, bei denen der Spieler einen gewissen Einfluß nehmen kann, sind gepaart mit strategischen Elementen. Die Graphik ist wieder vom Feinsten, was es für den AMIGA gibt. Sehr gut ist auch die komplett übersetzte Bedienungsanleitung. Der Hersteller weist übrigens darauf hin, daß es Grauimporte gegeben hat und deshalb auch Programme ohne deutsche Anleitung angeboten werden. Diese können nicht umgetauscht werden. Wenn Sie das Programm also bestellen, fragen Sie besser nach, ob es eine deutsche Anleitung hat. ■

RAMPAGE

RAMPAGE war lange ein klassisches ARCADE GAME mit einer eigenen Fangemeinde und hat noch heute den Ruf eines Kultspiels. Die Umsetzung für den AMIGA ist eine Katastrophe. Daß eine Firma mit einer so guten Reputation wie Activision sich erlaubt, diese sogenannte Umsetzung auf den Markt zu bringen, ist schamlos. Wenn es möglich wäre, ein Spiel direkt vom 8 Bit-Rechner zu portieren, das Resultat würde genauso aussehen. Die phantasielosen Sprites und ihre primitive Animation unterscheiden sich qualitativ nicht von dem niedrigen Niveau, auf dem sich die



8 Bitter auf dem AMIGA, so oder ähnlich könnte man RAMPAGE in dieser Umsetzung bezeichnen.

Häuser und Wolkenkratzer befinden.

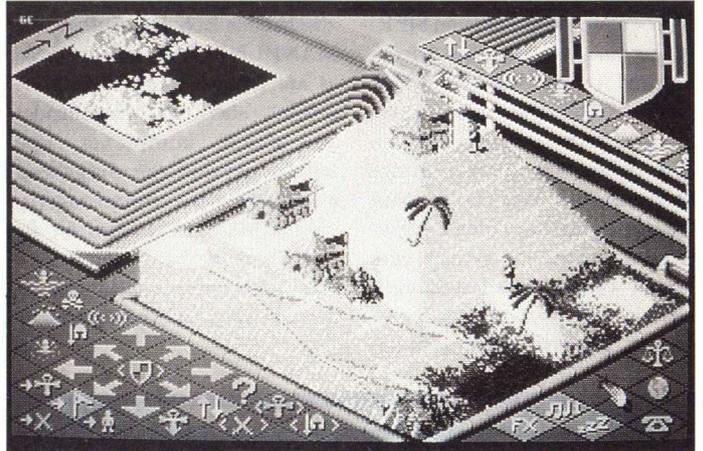
Das Spiel dreht sich um King Kong, Godzilla und einen Wolf Man, die versuchen, die Gebäude in 85 amerikanischen Städten zum Einsturz zu bringen. Wenn Sie sehen, WIE Godzilla einen Wolkenkratzer erklettert und von einem Kampfhubschrauber angegriffen wird, wird sich auch Ihnen die Frage nach dem Sinn dieser Veröffentlichung aufdrängen. ■

TOP 12

POPULOUS, unser TOP 12-Tip des letzten Monats, hat sich an die Spitze der Hitparade gesetzt. Dem neuen Liebling der Spieleszene ist es auf Anhieb geglückt, die Spitze zu erklimmen, und wir hoffen, daß sich das Spiel auch länger dort halten wird.

Mitmachen kann jeder, der die TOP 12-Postkarte (bitte keine andere!) ausfüllt und uns zurückschickt. Der Rechtsweg ist dabei ausgeschlossen. Einsendeschluß ist der 30. Juni 1988. Karten, die nach diesem Termin eintreffen werden im nächsten Monat berücksichtigt. Unter allen Einsendern verlosen wir eine Menge aktueller Spiele - mitmachen lohnt sich!

1. (12) Populous



2. (1) Elite

3. (2) Falcon

4. (6) Ports of Call

5. (3) Dungeon Master

6. (4) Holiday Maker

7. (-) Super Hang-On

8. (9) Bard's Tale II

9. (11) Zak McKracken

10. (8) Interceptor

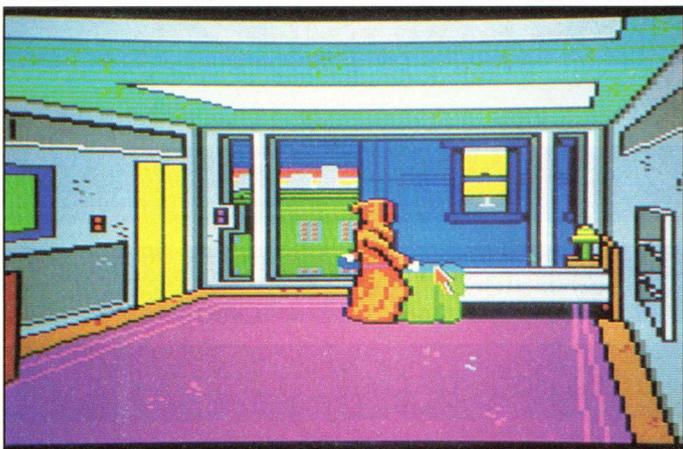
11. (5) Shanghai

12. (-) Carrier Command

MANHUNTER

Das neueste Programm von SIERRA ist eine Mischung aus Adventure und Arcade - MANHUNTER NEW YORK. Das Spiel benutzt das gleiche System wie viele andere Sierra-Abenteuer - Kings Quest, Leisure Suit Larry, Space Quest etc., die von vielen Computerfreaks entweder geliebt oder gehaßt werden. Alle Handlungen werden durchgeführt von einem großen, graphisch angenehm an-

zusehenden Bob. Der Aufbau von MANHUNTER ist gelungen, da es am Anfang recht einfach erscheint, aber mit zunehmender Spieldauer komplexer und schwieriger wird. Ihre Aufgabe ist es, den Planeten von einer außerirdischen Macht zu befreien, die die Erde überfallen hat. Voller Humor und hinter sinniger Elemente ist dies ein Spiel für Abenteuerfans. ■



Die Graphik ist wie immer bei Spielen von SIERRA nur Durchschnitt, aber die Spielhandlung ist wieder einmal Spitze.

STADT DER LÖWEN



STADT DER LÖWEN ist das zweite Werk nach HOLIDAYMAKER von den beiden Programmierern C. Föhlding-Hornschuh und M. Frisch.

Als Erstlingswerk brachten Chris Föhlding-Hornschuh und Markus Frisch das Adventure (Artventure) HOLIDAYMAKER heraus. Seit Monaten ist das erstklassige Adventure schon in den Spielcharts ganz oben vertreten. Mit STADT DER LÖWEN präsentieren die Entwickler in naher Zukunft ihr zweites 'Artventure'. Handlungsort wird Singapur sein,

das auch als STADT DER LÖWEN bezeichnet wird. Wie auch bei HOLIDAYMAKER sind wieder erstklassige Grafiken zu erwarten, ganz zu schweigen von der ausgeklügelten Spielhandlung. Legen die beiden Programmierer doch viel Wert auf Realitätsnähe, beispielsweise existierenden Straßen- oder Ortsbezeichnungen wirklich. ■

B L O O D M O N E Y

Blutiges Geld, wer jetzt meint, es handle sich um ein äußerst brutales Spiel, irrt. Es ist vielmehr ein weiteres Ballerspiel, das sich im Grunde kaum von alten Klassikern wie KATAKIS oder MENACE unterscheidet, letzteres Spiel stammt übrigens von den gleichen Programmierern. In England konnte es große Verkaufszahlen erzielen, in deutschen Ländern war der Verkauf jedoch bescheidener. Das Besondere an BLOOD MONEY sind jedoch ein paar extravagante Features. Der oder die Spieler müssen, was sonst, auf alles ballern, was sich vor die Laserkanone wagt. Nach Abschluß verlieren die Widersacher jedoch ihre Ladung, die die Form einer Münze besitzt. Sie sollten eingesammelt werden, da das "blutige Geld" zu Extrawaffen verhilft. Dazu müssen Verkaufsplattformen angefliegen werden, die hin und wieder auftau-



Besonders das gleichzeitige Spielen macht enorm viel Spaß.

chen. Die Grafik von BLOOD MONEY ist sehr ansprechend, alle Levels sind gut 'durchgestillt' und durchaus keine Beleidigung für das Auge. Beim Sound haben die

Programmierer jedoch etwas gespart. Der Schwierigkeitsgrad von BLOOD MONEY ist recht hoch, am besten spielt man zu zweit gleichzeitig, wobei der eine die

Gegner abballert und der andere das Geld einsammelt. FAZIT: Die Spielidee ist nicht neu, Ballerspielfans kommen aber auf ihre Kosten. Die kleinen Strategieelemente bereichern das Spiel ungemain. Besonders das Spielen zu zweit macht einen Heidenspaß. Vertrieben wird das Ballerspiel von dem bekannten Softwarehaus PSYGNOSIS, alleine der Anbieter bürgt schon für ein gutes Produkt, stammen doch BARBARIAN, OBLITERATOR oder BAAL aus diesem Hause.

ak

	<input type="checkbox"/>	Idee
	<input type="checkbox"/>	Grafik
	<input type="checkbox"/>	Sound
	<input type="checkbox"/>	Motivation
	<input type="checkbox"/>	

CYBERNOID II

**CYBERNOID II, THE REVENGE -
die Fortsetzung zu Cybernoid, der Kampfmaschine,
beinhaltet zusätzliche Levels, animierte Landschaften
und komplett neue Angriffswaffen.**



Ausgezeichnete Graphik und ein hoher Schwierigkeitsgrad sind
Markenzeichen auch bei CYBERNOID II.

Die Piraten sind zurückgekommen in einem neuen Schlachtschiff, das noch furchteinflößender ist als das erste, und haben die Warenlager der Föderation geplündert. Da Sie aufgrund Ihres letzten Einsatzes ein so hohes Ansehen genießen, hat die Kommission Sie ausgewählt, den Planeten von der Piratenplage zu befreien und die gestohlenen Güter zurückzubringen. Ausgerüstet mit den neuesten, unglaublichen Waffensystemen steigt Ihr Raumgleiter auf, um sich der Konfrontation zu stellen. Während Sie durch die dunklen Höhlenkammern fliegen, versuchen außerirdische Monster, Sie aufzuhalten, indem sie von herunterhängenden Lianen Säuretröpfen auf Sie sprühen. Wände, bestehend aus beweglichen Augen, beobachten, wie Sie versuchen, durch die engen und gewundenen Pfade zu manövrieren, ohne zu kollidieren. Der Bildschirm wird nicht gescrollt, sondern weitergeschaltet, was darauf hindeutet, daß das Spiel für den ATARI ST geschrieben und dann einfach portiert wurde. Die Handlung ist einfach zu verstehen - im wesentlichen fliegen Sie durch eine farbenprächtige Landschaft und sammeln zufällig verstreute Waffen ein, die die Feuerkraft des

Schiffes erhöhen. Eine genauere Beschreibung finden Sie in den Besprechungen von ... und ...nd ...d. Ihr aktueller Status wird stets aktuell gehalten. Auf dem Panel werden die Anzahl der verbleibenden Leben, die Menge der eingesammelten Waren und die erzielten Punkte angezeigt. Daneben erscheint noch die jeweils benutzte Waffe und ihre Ladekapazität. Als Letztes gibt es noch eine Anzeige für die verbleibende Zeit. Sollte sich diese auflösen, ereilt Sie das Schicksal. Wie zahlreiche andere Vertreter dieser Art besitzt auch CYBERNOID II sehr gute Graphik und Animation. Nachteilig für den Fortgang des Spieles ist, daß man, einmal "verstorben", ganz von vorne beginnen muß. Wenn Sie ein wenig masochistisch veranlagt sind und eine derartige Herausforderung Sie reizt, versuchen Sie es ruhig.

ddf



Die Wüste... heiß... flach. Straßen, die geradeaus in die Unendlichkeit führen. Der Traum jedes amerikanischen Porschebesitzers. Stellen Sie sich jetzt vor, Sie sitzen selbst am Lenkrad eines silbernen Boliden, bekannt als Porsche 959. Das ist das Gefühl, das Ihr Händler Ihnen beschrieben hat, als Sie ihm eine Viertelmillion Dollar in den Rachen gesteckt haben. Sie rollen mit gemütlichen 200 km/h dahin, sehr entspannt, der König der Landstraße. Nichts kann Ihnen etwas anhaben. Zumindest erscheint es Ihnen so. Plötzlich erscheint ein kleiner Punkt in Ihrem Rückspiegel. Was immer es ist, es muß mindestens 240 km/h schnell sein...

Soviel zur Einleitung durch ACCOLADE. Das Rennen zwischen dem legendären Ferrari F40 und dem Porsche 959 kann beginnen. Die Fortsetzung von Test Drive hat eine ganze Weile auf sich warten lassen. Angenehm ist, daß es neben der original Disk gleich noch die Datendisketten CALIFORNIA CHALLENGE mit zusätzlichen Kursen und THE SUPERCARS mit den Autos Lotus Turbo Esprit, Ferrari Testarossa, Porsche 911 RUF, Lamborghini Countach 5000S und Corvette ZR1 (Baujahr 89) angeboten werden. Die Cor-

The Duel Test Drive II



Ein Porsche 959 dient diesmal als fahrbarer Untersatz.

vette ist übrigens noch nicht im Handel, aber ACCOLADE hat die Daten direkt aus Detroit. Genau wie der Vorgänger TEST DRIVE ist THE DUEL eine Fahrsimulation. Am Programm selbst hat sich nur wenig geändert. Graphik

und Sound sind ausgezeichnet und vermitteln ein recht realistisches Gefühl. Wer sich in Spielhallen auskennt, hat vielleicht schon den neuen Fahr Simulator von ATARI gesehen. Die Simulatoren von ACCOLADE sind die einzigen

AMIGA-Programme, die diesem Spielhallenautomaten auch nur entfernt ähnlich sind.

Einige Unterschiede zum ersten Teil sind mir aber gleich bei der ersten Testfahrt aufgefallen. Die Cops haben ihre Streifenwagen inzwischen mit einem Turbolader nachgerüstet. War es in der ersten Version kaum möglich, erwischt zu werden, so ist es jetzt ganz schön schwierig, die Jungs abzuhängen. Mittlerweile hält das Auto auch nicht mehr von alleine an der Tankstelle, sondern der Fahrer muß selbst aufpassen, wie es mit seinem Sprit steht. Gleichgeblieben ist die manchmal zu träge reagierende Steuerung und das damit verbundene vorzeitige Ableben des Testfahrzeugs.

Zwei Bemerkungen noch zum Schluß: Gut finde ich, daß man THE DUEL auf der Festplatte installieren kann. Unverschämte ist die Forderung von \$15 für eine Sicherheitskopie.

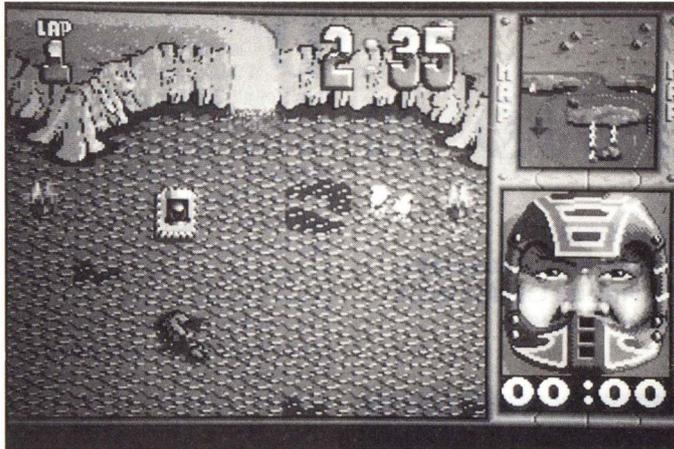
cpl



RUN THE GAUNTLET

RUN THE GAUNTLET nennt sich die Neuerscheinung von OCEAN. Pate stand eine populäre englische Fernsehserie, in der bekannte Persönlichkeiten in Spielen unterschiedlichster Art gegeneinander antraten. Dieser "sportliche Wettkampf" wird nun auf dem Computer fortgesetzt. Im britischen Fernsehen verhalf bereits die Tatsache, daß man "Lieblinge der Nation" dabei beobachten konnte, wie sie sich selbst zum Narren machten, der Serie zu Einschaltrekorden.

Für die bundesdeutschen AMIGA-Besitzer ist RUN THE GAUNTLET allerdings nur "noch ein Spiel" und deshalb wesentlich weniger "positiv vorbelastet". Eine Serie, in der sich die Darsteller selbst und gegenseitig zum Narren machen, gibt es auch im deutschen Fernsehen unter dem Titel "Heute im Parlament". Die Teilnehmer sind allerdings bei weitem nicht so sportlich wie in RUN THE GAUNTLET. Das Spiel selbst ist unterteilt in drei Disziplinen: Bootsrennen, Crossfahren und Geländelaufen.



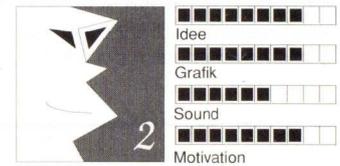
Erst nach einer geraumen Zeit werden Sie RUN THE GAUNTLET zu schätzen wissen.

Jede Kategorie hat Untergruppen, die per Zufallsgenerator ausgewählt werden. Es gibt Boote und Geländefahrzeuge mit unterschiedlichen Eigenschaften. Je nach Kurs und Material ist eine andere Taktik notwendig. Beim Bootsrennen stehen beispielsweise Tragflächen-, Renn-, Schlauchboote und Jet-Skis zur Wahl. Vier Teams nehmen an der Startlinie Aufstellung. Davon können bis zu drei menschliche Teilnehmer sein. Die Teilnehmer treten nacheinander an. Jedes Segment muß innerhalb einer vorgegebenen Zeit gelöst werden. Graphisch gut gelun-

gen ist der Kurs für das Bootsrennen, der aus der Vogelperspektive gezeigt wird. Er sieht sehr leicht aus, aber wie bei allen Disziplinen täuscht dieser Eindruck gewaltig. Die Steuerung des Fahrzeugs ist sehr gewöhnungsbedürftig, und viele Zusammenstöße, Dreher und Kollisionen werden hinter Ihnen liegen, bevor Sie das Fahrzeug unter Kontrolle haben. Der Geschmack an RUN THE GAUNTLET kommt beim Spielen. Jedes Fahrzeug hat einen individuellen Kurs. Ihre Position wird auf einer kleinen Karte angezeigt, die in der rechten Ecke des Bild-

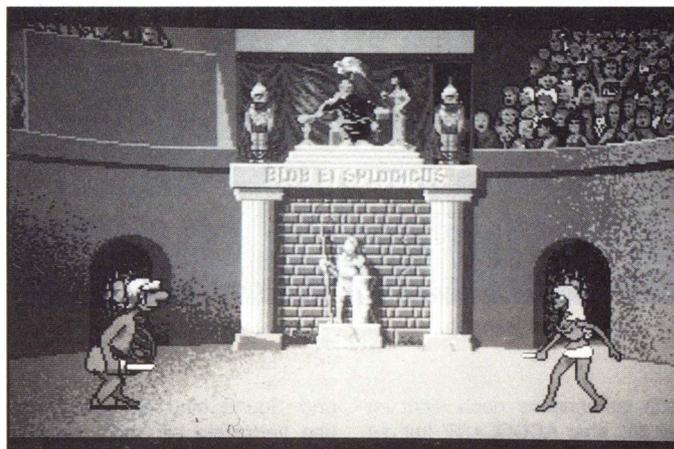
schirms plaziert ist. Aus dieser Karte ist auch die Strecke zu entnehmen, der Sie zu folgen haben. Graphische Effekte wie Wasserfälle und Explosionen verschönern den Kurs. Hüten Sie sich aber, ihnen zu nahe zu kommen. Der schwierigste Teil des Rennens ist der Geländelauf. Wer sich durch Sümpfe und über schlüpfrige Baumstämme gequält hat, weiß, wieso wir diesem Spiel eine bessere Steuerung wünschen. Dies ist einer der Gründe, warum es lange dauern wird, bevor Sie dieses Spiel gemeistert haben. RUN THE GAUNTLET ist der Typ von Spiel, zu dem die Liebe erst auf den zweiten Blick kommt. Am Anfang mag es enttäuschend sein, denn es ist schwierig, und Fortschritte sind nur langsam möglich. Wenn Sie sich allerdings einmal "eingespielt" haben, werden Sie auch so leicht nicht wieder davon loskommen.

ddf



Das Leben eines Gladiators war niemals leicht. Bewaffnet nur mit einer Toga, Schild und Schwert treten Sie Gegnern mit amüsanten Namen wie Herr Liss, Bon d'Age und Bud Vyzer gegenüber. Ludicrus ist ein ziemlich tragisch anzuschauender Vertreter der menschlichen Rasse, aber seine Aufgabe ist einfach, sobald es Ihnen einmal gelungen ist, die Steuerung zu handhaben. Jeder Gegner hat seine eigene Kampftechnik und ist deshalb auf eine ganz bestimmte Weise zu schlagen. Ein einfacher Zug wird Ihnen eine ganze Serie von Erfolgen beschern: Ziehen Sie den Joystick nach rechts und drücken Sie auf den Feuerknopf - und Sie werden feststellen, daß Sie auf diese Art und Weise 60% der Kämpfe gewinnen. Jeder Gegner muß in einem Durchgang über fünf Runden geschlagen werden. Am Ende signalisiert der Caesar per Daumen den Einzug in die nächste Runde. Sollte Ludicrus unterliegen, heißt es ab in die Löwengrube. Er steht dann einem Löwen gegenüber, den er besiegen muß, um nicht auszuschneiden. Der oben beschriebene Zug garantiert den

I LUDICRUS



In der Kampfarena gilt es, Gladiatoren niederzustrecken.

Erfolg bei dieser zahnlosen Katze. Mehr gibt es über dieses Spiel nicht zu sagen, außer daß es graphisch ganz akzeptabel aussieht. Es besitzt große ansprechende Cartoon-Charaktere, die recht fließend animiert sind. Besonders die vielen verschiedenen Bewegungsarten der Figuren sehen beeindruckend aus, aber leider erweisen sie sich im Spiel als absolut ineffektiv. Den Sound näher vorzustellen, wäre Verschwendung, ebenso wie der Humor, der das Spiel durchzieht, an I LUDICRUS verschwendet ist. Allerdings, um dieses Spiel mehr als einmal zu spielen, braucht man entweder eine gesunde Portion Masochismus oder sehr viel Humor.

ddf



Sie sind Stuart, ein junger Angestellter der IRC. Diese Firma baut und testet Roboter auf 10 Plattformen mitten im Weltraum. Wenn ein Roboter alle Plattformen durchsteht, darf er verkauft werden, eher nicht! Aber das geht Sie nichts an. Ihre Aufgabe ist es nur, von einer Umlaufbahn der Plattformen aus den Roby per Joystick durch die Landschaften zu steuern und am Ende jeweils den Ausgang aus dem Labyrinth zu erreichen. Das ist manchmal ganz schön knifflig. Türen mit Laserstrahlen versperrten den Weg. Da muß man erst die passende Schlüsselcodekarte finden. Außerdem liegen überall unsichtbare Minen herum, die in die Luft gehen, wenn man drüberläuft. Deshalb empfiehlt es sich, den Minendetektor zu suchen, der herumliegende Minen sichtbar macht. Auch die Codekarte, welche die Schußfunktion Ihres Roboters auslöst, wäre ratsam zu finden. Auf den Plattformen ballern nämlich Geschütze wild in der Gegend herum. Manchmal muß man auch gefährliche Minenleger beseitigen. Außerdem gibt es noch Codekarten, die einen zum Ausgang führenden Zeiger auf den Bildschirm bringen. Man sollte

NIGHTDAWN

Roboter auf der Prüfpiste

versuchen, die gesamte Ausrüstung zusammenzuraffen. Denn in den höheren Leveln kommt man ohne sie nur schwer weiter.

Fazit: Sehr gefallen haben mir bei Nightdawn die Soundeffekte. Das Öffnen der Schiebetüren, Explosionen und die surrende Kreis-



Grafisch kann NIGHTDAWN keineswegs überzeugen, dazu sind die Roboter und die Plattformen zu grob aufgelöst gestaltet.

bombe sind hervorragend akustisch in Szene gesetzt. Endlich einmal gelungene Soundeffekte! Auch die Titelmelodie ist nicht zu verachten. Aber was Magic Bytes beim Sound an Lob verdient, das verdienen sie an Tadel bei der Grafik: Das Scrolling ist zwar ruckelfrei, jedoch bringen verwaschene Farben der Plattformen bei einer Darstellung aus der Vogelperspektive und das kümmerliche Aussehen des Roboters die Augen zum Tränen. Für 85 DM darf man größere grafische Leistungen erwarten. Aber dafür macht "Nightdawn" wenigstens Spaß. Das ist doch schon was!

cbo

Hersteller: Magic Bytes
Info: Ariolasoft



LEONARDO

Ein diebisches Vergnügen...

Nachts wird Leonardo aktiv. Dann bricht er in fremde Häuser ein und klaut alles, was nicht niet- und nagelfest ist. Er hält sich zwar für den perfektsten Meisterdieb aller Zeiten, trotzdem

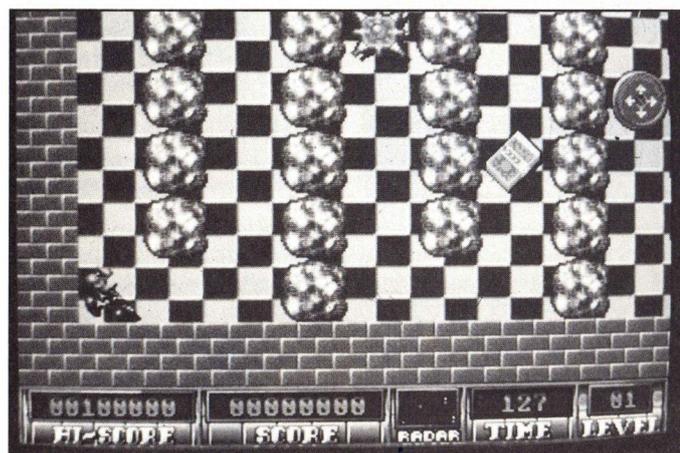
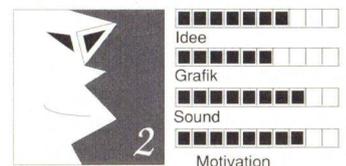
sind seine nächtlichen Unternehmungen höchst gefährlich. Die Polizei würde den quirligen Gauer zu gerne in eine Gefängniszelle stecken. Jetzt hat der verrückte Leo einen tollkühnen Plan. Fünfzig

Häuser will er heimsuchen. Dort befinden sich wertvolle Gegenstände, die es zu stehlen lohnt: Brillanten, Geldscheine, Computer... Sie können ihm dabei helfen. Das Szenario präsentiert sich ähnlich wie bei Boulder Dash aus der Vogelperspektive. In jedem der fünfzig Räume befinden sich Wertgegenstände, die Leo zusammenraffen muß. Dazu schiebt er sie alle in eine Ecke. Steine und andere Hindernisse muß er zuvor wegräumen, damit der Weg für die Gegenstände frei ist. Doch Vorsicht! Polizisten, Gespenster und andere Widersacher trachten Leonardo nach seinen fünf Bildschirmleben. Wenn seine Verfolger zu dreist werden, kann Leo sie genau wie bei Boulder Dash mit Felsbrocken (sowas hat schließlich jeder im Wohnzimmer herumliegen!?) zermatschen. Neben Felsen und Wertgegenständen trifft Leo auch auf Bonussymbole, die ihm mehr Punkte verschaffen oder seine Widersacher für kurze Zeit außer Gefecht setzen.

behält man mit dem Radar den Überblick. Vorausgesetzt, das Ding funktioniert. Leos Konstruktionen sind leider oft sehr fehlerhaft. Da kann es schon mal passieren, daß das Radargerät mitten im Spiel ausfällt. Der ganze Trubel, der in jedem Level herrscht, sorgt für unterhaltsame Spielstunden. Man will schließlich jeden der 50 Räume heimsuchen. Der einzige Schwachpunkt bei Starbytes Leonardo liegt bei der leider durchschnittlichen Grafik. Wer auf Spitzengrafik verzichten kann, sollte mal ein Probespielchen wagen.

cbo

Hersteller: Starbyte
Info: Bomico



Leonardo, der Meisterdieb, muß 50 Räume leerräumen.

Abwechslung bringt auch Leos selbstgebasteltes Radargerät ins Spiel. Da sich ein Raum stets über mehrere Bildschirme erstreckt,

Battletech

Battletech spielt in ferner Zukunft. Fünf mächtige Staaten kämpfen in einem Krieg ohne Ende um die Weltherrschaft. Panzer, Atomwaffen, Raketen und Gewehre gibt es schon seit langem nicht mehr. Stattdessen kämpft man mit riesengroßen Kampfmaschinen, den Battlemechs. Von außen sehen Sie wie leblose Roboter aus. Doch im Innern der riesigen Stahlhülle verbirgt sich stets ein Mensch, der die Kampfmaschine steuert. Das ist natürlich äußerst kompliziert, deshalb müssen alle Battlemechs vor dem Einsatz in ein Trainingscamp. Dort gibt es Gebäude, in denen man Informationen bekommt. Im Trainingszentrum kann man die Kampfkraft seines Battlemechs verbessern. Zuvor darf man sich allerdings jedesmal über den Kopierschutz ärgern. Immer, wenn man ins Trainingszentrum möchte, blendet der AMIGA einen Kampfroboter ein. Mit Hilfe des

Handbuchs muß man alle Teile erkennen, dann erst geht's ins Trainingszentrum. Da man seinen Battlemech öfters trainieren muß, wird diese Prozedur auf die Dauer ganz schön lästig. Im Trainingslager bekommt man mehrere Aufgaben, die es zu erfüllen gilt. Hat man mehrere Trainingsstufen absolviert und die Eigenschaften des Battlemechs verbessert, ist man gewappnet für den Kampf mit feindlichen Robotern. Dies macht den Hauptinhalt des Spiels aus. "Battletech" ist ein passables Rollenspielchen, jedoch war ich von der Umsetzung ein wenig enttäuscht. Ist man doch von Infocom nur erstklassige Programme gewohnt. Bei "Battletech" gibt es keine Joysticksteuerung. Man muß da mit einer unpräzisen Maussteuerung oder der Tastatur vorlieb nehmen. Das Scrolling im Trainingscamp ruckelt wie bei einem Erdbeben. Lange konnte mich "Battletech" beim Test nicht



INFCOM, bekannt durch erstklassige Textadventures, landet mit Battletech wohl einen Flop.

motivieren. Alle Vorgänge wiederholen sich ständig. Das andauernde Heraussuchen von Roboterteilen aus dem Handbuch ging mir so auf den Geist, daß mein Testmuster jetzt im Mülleimer liegt.

Hersteller: Infocom
Info: Ariolasoft

cbo



Das neue kanadische Softwarehaus Visionary Design bringt als Nachfolger von Dragons Lair DATASTORM auf den Markt. Der Programmierer ist bereits ein alter Programmierhase, sein Pseudonym ist 'SO-DAN'. Bei DATASTORM handelt es sich um eine waschechte Defender-Variante. Bereits vor etlichen Jahren kam der gleichnamige Spielautomat auf den Markt und machte Furore. Eine Umsetzung auf den C64 ließ nicht lange auf sich warten. Demzufolge ist das Spielprinzip gleich geblieben: Der Spieler steuert ein Raumschiff auf einem links/rechts-scrollenden Bildschirm und ballert auf fast alles, was sich dort so tummelt, und das ist nicht gerade wenig - Lander, Crazies, Tornados, Blitter, Gravitrans, Pulsare, Saturns, Baiter, Hunter, Bomber bevölkern den Bildschirm, doch das sind noch lange nicht alle. Die Aufgabe in jedem Level besteht darin, sogenannte Pods zu einer Warpstation zu bringen, jedoch sammeln die Lander die Pods ebenfalls auf und verwandeln sich dadurch in einen äußerst gefährlichen Crazy. Sind alle Pods gerettet und alle Eindringlinge vernichtet, geht es im nächsten Level weiter. Der Spieler sieht nur einen begrenzten Teil des Rundumspielfeldes, ein Radar

DATASTORM

Defender läßt grüßen



Überdimensionale Krakenraumschiffe gilt es erst in höheren Levels aus dem Weg zu räumen.

am unteren Bildrand gibt aber Auskunft, wo sich noch Eindringlinge und Pods befinden. Zu Beginn des Spiels kann der Spielmodus eingestellt werden - Geschwindigkeit, Anzahl der Spieler, Joystick oder Tastatur. Ein besonderes Feature: Es können auch

zwei Spieler gleichzeitig den Eindringlingen zu Leibe rücken. Extrawaffen tauchen hin und wieder nach Abschluß eines Gegners auf, sie verleihen dem Raumschiff einen Schutzschild, Dauerfeuer oder etliches mehr. Ab dem vierten Level tauchen noch Supergegner

auf, beispielsweise der überdimensionale SPACEQUID oder der INTERGALACTIC SPACE SKULL. FAZIT: Die Spielidee von DATASTORM ist nicht neu, die Realisation des Spiels ist aber perfekt. Die Action ist wohl kaum zu überbieten. Die Grafik reißt zwar keine Bäume aus, ist aber recht gut. Die Animation sucht ihresgleichen, in wahnsinniger Geschwindigkeit bewegen sich die mitunter sehr zahlreichen Objekte über den Bildschirm. Die Motivation ist sehr hoch, allerdings brauchte ich nach jedem Spielchen eine kleine Verschnaufpause, um die Finger ausruhen zu lassen. Ballerspielfans, die viel Action lieben, sind mit DATASTORM bestens bedient.

ak

Anbieter:
Cachet
Ostendstr. 32
6524 Oestringen
Tel.: 0 72 53 / 2 24 11



OIL IMPERIUM

Moneten, Macht und miese Tricks

Normalerweise sind Wirtschaftssimulationen langweilig und trocken. Riesige Zahlenkolonnen, nüchterne Grafik, unzählige Statistiken und komplizierte wirtschaftliche Zusammenhänge machen diese Art von Spiel zwar realistisch, aber dafür langweilig. Ganz anders liegt

der Fall bei "Oil Imperium". Reline hat bei diesem neuen Wirtschaftsstrategiespiel weniger Wert auf Realitätsnähe, sondern mehr Sorgfalt auf exzellente Grafik und eine spannende Handlung verwendet. Als Geschäftsführer einer kleinen Ölfirma haben Sie zu Beginn die Wahl zwischen verschie-



In die Fußstapfen von JR Ewing läßt einen das Strategiespiel OIL IMPERIUM treten.

denen Firmenemblemern und Büroräumen. Geben Sie Ihren Namen ein, und los geht der Kampf um die Macht auf dem Ölmarkt. Bis zu vier Spieler können sich an "Oil Imperium" beteiligen. Gewinner ist, je nach Wahl des Spielziels, wer nach drei Jahren über das meiste Kapital verfügt, als erster 60 Millionen Dollar sein eigen nennt, 80% Marktanteil besitzt oder seine Gegenspieler ruiniert hat. Da wird in jedem ein kleiner JR Ewing geweckt. Man kann nämlich Saboteure anheuern und fremde Ölfelder in Brand stecken, die Hausbank der Gegner ausplündern, Handelsvertreter bestechen oder Öltanks in die Luft jagen. Jeder Saboteur kostet Geld. Je billiger sich eine zwielichtige Gestalt auf dem "Markt" anbietet, desto größer ist die Chance, daß sie bei Sabotageakten erwischt wird. Das hat unangenehme Folgen.

Neben diesen dunklen Mächenschaften kann man als Ölmanager natürlich auch neue Felder und Tanks kaufen bzw. verkaufen, Pipelines bauen, nach Öl bohren, sich Informationen über den Ölmarkt aus der Zeitung aneignen, Statistiken anschauen und Lieferverträge abschließen.

Alle Vorgänge werden durch wunderschöne Grafiken illustriert. Strategiespielfans haben in noch

keinem Spiel dieses Genres eine so gute Grafik zu Gesicht bekommen. Doch "Oil Imperium" ist nicht nur für Strategiefans interessant. Auch an die Fans von Actionsequenzen hat Reline gedacht. In drei im Programm integrierten Szenen müssen Sie an den Joystick, wenn Sie nicht das Geld für Spezialisten ausgeben wollen.

In diesem Testbericht können wir die mannigfaltigen Optionen von "Oil Imperium" nur streifen. Es gibt noch viele Optionen, die wir aus Platzgründen nicht nennen konnten. Relines neues Strategiespiel überzeugt nicht nur durch gelungene Grafiken, Komplexität und eine spannende Handlung, die ein oder mehrere Spieler stundenlang an den Bildschirm fesselt, sondern auch durch eine komfortable Benutzerführung.

cbo

Hersteller: Reline

Info: Rushware



Space Pilot 89

Klassiker neu aufgelegt!

Kingsoft, bekannt durch billige AMIGA-Software, hat in den letzten Monaten einen ganzen Haufen neuer Spiele auf den Markt geworfen. Darunter auch "Space Pilot 89", ein Ballerspielchen, das das Licht der Welt schon vor einigen Jahren auf dem C64 erblickte. Damals war "Space Pilot" monatelang in den englischen Verkaufscharts ganz oben. Aufgrund des großen Erfolgs hat man sich dazu entschlossen, das Spiel auch für die 16-Bitter umzusetzen.

"GET READY!" ruft eine digitalisierte Stimme. Ich umklammere meinen Joystick. Los geht's! Brauner Hintergrund und rote Wölkchen, die gleichmäßig über den Bildschirm scrollen, ziehen meine ersten Blicke auf sich. Per Joystick manövriere ich ein kleines Raumschiff in alle Himmelsrichtungen. Doch da! Ufos mit fünf roten Punkten kreuzen meine Flugbahn. Feu-

er, Feuer, bäh! Alle weg! Auf einer Anzeigetafel am rechten Bildschirmrand erkenne ich, wann ich in den nächsten Spielabschnitt gelange: Je mehr Ufos ich abballe, desto größer wird der Streifen. Hat er seine maximale Ausdehnung angenommen, kann man im nächsten Level andere Flugobjekte abschießen. Das wäre ganz schön langweilig, wenn nicht manchmal kleine Sonden auftauchen würden, die Extras bereithalten. Einfach einsammeln, und Schutzschilder, Schnellfeuerkanone oder Waffen mit einem größeren Streuwinkel erleichtern es, mit den unzähligen Flugobjekten fertigzuwerden. Die meisten Gegner fliegen nur stupide über den Bildschirm. Mit denen wird man ganz leicht fertig: Ausweichen oder abballern - und das Problem ist erledigt. Doch dummerweise feuern einige Bösewichter auch automatisch gelenkte Suchraketen ab.



Space Pilot erblickte das Licht der Computerwelt bereits auf dem alten C64.

Ausweichen bedeutet da besonders in den höheren Spielstufen ganz schönen Streß.

"Space Pilot 89" ist ein nettes Game für ein Ballerintermezzo zwischendurch. Doch weder in punkto Sound, Grafik noch Spielidee hält es den Standard, den wir von AMIGA-Software gewohnt sind.

Hersteller: Kingsoft

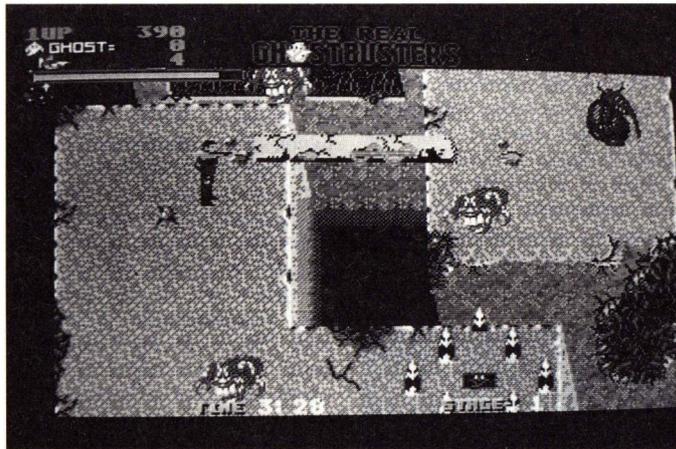
Info: Kingsoft



cbo

The Real Ghostbusters

Die Geisterjäger sind bei uns durch den großen Kinoerfolg bekannt geworden. Bei "The Real Ghostbusters" handelt es sich um eine Automatenumsetzung des lustigen Filmstoffs. Es gilt, Geister einzufangen und aufzubewahren. Das Handwerkszeug dazu schleppen Sie mit sich herum: ein Gewehr und einen Protonenstrahler. Beide Waffen werden per Feuerknopf ausgelöst: Mit einem kurzen Druck schießen Sie alle Gestalten ab. Daraufhin erscheinen die Geister, die mit einem langen Knopfdruck in den Rucksack gesaugt werden. Der Protonenstrahler muß wieder aufgetankt werden. Dazu gibt es Bonussymbole wie Ölfässer, Heizkessel oder auch Schubkarren. Man kann alleine oder zu zweit auf Geisterjagd gehen. Im Zwei-Spieler-Modus laufen beide Spieler gleichzeitig auf dem Bildschirm herum. Ähnlich wie bei "Gauntlet" sieht man die Action aus der Vogelperspektive. Leider ist "The Real Ghostbusters" ziemlich



The Real Ghostbusters ist das Spiel zur erfolgreichen englischen Fernsehserie.

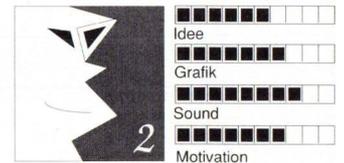
schwer. Ein Schuß der Gestalten nur knapp am Kopf des Helden vorbei, und schon ist ein Bildschirmleben futsch! Diese übertriebene Schwierigkeit hätte nicht sein müssen. Dafür wird dem Auge aber Entzückendes geboten: Zehn

abwechslungsreiche Levels, eine Vielzahl an Monstern, wie beispielsweise Werwölfe, elefantenähnliche Wesen, häßliche grüne Fratzen, Würmer oder das typische Bettlakengespenst bevölkern die Spielfelder. Dagegen

sehen die Spielerfiguren etwas kläglich aus: Gewehr im Anschlag mit grober Grafik. Genauso, wie sie aussehen, gehen die Helden auch: mit merkwürdig breiten und abgehackten Bewegungen. Dagegen ist die Musik ein Genuß: eine gute Umsetzung von Ray Parker Jr. "Ghostbusters". Sieht man von dem übertrieben hohen Schwierigkeitsgrad und der teilweise schlampigen Grafik beim Helden ab, bleibt ein zufriedenstellendes Spiel übrig.

cbo

Hersteller: Activision
Info: Ariolasoft



In einem von Bösewichtern verseuchten Plattformland muß "Skweek" (gesprochen: "Squiek") wieder lebensfreundliche Verhältnisse schaffen. Dazu läuft er über die verseuchten blauen und grauen Felder, worauf sich diese in ein fröhliches Rosa verwandeln. Auf jeder der 99 Plattformen ist eine andere Anzahl von Feldern einzufärben. Das geht natürlich nicht einfach so: Da wäre zuerst das Zeitlimit. Der kleine gelbe Held hat für jedes Level 99 Sekunden Zeit, sonst verliert er ein Bildschirmleben. Zusätzlich erschweren sechs bewegliche Plüschmonster in Bonbonfarben die reibungslose Malerarbeit. Man darf sich nicht von einem Widersacher erwischen lassen. Das kostet ebenfalls ein Bildschirmleben. Als letztes sorgt der Aufbau der Plattformen für einige Schwierigkeiten: Mauern und Löcher behindern Skweek. Ein "Boom"-Feld läßt die umliegenden Felder verschwinden, und ein paar stationäre Monster wollen den Spieler gern verspeisen. Schlechte Aussichten? Mitnichten! Eine Vielzahl von Boni warten nur darauf, von Skweek eingesammelt zu werden. Neben Punktegutschriften in Form von Eis und Sanduhren, die mehr Zeit bringen, gibt es eine Reihe von anderen Bonussymbolen. Durch

SKWEEK Plüschmonster und pinke Plattformen



SKWEEK, ein kleines Plüschmonster, muß Plattformen rosa färben.

ein "Exit"-Feld gelangt man sofort ins nächste Level. Zum Schluß kann man gleich fünf(!) neue Leben erwerben, indem man vier verschiedenfarbige Teddybären einsammelt. Fast alle Möglichkeiten

werden in einer sehr gut gemachten Demo (Dauer: knapp 9 Minuten!) vorgeführt. Da kommt echte Spiellust auf. Und wenn man sich dann endlich wieder von seinem Computer getrennt hat, sucht man

womöglich nach einem Eimer rosa Farbe, um die Bude zu "verschönern". "Skweek" macht unglaublich viel Spaß. Dazu kommt, daß der Highscore abgespeichert wird. Auch für Wettspiele ist gesorgt - zwei Spieler können nacheinander antreten. Der eine kann dabei aus den Fehlern des anderen lernen. Die Plattformen sind größer als ein Bildschirm. Wenn man nicht vorsichtig genug ist, rennt man vielleicht geradewegs in ein Loch oder ein Monster. Der Bildschirm wird dabei senkrecht gescrollt. Die Beschaffenheit der Plattformen trägt entscheidend zum Schwierigkeitsgrad bei, der von sehr einfach bis ziemlich schwer reicht - und dies in einer ausgewogenen Mischung. "Skweek" bricht zwar keine Grafik- oder Soundrekorde, dafür weiß es durch ungeheuren Spielwitz und hohe Motivation zu begeistern.

cbo

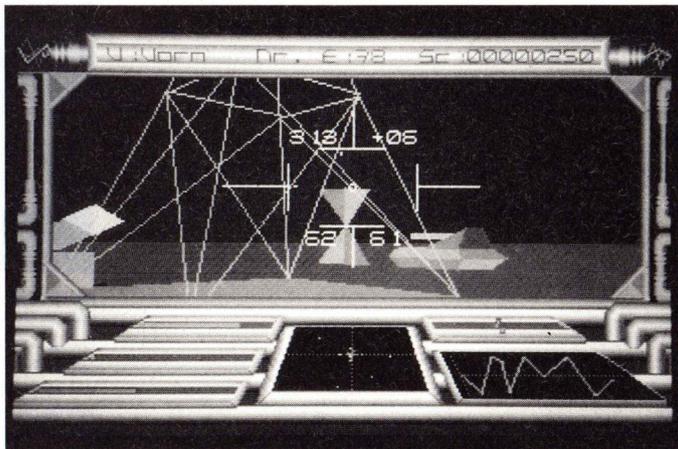
Hersteller: Loriciels
Info: Profisoft



Voyager ist ein Spiel mit historischem Hintergrund. 1977 wurde der Satellit Voyager II auf seine Reise ins Universum geschickt. Vorbei an den äußeren Planeten unseres Sonnensystems wird er demnächst dessen Grenzen erreichen und von dort in die unendlichen Weiten des Alls davontrudeln. Er beinhaltet eine Nachricht an andere intelligente Lebensformen mit einer "Einladung", uns einen freundlichen Besuch abzustatten. Daneben befinden sich noch Artefakte, eine kurze technische und historische Beschreibung der Erde und die Beschreibung unserer Position.

VOYAGER spekuliert nun mit dem Erfolgsfall(?). Im Jahr 2030 unserer Zeitrechnung entdeckt Non, ein weiblicher Scout der Roxiz, den Satelliten. Der Herrscher der Roxiz entschließt sich auf Ihren Vorschlag hin, die "herzliche Einladung" anzunehmen und den kleinen Planeten heimzusuchen. Der Besuch geht anders aus als erwartet - die Roxiz erobern die Erde und zerstören sämtliche Raumschiffe (die Erde hat sich inzwischen auch weiterentwickelt).die letzte Chance ist Luke Snayle der sich mit einem voll ausgerüsteten Kampfraumer auf dem Weg zurück ins heimatliche Sonnensystem befindet, als die Besucher zudringlich werden. Als er die Ro-

VOYAGER



VOYAGER bietet weit mehr als beispielsweise STARGLIDER II.

xizianer entdeckt, die sich gerade auf den äußeren Planeten des Sonnensystems häuslich niederlassen, wird ihm schnell klar, daß sein Schiff die einzige Rettung für Mutter Erde darstellt. So beschließt er, die zehn Monde des Saturn anzugreifen, auf denen die Aliens ihre Stützpunkte installiert haben.

Der graphische Aufbau des Spiels

ist ähnlich dem von Starglider II, aber VOYAGER geht wesentlich weiter. Die Oberflächen der 3D-Objekte sind mit mehr Feinheiten ausgestattet und die Landschaften sind wesentlich komplexer. Man arbeitet sich von Mond zu Mond immer weiter in die feindliche Verteidigungszone hinein, indem man die gegnerischen Raumschiffe deaktiviert (genau!). Erschwerend

kommt hinzu, daß es innerhalb einer bestimmten Zeit auch noch ein Portal zu finden gilt. Die Passage zwischen den einzelnen Monden ist nicht geradlinig, und man muß geschickt navigieren, um nicht an den Wänden die eigenen Schutzschirme zu zerstören. Jeder einzelne Mond stellt große Anforderungen an Strategie und Geschick. Selbst die Kameras können zweckentfremdet, aber nutzbringend eingesetzt werden. Wenn man VOYAGER mit STARGLIDER II vergleicht, fällt nicht nur auf, daß die einzelnen Objekte mehr Details bieten, sondern auch, daß es gelungen ist, das ganze Spiel mit mehr Handlung zu füllen. Es gibt wesentlich mehr zu tun und zu sehen. Der Sound ist, wie bei allen Spielen dieser Art, begrenzt. Was aber an Sound geboten wird - Titelmusik etc. -, ist ansprechend. Insgesamt ist VOYAGER weit davon entfernt "noch ein Schießspiel" zu sein.

'Travel to your nearest shop and beam up a copy'.

ddf



INSERENTENVERZEICHNIS

A.P.S.	47	IDEE - SOFT	13
ALPHA TEAM	47	KUPKE	132
AMIGAOBERLAND	29	MAXON	9, 17, 21, 55, 111
ABC - SOFT	61	MVC	67
ALCOMP	13, 131	NOVACOM	50
COMPEDO	34	OMEGA - VIDEO	26
CIK - COMPUTERTECHNIK	109	PBC - BIET	47
COMPI MATE	31	PENUGAOW	113
CWTG	26	PRINT - TECHNIK	47
DIEZMANN	39	RAINBOW - DATA	26
DONAU - SOFT	67	RHEIN - MAIN - SOFT	61
DREWS	31	SPACE - SOFT	47
DTM	2	SOFTWARE 2000	67
DFU - SHOP	50	SCS	61
EDOTRONIK	109	STATE	50
FISCHER	61	VIDEO - LOFT	61, 109
FE - GRAFIK + VIDEO	113	VORTEX	25
FSE	50		
HEIM	12, 59, 103		

KICKSTART PU

Auch in diesem Monat gibt es wieder zwei randvolle Disketten mit MODULARROUTINEN, die die Programmierung erleichtern sollen. Für Pascal-Fans und dabei besonders für Einsteiger gibt es ein komplettes PASCAL-SYSTEM mit Editor, Compiler, Assembler, Linker und Bibliotheken (Libraries). Besonders gut gemacht ist auch das BUNDESLIGAVERWALTUNGSPROGRAMM, das durch gute Benutzerführung, Grafik und einfache Bedienung glänzt. Als Dateiverwaltung ist HYPERADRESS oder für komplexere Aufgaben HYPERBASE zu empfehlen. Zwei sehr interessante und spannende Spiele und je eine Grafik- und Musikediskette runden die Neuzugänge der Sammlung ab.

Bis zum nächsten Mal

Markus Nerding

KICK 175: ANWENDUNGEN



BUNDESLIGA: Ein sehr schönes und komfortables Programm zur Verwaltung der Bundesliga. Die Eingabe der Spielergebnisse erfolgt ausschließlich mit der Maus. Das Programm errechnet daraus sofort die Tabelle mit Punkt- und Torverhältnissen sowie den Heim- und Auswärtsbilanzen. Außerdem kann man sich den Saisonverlauf als Grafik ausgeben lassen. Die Tabelle wird zudem mit Emblemen der Vereine dargestellt. Mit umfangreicher deutscher Anleitung. Sehr komfortabel und grafisch ansprechend.

DIE NEUHEITEN

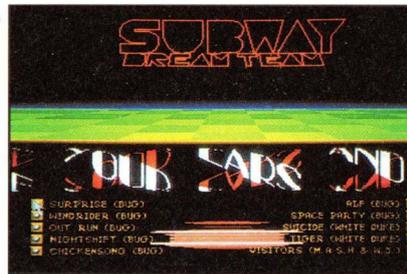
KICK 171: SPIELE



BILLARD: ein gutes Billardspiel mit schöner Grafik und mit vielen Möglichkeiten. Es gibt drei Spielvarianten (Carambolage, Dreiband, Pool) und einen Trainermodus. Im Trainermodus können zudem die Bälle beliebig platziert werden. Die Stoßkraft wird mit einem Schieberegler eingestellt, danach erfolgt die Platzierung des Treffpunktes und die Justierung der Stoßrichtung mittels Gummiband. Wenn diese erledigt ist, fehlt nur noch der auslösende Klick, und die Kugeln kommen in Bewegung. Mit deutscher Anleitung und Regeln, ein oder zwei Spieler. Autoren: Axel Illenburg, Gerold Schickram.

komme ich nur in die HiScore-Liste") sind die Mindestanforderungen an den Spieler. Autoren: Ottfried Schmitt, Ulrich Muehl

KICK 173: SOUNDS



WORLD'S WAVE II: eine fantastische Sound-Diskette, erstellt mit Soundtracker. Auch grafisch wird einiges geboten. Verantwortlich für diese Demo ist das Subway Dream Team, denen dafür ein Lob ausgesprochen werden muß. 10 Sounds mit langer Spielzeit, autobootend.

KICK 176: ANWENDUNGEN

HYPERADRESS: eine einfache Adreßdatei mit vorgegebener Eingabemaske. Einfaches Eingeben, Anzeigen und Suchen der Datensätze mit allen Kriterien. Mausbedienung, deutsche Anleitung.

HYPERBASE V1.6: eine Shareware-Datenbank mit Menüführung und Mausbedienung. Mit zwei Beispieldateien. Leistungsfähig, flexibel und einfach zu bedienen. Mit ausführlicher Anleitung.

KICK 177: MODULA (AMOK 17)

MEGAWB: Endlich keine überfüllte Workbench mehr - MegaWB vergrößert die BitMap der Workbench auf eine beliebige Größe (z.B. 1024 x 512). Damit weiterhin die gesamte Workbench benutzt werden kann, scrollt sie bei Randannäherung sanft weiter.

IFF2IMAGE: wandelt IFF-Bilder in einfach nachzuladende Image-Dateien um.

ILBMHANDLER: enthält Prozeduren zum Laden, Speichern, Colorcycling und Ein- und Ausblenden von Screens.

M2TEST: zeigt Fehler des M2-Compilers bei der Variablendeklaration.

GETIT-ZEIGER: erlaubt einfaches Verändern der Workbench-Farben und des Mauszeigers.

KICK 172: SPIELE



BLIZZARD: Ein Wahnsinnsballerspiel liegt mit Blizzard vor. Mit einem Raumschiff durchfliegt man eine sauber horizontal scrollende Szene und harrt der Dinge, die da kommen. Und es kommt einiges: bildfüllende Angreiferwellen verschiedenster Formen, hopsende Kugeln und bewegte Quader. Ein blitzschneller Feuerfinger, gute Reaktionen, stahlharte Nerven und Durchhaltevermögen ("wie

KICK 174: GRAFIK



FROGMOVIE: Eine starke Grafikedemo liegt mit dem "Froschfilm" vor. Zuerst sitzt der knallgrüne Kerl augenrollend und scheinbar friedlich auf seinem Ast, doch wenn die Fliege erscheint, wird er plötzlich aktiv ...

KICK 178: MODULA (AMOK 18)

PROFILE: erweitert Modula-Quelltext um Informationen über den Wertebereich von Variablen, Anzahl der Schleifendurchläufe und der Aufrufe von Prozeduren.

POPUP-MENU: Modul zur Unterstützung von PopUp-Menüs.

IFFLIB: IFF-Library zum Laden und Speichern aller Arten von IFF-Dateien.

KICK 179: PASCAL

PCQ-PASCAL ist ein guter Pascal-Compiler, der mit dem bekannten Assembler/Linker-Gespann (A68K/BLINK) zusammenarbeitet. Ein

BLIC DOMAIN

leicht bedienbarer Editor (HED) und die Bibliotheken SMALL.LIB und PCQ.LIB (mit Source in Assembler) runden das System ab, das damit als komplettes Pascal-System vorliegt. Es ermöglicht sogar die Nutzung amigaspezifischer Funktionen über schon vorhandene Include-Dateien (z.B.

```

Mockbench Screen [Virus? 3] [Chip:327] [Fast:1702] [Time:28:28:31]
New 4:1
02. execute make.snow
PCQ Compiler 1.8 (February 1, 1989)
Copyright © 1989 Patrick Quaid. All rights reserved.
Compiling Snowlake
There were no errors.
68000 Assembler - version 1.2 (July 11, 1988)
Copyright © 1985 by Brian R. Anderson
AmigaDOS conversion copyright © 1987 by Charlie Gibbs.
Assembling t:snow.asm
PASS 1
PASS 2
End of assembly - no errors were found.
Heap usage (bytes) - 60786, 58
Total block sizes: 484 code, 89 data, 194 BSS
BLINK - Version 6.7 - 15 October 1988
Copyright © 1986 The Software Guild. All Rights Reserved.
235 Trillingham Lane, Cary NC 27511 - BBS: (919)-471-6436
BLINK Complete - Maximum code size = 6576 (800001908) bytes
07. snow 4
    
```

DOS, Exec, Intuition, Graphics, Math, Ports usw.), die noch erweiterbar sind. Über Batchdateien ist das System sehr einfach zu bedienen, zudem helfen viele Beispielprogramme beim schnellen Einstieg. Mit umfangreicher Anleitung und deutscher Kurzanleitung für die ersten Schritte. Läuft auch mit 512k Speicher! Für Anfänger und Pascal-Fans sehr empfehlenswert!

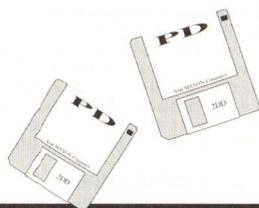
KICK 180: UTILITIES

VIEWER: ein Programm zum Anzeigen von IFF-Grafiken. Vollständig in Assembler und noch nicht einmal 1k lang.

PYRO: der Bildschirmschoner, der ein Feuerwerk startet, wenn Sie Ihren AMIGA einige Zeit unberührt lassen.

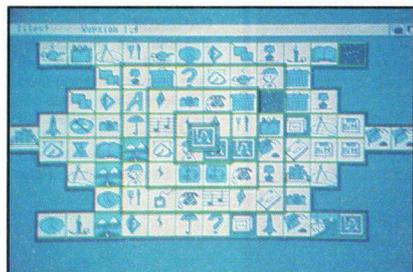
TRANS: ein Programm, das SEKA-Programme patcht, damit sie problemlos mit dem KICKASS eingelenen werden können.

SMALL.LIB: die kleine Bibliothek, die von einigen Assemblern und Compilern verwendet wird.



ANRUF GENÜGT
06196 / 481811

X KICK 164: SPIELE



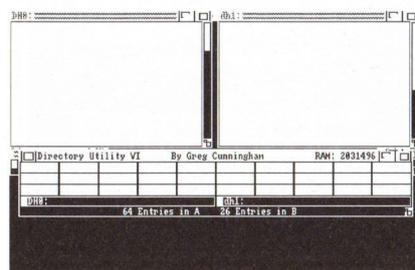
TILES: Eine schöne Version des unter dem Namen Shanghai bekannt gewordenen Spiels. Aus einem Steinestapel müssen solche mit gleichen

Symbolen gefunden werden, die gleichzeitig freiliegen, also nicht von anderen Steinen umgeben sind. Wird es gelingen, den ganzen Stapel aufzulösen? Ein spannendes Taktikspiel für alle Spielefans.

X KICK 165: SOUNDS

SOUND QUEST I: Einige sehr gute Sounds (Blue Monday, Rocking, Bomb the Kaes, Slice Me Nice, Tron, Hunters Moon, That's all), die von ACS mit dem Soundtracker erstellt wurden.

X KICK 166: SIMULATIONEN



Diese Version ist auch unter dem Namen "Krieg der Kerne" bekannt. Dabei bekämpfen sich zwei Programme innerhalb des Rechners. Sie testen Speicherzellen, umkreisen den Gegner oder greifen ihn direkt an. Komplettes System zum Erstellen der Programme (Assemblersprache REDCODE) und Abwickeln der Simulation (MARS). Mit umfangreichen deutschen Anleitungen (200 KB) und 12 Beispielprogrammen. CORE WARS ist Shareware. Autoren: Roger Meier und Patric Hofmann von Unicorn Systems.

KICK 167: MODULA (AMOK 15)

Kurve: leistungsfähiges Kurvendiskussionsprogramm (Ableitungen, Integrale, Nullstellen, Extrema, Wendepunkte usw.)

Detach: endpositioniert Icons

Originale: Rückübersetzung einiger M2Amiga-Module

Seafarer's Manual: Programme und Lösungen

Calc: ähnlich EVAL von WB 1.3

KICK 168: MODULA (AMOK 16)

M2MAKE: unverzichtbares Make-Utility für jeden ernsthaften Modula-Programmierer.

MODLIST: formatierter Ausdruck von Modula-Listings mit hervorgehobenen Schlüsselwörtern.

ROTATEIFF: rotiert ein IFF-Bild um die Horizontale.

MULTISELECT: Mehrfachauswahl mit dem rechten Mausknopf anstelle von Shift.

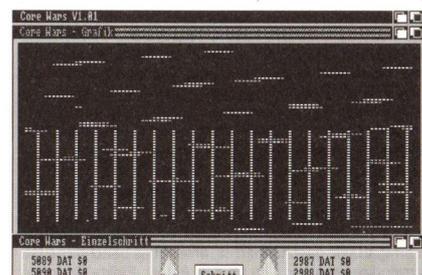
Außerdem Updates von: MemSystem V1.3, AVL V1.2, LIST V1.1, QUEUE V1.2

KICK 169: UTILITIES

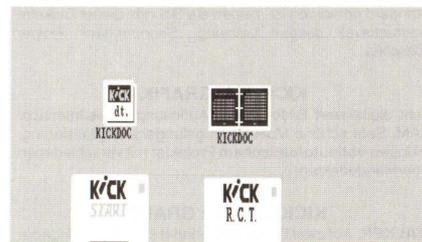
IFF-STANDARD: Dies ist die neue Diskette zum IFF (Interface File Format)-Standard von ECA und COMMODORE, die anlässlich der DevCon 89 veröffentlicht wurde. Hier sind die Standards (ILBM, FTXT, SMUS, 8SVX) und neue Kennungen spezieller Anwendungen (ANIM, MIDI, WORD, RGB4 u.v.a.) festgelegt, an die sich jeder Softwareentwickler halten sollte, um kompatible Programme zu erhalten. Mit den neuesten Versionen von SHOWILBM, DISPLAY, SCREENSAVE und MORE.

KICK 170: UTILITIES

DISKPERF: Ein Benchmark-Programm für Diskettenlaufwerke, Festplatten und andere Speichermedien. Enthält umfangreiche Tests für Schreib- und Lesezugriffe und gibt eine Liste der Daten auf. Einige Daten für verschiedene Medien und Rechner (VAX unter UNIX) sind bereits enthalten. Das neue Fast File-System der WB 1.3 wird berücksichtigt. Source in C.



DIRUTIL VI: Ein sehr komfortabler und leistungsfähiger Ersatz für das CLI. Alle wichtigen Befehle und Funktionen können mit der Maus ausgeführt werden.



KEYMAPED: Ein Editor für die Tastaturbelegung der SetMap-Dateien. Sehr komfortabel und einfach mit der Maus zu bedienen. Mit Source in C.

DIFFDIR: Vergleicht zwei Directories. Sehr nützlich für einen Aktualitätscheck. Source in C.



Fortsetzung umseitig !

KICK 163-141

KICK 163: SPIELE

WORLD: Ein Textadventure ähnlich den Infocom-Adventures Planetfall und Starcross. Guter Wortschatz (ca. 500) und umfangreiche Texte (130 KB). Mit Source-Code in C.

KICK 162: SPIELE

PACMAN87: eine gelungene Pacman-Variante mit einigen neuen Features wie z.B. Feuerfallen, Messern, Elektrofeldern und einiges mehr. Drei Schwierigkeitsgrade, HighScore-Tabelle, Sound, Joysticksteuerung.

KICK 161: SPIELE

MORIA: Irren Sie gerne durch dunkle Verliese, und wollen Sie unheimliche Mächte und ihre Wesen bekämpfen, dann ist dieses Spiel das Richtige. Zu Beginn müssen Sie im Dorf Hilfsmittel, Waffen und magische Sprüche erwerben, und dann können Sie sich in das Abenteuer stürzen. Mit schöner Grafik, einfacher Benutzerführung (Menüs), vielen Optionen, die die Spielbarkeit erhöhen und dem AMIGA gerecht werden. Benötigt 1MB Speicher.

KICK 160: UTILITIES

VSCREEN: Sensationell, denn dieses Programm vergrößert den WB-Screen auf einen beliebigen Wert (begrenzt vom ChipMEM). Wenn man dann die Maus an den Rand bewegt, scrollt der Screen sanft weiter und weiter.

KLICKIT: Dieses Programm ermöglicht es, CLI-Befehle und Batch-Dateien von der Workbench aus zu starten. KLICKIT generiert dazu ein Startprogramm, das auf Wunsch ein Fenster für eine Eingabe/Ausgabe öffnet. Einfache Maussteuerung.
Autor: Udo Marquardt.

WICON: Schafft Ordnung auf dem Bildschirm. Jedes Fenster kann von nun an auf Knopfdruck bis auf die Menüleiste verkleinert und entsprechend auch wieder vergrößert werden.

KICK 159: UTILITIES

HERMES' ICONLAB V1.1: Das beste Hilfsprogramm zum Bearbeiten von Icons. Es erlaubt das Wählen von 'Icon Type', 'Default Tool', 'Tool Types' und des Stacks. Außerdem können Icons zusammengefügt, getrennt und aus Brushes (z.B. von Deluxe Paint) erzeugt werden. Sehr komfortable Bedienung.

KICK 158: ANWENDUNGEN

KLECKS V1.0: Ein Malprogramm mit vielen Funktionen, guten Fähigkeiten und komfortabler Bedienung. Obwohl es in AmigaBasic geschrieben ist, erreicht es beachtliche Geschwindigkeiten (ein Lob an den Autor). Einige Optionen: Dateiauswahlbox, Füllmustereditor, Pal, Hires, Overscan, Fonts, Pinsel, Vergrößern, Drehen, Spiegeln, Kopieren usw. Benötigt mind. 1 MB. Autor: Thomas Auf der StraÙe

KICK 157: SOUNDS

Nachdem der **Soundtracker** unter Freaks schon lange zum Standard geworden ist, zeigen die Sounds dieser Diskette eindrucksvoll dessen Leistung. *Soundmaster: Holger Blaschka*

KICK 156: GRAFIK

Acht digitalisierte Bilder in der Auflösung Lores/Interlace/HAM. Sehr schöne Motive und gelungene Digitalisierung. Inklusive **vollautomatischem** Projektor mit verschiedenen Überblendeffekten(!).

KICK 154/155: GRAFIK

WALKER: Auf zwei Disketten befindet sich diese Megademo. Sie benötigt mindestens 2MB Speicher und zeigt einen Walker (bekannt aus 'Krieg der Sterne'), der vor einem AMIGA umhermarschiert. Phantastische Demo mit sehr guter Animation und bombastischem Sound.

KICK 153: SPIELE

TETRIS: Herabfallende Formen müssen so gedreht werden, daß sie zusammenpassen. Trotz dieser einfachen Spielidee ist es eines der besten Spiele für den AMIGA. Wer einmal damit angefangen hat, den wird das Spiel so schnell nicht mehr loslassen.

FIGHT: Eine Variante des Pong-Spiels. Zweispielermodus, Source in C (interessante Effekte für eigene Programme). Autor: Matthias Kühn

KICK 152: SPIELE

CASTLE OF DOOM: Ort der Handlung ist Sherwood Forest in England im 12. Jahrhundert, und die Aufgabe ist an die Robin Hood-Sage angelehnt. Ein schönes Adventure-Spiel mit Grafik. Für Anfänger geeignet, mausgesteuert, englisch.

KICK 151: SPIELE

BROKER 1.0: Das Handeln mit Aktien ist die Aufgabe der bis zu vier Spieler. Beim Kaufen und Verkaufen sind Bonus, Steuern, Optionen, Dividenden, Zuschießen, Kontozinsen und Kredite zu berücksichtigen. Spiel in zwei Versionen (deutsch und englisch), mausgesteuert.
Autor: Andreas Page

KICK 150: UTILITIES

TASKDRIVE: ermöglicht ein bequemes Ansehen und Ändern der Abarbeitungsprioritäten jedes Prozesses/Tasks im System. Eigener Screen, Source in Lattice C. Autor: Rudolf Gerlinger (Österreich)

IMAGCALC: ein Taschenrechner, der mit komplexen Zahlen arbeitet und deshalb z.B. die Wurzel einer negativen Zahl in Imaginär- und Realteil zerlegt. Autor: Peter Schaefer

Aus Platzgründen können leider nicht mehr alle PD-Disketten aufgeführt werden. Sollten Sie keine ältere KICKSTART zur Hand haben, dann können Sie die Liste auch bei uns anfordern. Senden Sie dafür einen adressierten und mit DM 1.- frankierten DIN A5 Rückumschlag an folgende Adresse:

MAXON Computer GmbH
KICKSTART PD-Liste
Postfach 55 69
6236 Eschborn

Machen Sie mit!

Möchten auch Sie selbstgeschriebene Programme der Allgemeinheit zur Verfügung stellen, so schicken Sie sie uns einfach zu (bitte mit Dokumentation auf Diskette!).

Als kleine Anerkennung können Sie sich dafür 5 Disketten aus unserem PD-Service auswählen.

MAXON-Computer GmbH
KICKSTART-Redaktion
PD-Einsendung
IndustriestraÙe 26
6236 Eschborn

VERSANDBEDINGUNGEN:

Um einen schnellen und problemlosen Versand zu gewährleisten, beachten Sie bitte folgende Punkte:

- Bestellungen per Nachnahme oder Vorkauskasse
- Für jede Diskette ergibt sich ein Unkostenbeitrag von DM 8.-
- Pro Sendung kommt ein Versandkostenbetrag (für Porto und Verpackung) von DM 5.- (Ausland DM 10.-) hinzu.

Bei einer Bestellung von 5 oder mehr Disketten entfällt der Versandkostenbeitrag!

- Bei Nachnahme zuzüglich 3.70 DM

Anschrift:
MAXON Computer GmbH
KICKSTART PD
Postfach 55 69
6236 Eschborn

Die Diskettenbestellung kann auch telefonisch erfolgen. Der Versand erfolgt dann per Nachnahme.

Tel.: 06196 / 48 18 11
(Mo.-Fr. von 9⁰⁰-13⁰⁰ u. 14⁰⁰-17⁰⁰ Uhr)

CODE: ein Chiffrier- und Dechiffrier-Programm mit Paßwortschutz. Autor: K.J. Wolf

SHC (Screen-Hard-Copy): Ein sehr schnelles (Assembler) Druckprogramm für eine Bitplane eines Screens. Für STAR LC-10, NL-10, EPSON LX-800 und Kompatibel. Mit Source. Autor: Karsten Kalweit

KICK 149: MODULA (AMOK 14)

ARP: M2AMIGA-Implementation der arp.library, die das BCPL-DOS durch leistungsfähigere Befehle ersetzt. **TIMERSUPPORT:** erleichtert die Benutzung des Timer-Devised

KICK 148: MODULA (AMOK 13)

COMPCONTROL: erleichtert die Bedienung von Editor, Compiler und Linker. **ROWS:** generischer Datentyp, während der Laufzeit dimensionierbare Arrays. **WITH:** ein neuer CLI-Befehl. **FLIPMOUSE:** invertiert die Richtung der Mausbewegung. **WBShadow:** Schatten für die Workbench. **WPIC:** bringt ein Bild auf die Workbench.

KICK 147: ANWENDUNGEN

M.S.TEXT: Eine leistungsfähige Textverarbeitung mit den wichtigsten Funktionen. Einfache Bedienung über Menüs oder Tastensequenzen, Blockfunktionen (markieren, kopieren, löschen, laden, speichern, drucken), eine Dateiauswahlbox und die schnelle Scrollgeschwindigkeit zeichnen sind die wichtigsten Merkmale. Quellcode in AZTEC C, komplett in Deutsch. Autor: Martin Stepler

KICK 146: ANWENDUNGEN

NOSCOM-MATCH 1.1: Hierbei handelt es sich um ein Auswertungsprogramm für Wettkämpfe des offiziellen nationalen Schießsports. Es ist sehr komfortabel und fast gänzlich mit der Maus zu bedienen. Es erleichtert Darstellung, Auswertung und Ausdruck der Ergebnislisten. Eine Version 1.2 ist in Vorbereitung und wird vom Autor angeboten. Komplett in Deutsch, mit ausführlicher Dokumentation; benötigt 1MB RAM. Autor: Norbert Süßdorf

KICK 145: ANWENDUNGEN

GIROMAN V3.00: Ein sehr leistungsfähiges Girokontoverwaltungsprogramm, das vom Autor schon seit 1980 auf einem PET eingesetzt wird. Das Programm ist sehr komfortabel und einfach zu bedienen. Zu den wichtigsten Funktionen zählen: Einträge suchen, Kontostand eines bestimmten Tages abfragen, monatliche Festposten (Daueraufträge usw.) erledigen, Ausdruck mit Saldoangabe und manuelle Buchungen (hinzufügen, ändern, löschen). Mit Anleitung, komplett in Deutsch, Source in Aztec C. Autor: Sönke Netzband

KICK 144: GRAFIK

AMIGAPAIN V1.5: ein deutsches Malprogramm mit vielen Optionen (Freihand, Punkte, Linie, Rechteck, Block, Fläche, Kreis, Füllen, Radiergummi, Text, Spraydose, Füllmuster und Lupe). Es arbeitet im Lores-Modus mit 320x200x5 und 32 Farben. Auf der Diskette ist das Listing in AmigaBASIC (etwa 1240 Programmzeilen) enthalten.

KICK 143: MUSIK

WIZARD OF SOUND: ein Musikprogramm zum Erstellen von Musikstücken nach Notenvorlage oder zum Komponieren. Einige Funktionen im Überblick: bis zu 8 Instrumente gleichzeitig, 4 Stimmen, verarbeitet gesampelte Instrumente im IFF-Format (8SVX / 5 Oktaven), Inplay-Modus (Spielen mit Begleitmusik), Spezialfunktionen (Crescendo, Accelerando, Ritardand, Fermate, Triole usw.), autom. Taktorganisation, Ausdruck der Notation, Player-Programm (läuft im Hintergrund), multitaskingfähig, komplette Maussteuerung, Help-Funktion, dt. Anleitung. Autor: Jörg Schließer

KICK 142: SPIELE

GAMBLER: Ein allgemeines Spielsystem, das Strategie-spiele wie Dame, Vier gewinnt, Tic-Tac-Toe u.ä. spielen kann. Mit diesem Programm errang Guido Appenzeller einen Preis bei der Regionalauscheidung von 'Jugend forscht' 1988/89.

BALLY: Das Spielprinzip ist sicherlich bekannt - 75% einer Fläche müssen umrandet und damit ausgefüllt werden, nur darf man sich nicht von den herumschwirrenden Punkten erwischen lassen. Sehr schnell, Joystick erforderlich! Autor: Oliver Wagner

KICK 141: SPIELE

PARANOID: eine Breakout-Variante mit High-Score-Liste und Level-Editor. Mit dem Leveleditor kann man sich auf einfache Weise beliebige Levels gestalten. Auf der Diskette sind bereits 24 Levels vorhanden. Mit gutem Sound. Autor: Felix Wente



KICK 140-101

KICK 140: UTILITIES

RAMCOPY: Kopierprogramm für ein Laufwerk und mind. 1MB Speicher
DISKSALV: rettet gelöschte Dateien
SPUDCLOCK: Uhrzeitanzeige
MACHII: Maustool, Hotkeys, Popcli usw.
MULTICALC: RPN-Taschenrechner
MRPRINT: Druckprogramm für Dateien, mit vielen Optionen
AFTERM: Terminalprogramm
MFIX: patcht das Kopierprogramm MARAUDER und steigert damit die Geschwindigkeit erheblich

KICK 138 & 139: UTILITIES

Auf dieser und der KICKPD 139 befinden sich die Sourcecodes der meisten Programme von Matt Dillion, der sicherlich jedem PD-Anwender bekannt ist. Im einzelnen sind dies: CONFIG V1.00, CLOCK V1.00, DME V1.31, DMOUSE V1.10, BACKUP V2.01, SUPLIB, LIBREF, DRES V1, DASM V2.11, FTOHEX, FILES V1.2, SHELL V2.10, FINDIT V1.00, LIBS V1.00, SCAT, ADDCR, REMCR, CMP.

KICK 137: SPRACHEN

Für alle Programmiersprachenbegeisterten ist hier eine kleine Fundgrube. **SCHEME:** ein Dialekt ähnlich Lisp, **PLOT:** eine Umsetzung dieser Sprache für den AMIGA, **ADA:** Syntaxkontrolle für ADA, mit YACC-Source, **PERL:** Interpreter

KICK 136: ANWENDUNGEN

AMIGASPELL: Wörterbuch (Wortschatz: 9085 engl. Wörter) zum Korrigieren von (ASCII)Texten. Falsche Wörter können ausgedruckt, im Text korrigiert (der Satzzusammenhang wird angezeigt!) oder übernommen werden. Schnell (Assembler) und speichereffektiv. Es kann leicht ein deutsches oder hessisches Wörterbuch erstellt werden. **NAG:** der absolute, blinkende, schreiende, singende Terminkalender, damit Sie nie einen Geburtstag oder eine Fete oder vergessen. Sehr komfortable Eingabe, viele Möglichkeiten, immer greifbar.
HP11: HP11C-Calculator mit allen Feinheiten.

KICK 135: GRAFIK

MANDEL 2.1: Mandelbrotgenerator für repetitive Muster. Source in AmigaBASIC. Autor: Michael

KICK 134: BILDER

Viele gute Grafiken (SexyRobot, Roland SH-101, Sailboat, Max, Ferrari GTO, Dali, ST), Cartoons (Batman, Garfield) und digitalisierte Bilder (Supercollage von Marilyn, Mona Lisa, Alien).

KICK 133: GRAFIK-SHOW

CRAZY-MOTOR-SHOW: gut gezeichnete Sportwagen und eigene Konstruktionen mit Angabe der technische Daten und interessanten Hintergrundinformationen. Mit digitalisierter Musik. 1MB Speicher erforderlich. *Autofan: I.K.*

KICK 132: SOUNDS

TGB SOUND DISK NO.1: Fünf Musikstücke (erstellt mit Soundtracker) werden vor dem Hintergrund von HAM-Bildern abgespielt. *Komponist: Thomas-Georg Brügge-mann.*

KICK 131: SPIELE

TENNIS: Tennisspiel für zwei Spieler mit Trainingsmodus, Grundlinien- und Netzspiel. Die Schlagstärke und Richtung können mit dem Joystick bestimmt werden. Erfordert 1MB Speicher. Autor: Christian Dreke.

KICK 130: ANWENDUNGEN

SUPERLIGA: Fußballergebnisverwaltung mit 577 Mannschaften aus 23 Ländern und 34 Ligen. Über 10000 Ergebnisse können gespeichert werden. Das Eingeben gestaltet sich sehr einfach. Aus den bisherigen Ergebnissen kann eine Prognose (Regressionsanalyse) für kommende Spiel und die Meisterschaft erstellt werden. Autor: Rolf Morlock

KICK 129: SOUNDS

Neue Kompositionen von Andreas Starr aus Gronau, erstellt mit Sonix. Effektvolle Stereostücke mit Player.

KICK 128: SOUNDS

Scores, Drumkits und Soundfiles für Dynamic Drums von New Wave Software.

KICK 127: GRAFIK

Mandelbrot-Show II: dies ist der zweite Teil der Reise durch die Welt der Mandelbrotgrafiken. Autor: R. Beck

KICK 126: GRAFIK

Mandelbrot-Show I: eine sehr informative Reise durch die Welt der Mandelbrotgrafiken. Jedes Bild wird ausführlich erläutert und das jeweils darauffolgende ist eine Vergrößerung eines Ausschnitts davon. Mit Mandelbrotgenerator MSE. Autor: R. Beck

KICK 125: GRAFIK

SLIDESHOW: Komfortables Diashow-Programm für DPaint-Bilder (320x200). Alle Bilder des aktuellen Direktories werden nacheinander auf Mausclick angezeigt. **DARK:** Copper-Demo, die viele Elemente eines Ballerspiels enthält. C-Source. **GLOBE:** Eine kleine drehende Weltkugel. PopUp-Menu, Source in C. **ICONS:** verschiedenste, animierte Icons (Disk, Drawer, Tool, Project, Garbage). Sehr originell und lustig!

KICK 124: ANWENDUNG

STARCHART: umfangreiches und interessantes Sternprogramm mit unendlich vielen Einstellungen. Mit der Maus können Informationen über die Sterne abgerufen werden. **AIRFOIL:** ein einfacher Windkanalsimulator, verschiedene Einstellungen.

KICK 123: UTILITIES

Neue Utilities, die aus den aktuellen Einsendungen zusammengestellt werden. Hier gibt es immer ein riesiges Angebot, denn der AMIGA läßt viele Möglichkeiten für Hilfsprogramme.

KICK 122: MODULA (AMOK 12)

Auch auf dieser AMOK-Diskette gibt es wieder jede Menge Routinen und Programme: Modul zum Ansteuern des Fischertechnik-Interfaces, ein schnelles und sicheres HD-Backup, Disassembler für M2Amiga-Object-Files, ArraySort, Janus, Timer, StripText, ConvLibSym, MakeInLine, Diff, Tips zum M2-Compiler

KICK 121: MODULA (AMOK 11)

R.o.M.: ein komfortables Mathematikprogramm und Funktionsplotter. **TrackDiskSupport:** erleichtert Programmierung der TrackDisk. **M2Druka:** Druckutility für ASCII-Dateien. **Beep:** erzeugt akustisches Warnsignal.

KICK 120: UTILITIES

PCopy, HexCalc, Dux5 (CLI-Utility), SDBackup, FlamKey (Computerschloß), VirusX, SCT (Farbeeinstellung von WB), Surveyor, HPMan (totale Kontrolle über den Laser)

KICK 119: AMOK #10

FlightDemo u. 3D-Demo (Echtzeitdemonstrationen bewegter räumlicher Grafik - super), FileRequester, SuperLists 1.3, Speech (dt. Sprachausgabe), Feigenbaum.

KICK 118: AMOK #9

GraphicSupport (Makros für Graphics-Primitives/Intuition), PrintIt2.0 (Hardcopyroutine für Epson), M2Test (Dokumentation über Fehler bei ARRAYS mit Index BOOLEAN), TextFont (Laden v. Zeichensätzen), Sounds und Graphics.

KICK 117: AMOK #8

MemSystem (Speicherverwaltung), PrinterSupport (unterstützt Printer.device, IDCMP (Hilfsroutinen), Blitter (direkte Blitterprogrammierung), SpaceDemo (3-D Demo), IFF8SVXLoad (Laden/Abspielen gesampelter Sounds).

KICK 116: AMOK #7

Reminder (Wecker für Programmierer), WarpText (schnelle Textausgaberroutine), MuchMore (Softscroll), M2Test (Hinweise auf Fehler in den Bibliotheksmodulen), SoundTask, AVL-Bäume, Listen, Queues, Stack.

KICK 115: SPIELE

CHEES-BOXES ist das allseits bekannte "Käsekästchen"-Spiel, das man jedoch nicht mit dem folgenden verwechseln sollte. 2-4 Spieler, mausgesteuert, AMIGABasic. Autor: J.D. Mallander. **VIER GEWINNT:** zwei Spiele, mausgesteuert, AMIGABasic. Autor: Björn Kessels. **COSMIC:** Strategiespiel. Source in C. Autor: Carl Edman. **SOLITAIRE** löst beliebige Stellungen des bekannten Spiels. Source in C. Autor: Thomas Lagally. **GRID IT:** spannendes Rennen für zwei Spieler. Benötigt Joystick, source in C. Autor: Jörg Tuttas. **LABYRINTH II:** Textadventure in Stil der INFOCOM-Spiele. Englisch, C.

KICK 114: BILDER-SHOW

Eine Grafik-Show mit Mandelbrotbildern und Colorcycling. Dadurch wird die Wirkung der eindrucksvollen Bilder hervorgehoben. Autor: Hermann Dörries, Wildeshausen

KICK 113: SOUNDS

Auf dieser Diskette ist der zweite Teil der KICKSTART SOUND DEMO III mit den Stereo-MIXES: Return to Defender, Dark Castle Mix, Chicago-Tune, Disco-Mix und Crazy Tune. Wichtig: Die Diskette läuft nur mit KICK 112!

KICK 112: SOUNDS

Zum dritten Mal haben unsere Soundspezialisten TOB & TWP zugeschlagen. Das Ergebnis ist die KICKSTART SOUND DEMO III mit einer Menge hervorragender Sounds auf zwei Disketten (siehe KICK 113). Mit von der Partie sind in der Rubrik CHARTS: Relax, Jean Michelle Jarre, Every Breath You Take, Axel F., Push, Tell It To My Heart und Blue Monday. Alles HiFi-Stereo mit einer extralangen Spielzeit.

KICK 111: BILDER

20 wunderschöne IFF-Grafiken von Frauenprofilen. Mit Diashow, sehenswert!

KICK 110: UTILITIES

MEMORYCLOCK: zeigt Uhrzeit, Datum, RAM- und Chip-Memory. **DATAMAKER, IFF-CONVERTER:** wandelt Dump-Format-Bilder in IFF-Format. Source zu allen Programmen in Assembler, Autor: Roger Fischlin. **SDBACKUP:** Festplatten-Backupprogramm, das die komplette Platte in komprimierter Form auf Disketten abspeichert. Viele Optionen. **RUNBACK:** läßt Programme im Hintergrund laufen, damit sie keine CLI-Fenster benötigen.

KICK 109: SCHULE

VOK-BOY V1.0: Vokabeltrainer in AmigaBasic. Autor: Michael Hennemann
WORD: Fremdsprachenlernprogramm, das Begriffe und ganze Sätze abfragt. Mit umfangreichen Lektionen in Englisch und einigen in Französisch. Kann leicht erweitert werden. Verwaltet beliebige Sprachen, sehr komfortabel. Autor: Daniel Neukomm, Bern (Schweiz)

KICK 108: BILDER

Viele schöne Bilder zum Anschauen und Weiterbearbeiten.

KICK 107: ANIMATIONEN

Weitere Animationen von Tobias Richter: JET (Düsenjet-Landeanflug), RELIANT (ein Raumschiff im frontalen Anflug), NELSON (rotierendes Raumschiff). Die Animationen demonstrieren, wie eindrucksvoll die mit VideoScape erstellten Sequenzen sein können. Sehenswert!

KICK 106: ANIMATIONEN

Tobias Richter hat uns wieder einige seiner mit VideoScape 3D-Animationen geschickt: LOTUS (der bekannte Lotus in einer neuen Sequenz), WILKERSON, DURETT und RELIANT (Raumschiffe in schneller Rotation).

KICK 105: ANIMATIONEN

LLEWELLYN ist eine mit dem Sculpt 3D-Animator erstellte Filmsequenz mit musikalischer Untermalung. Eine schöne Demo dieses Programms von Michael Clnard.

KICK 104: ANWENDUNGEN

DIRECTORY MASTER V1.1: Dieses Programm bringt Ordnung in Ihre Diskettensammlung. Die Dateien jeder eingelegten Diskette werden automatisch in eine Liste übernommen. Diese kann dann auf komfortable und vielfältige Weise bearbeitet und verändert werden. Sehr viele Funktionen, einfache Bedienung, sehr leistungsfähig und schnell. Als Zugabe gibt es die jeweils aktuelle Liste des KICKSTART PUBLIC DOMAIN SERVICE.
MEMOPAD: ist ein echt nützliches Hilfsprogramm für vergessliche AMIGA-Anwender. Hier können wichtige Termine und Daten festgehalten werden. Bindet man den Befehl CHECKMEMO in die Startup-Sequence ein, dann erinnert das Programm an diese Termine. Einfache Mausbedienung, sehr komfortabel und hilfreich.

KICK 103: GRAFIK

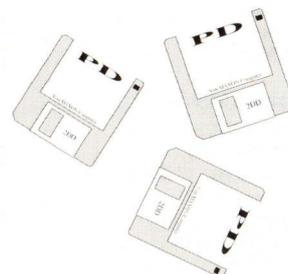
C-LIGHT: ein sehr einfach zu bedienendes, mausunterstütztes Raytracer-Programm. Die Einstellungen erfolgen über viele Regler und Schalter. **NEWJUGGLER:** ein Grafikprogramm, das den Eric Graham's JUGGLER innerhalb eines Bildes ablaufen läßt. Source in C; Autor: Peter Weiland, Wietze. **HAM'S:** einfaches Malprogramm für den HAM-Modus in AmigaBasic. Das Programm kann leicht geändert werden, da der Source-Code gut strukturiert ist. Autor: Henning Frommer, Vallendar

KICK 102: SPRACHEN

STONY BROOK PROLOG: auf dieser Diskette befinden sich die Source-Codes (Lattice C 4.0 und Prolog) des PD-Prolog-Systems von Diskette 101.

KICK 101: SPRACHEN

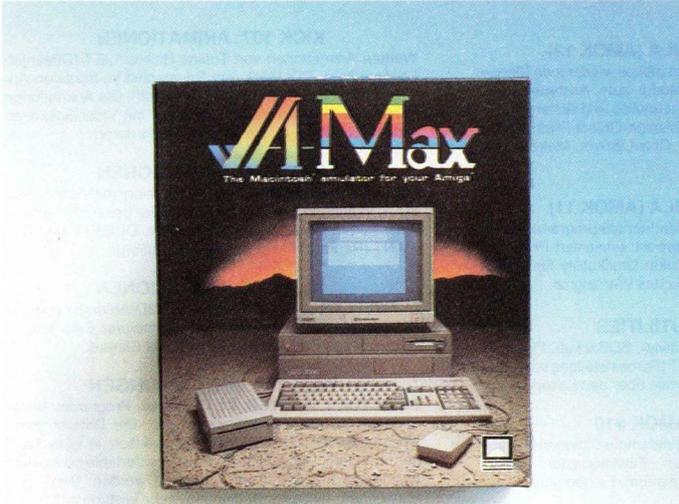
STONY BROOK PROLOG Version 2.3.2: Dieses PD-Prolog-System basiert auf einem System, das ursprünglich für UNIX-Rechner konzipiert wurde. Das System beinhaltet sowohl einen Interpreter als auch einen Compiler (bis zu 3000 LIPS). Der Sprachumfang entspricht weitgehend dem Edinburgh-Standard. Das System besitzt zusätzlich einige Besonderheiten z.B. dynamische Prädikateinbindung, Mischen von interpretiertem und kompiliertem Code, Macros und andere Hilfen. Inklusiv ca. 100seitiger Dokumentation. Benötigt mindestens 1MB Speicher.



VORSCHAU

Der MAC-Emulator

Durch eine kleine Hardware-Platine, auf der sich die MAC-ROMs befinden und einem Software-Emulator wird der AMIGA zum MAC. Was von dem Emulator erwartet werden kann, wie es mit der Geschwindigkeit aussieht und wie kompatibel die kombinierte Hard-/Software-Lösung ist, darüber können Sie sich in der nächsten Ausgabe informieren.



SCSI-Wechselplatte

Festplatten für den AMIGA sind immer mehr im Kommen, das neueste Produkt aus dem Hause GVP ist eine SCSI-Wechselplatte mit 44 MByte Speicherkapazität. Ob sie mit normalen Platten mithalten kann, muß sich erst zeigen. Mehr darüber im nächsten Heft.

AMIGA-GRAFIK

GRAFIK wird das Leitthema der nächsten Ausgabe sein. Dazu lesen Sie ausführliche Tests der verschiedensten Grafikprogramme: C-LIGHT II, das neue Raytracing-Programm aus England; PHOTON PAINT 2.0, jetzt mit Animation; DELUXE PAINT III, der Nachfolger des legendären Zeichenprogramms; ANIMAGIC und VIDEOSCAPE 3D V2.0 von AEGIS; das neueste aus dem Hause AEGIS und vieles mehr.

Neben den brandneuen Programmen stellen wir kurz noch einmal ältere Klassiker vor, beispielsweise die "Grafik-Programmiersprache" THE DIRECTOR oder den Bildermanipulator PIXMATE. Grafikbegeisterte AMIGA-User dürfen sich das nicht entgehen lassen.

Und was lesen Sie sonst?

Neben dem komplexen Grafikteil lesen Sie, was in dem Diskettenbeschleuniger B.A.D. wirklich steckt, was sich rund um den AMIGA getan hat, und dazu gibt's eine Menge Listings. Auch Spieletafeln werden nicht zu kurz kommen, soviel darf vorweggenommen werden.

Ab 18. August an Ihrem Kiosk.

Änderungen vorbehalten

Impressum

KICKSTART

Chefredakteur:

Uwe Bärtels (Chefredakteur)(UB)
Markus Nerdling (Stellvertreter) (MN)

Redaktion:

Andreas Krämer (AK)
Harald Schneider (HS)
Marcelo Merino (MM)
Harald Egel (HE)
Christian Keller (CHK)

Herausgeber:

MAXON-Computer GmbH
Industriestraße 26
6236 Eschborn
Tel.: 06196/481811
FAX: 06196/41137

Redaktionelle Mitarbeiter:

Gerald Carda (GC)
Jobst Hermeier (JH)
Carsten Borgmeier (CBO)
Martin Silbernagl (MS)
Sven Stillech (SV)
Florian Du Bois (FDB)

Redaktionsanschrift:

MAXON-Computer
KICKSTART
Postfach 5569
6236 Eschborn
Tel.: 06196/481811

Public Relations:

Claus Peter Lippert

Auslandskorrespondenz:

Derek Dela Fuente (ddf/GB)

Titelbild:

Gabriele Imhof

Fotografie:

Gabriele Imhof
Klaus Ohlenschläger Foto & Design
Bernd Opitz

Illustrationen:

Barbara Jacobs

Comics:

Pit Burkhard

Produktion:

Gabriele Imhof
Karl-Heinz Hoffmann

Druck:

Ferling Druck, Darmstadt

Verlag:

Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt 13
Tel.: 06151/56057
FAX: 06151/55689 - 56059

Verlagsleitung:

Hans-Jörg Heim

Anzeigenverkauf:

Kyriakulla Margaritis
Uwe Heim (Ltg.)

Anzeigenpreise:

nach Preisliste Nr.3, gültig ab 1.1.88

Bezugsmöglichkeit:

Zeitschriftenhandel, Kauf- und Warenhäuser, Commodore-Fachhändler oder direkt beim Verlag.
KICKSTART erscheint 11 mal im Jahr
Einzelpreis: DM 7,-, ÖS 56,- SFr 7,-
Jahresabonnement/Inland: DM 70,-
Europ. Ausland DM 90,-
Luftpost DM 120,-

Alle in KICKSTART erscheinenden Beiträge sind urheberrechtlich geschützt. Reproduktionen gleich welcher Art, ob Übersetzung, Nachdruck, Vervielfältigung oder Erfassung in Datenverarbeitungsanlagen, sind nur mit schriftlicher Genehmigung des Herausgebers und des Verlags erlaubt. Programm listings, Bauanleitungen und Manuskripte werden von der Redaktion gerne entgegengenommen. Sie müssen frei von Rechten Dritter sein. Mit ihrer Einsendung gibt der Verfasser die Zustimmung zum Abdruck und der Vervielfältigung, Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte wird keine Haftung übernommen.

Sämtliche Veröffentlichungen in KICKSTART erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes, auch werden Warennamen ohne Gewährleistung einer freien Verwendung benutzt.

Für Fehler in Text, in Schaltbildern, Aufbauskißzen, Stücklisten, usw., die zum Nichtfunktionieren oder evtl. zum Schadhafwerden von Bauelementen führen, wird keine Haftung übernommen.

(c) Copyright Heim Verlag

Profilaufwerk 3,5"

Metallgehäuse • einstellbare Laufwerksnummer mit Displayanzeige • digitale Trackanzeige • Write Protect am Laufwerk schaltbar • abschaltbar • durchgeschleif-ter Bus
1 Jahr Garantie
Super ALCOMPPreis

329,-

Laufwerk 5,25"

40/80 Track • Laufwerksbus durchgeschleift • abschaltbar • einstellbare Adressen • MS-DOS-kompatibel • mit Diskchange • Amigafarbene Blende
Super ALCOMPPreis
HD 1,6 MB (umschaltbar)
Write Protect Schalter

308,-

328,-

+15,-

Gemischtes Doppel 3,5/5,25"

einzel ein-/abschaltbar • einstellbare Laufwerksnummern mit Anzeige • durchgeschleif-ter Bus • bei 5,25" 40/80 Tracks umschaltbar • Metallgehäuse • 1 Jahr Garantie
Super ALCOMPPreis

558,-

500er Speichererweiterung

Für 512k zusätzliches RAM • alle RAM-s gesockelt • selbstkonfigurierend • abschaltbar • Uhrenschaltung auf Platine mit Akku- bzw. Batteriepufferung nachrüstbar
Komplett mit 512k Preis auf Anfrage
Superpreis mit Uhr Preis auf Anfrage
Bauteilesatz für Uhr ohne Akku **24,-**
Leerplatine mit Stecker ***39,-**
*mit Schaltplan und Bestockungsliste

Laufwerkanschlußkabel

Zum Anschluß von Laufwerken an alle Amigas • mit Ansteuerelektronik
Für 3,5" Laufwerk **39,-**
Für 5,25" Laufwerk **49,-**

Steckplatzerweiterung 3-fach für Laufwerke

Jeder Steckplatz abschaltbar und einstellbare Laufwerksnummer • Steckplatzerweiterung direkt am Amigagehäuse • Dadurch keine Kabel-längenprobleme
Anschlußfertig zum Super ALCOMPPreis **39,-**

Soundsampler

Für alle Amiga's mit Software • Type bei Bestellung bitte angeben • 8-Bit Datenbreite • Betrieb am Parallelport (Druckerport) • Mit Vorverstärker für Micro-Anschluß (Cinch-Buchsen) • Musik- und Sprachdigitalisierung möglich • Arbeitet mit fast allen Digitizer-Programmen • Formschönes Gehäuse
Super ALCOMPPreis **79,-**

Sampler Studio

• Professionelles Sampler-Programm • 4-Kanal-Technik • speichern auf 4 Disketten hintereinander möglich • alle gängigen Formate (IFF, Data, Future) • Echtzeitdisplay mit Zoomfunktion • viele Verformungsmöglichkeiten • Echo, Hall, Reverse
69,-
129,-
Paket: Sampler + Software

MIDI-Interface

4 Kanäle einschließlich 1 Thru • Optische Datenanzeige • Formschönes Gehäuse
Wahnsinnspreis von nur **89,-**

ausgereifte Ingenieurleistung • 14 Tage Umtauschrecht • fast alle IC's gesockelt • nur professionelle Leiterplatten • Bauteile namhafter Hersteller • mit Bedienungsanleitung

3,5" Laufwerk

Für alle Amiga's • einstellbare Gerätenummer • abschaltbar • Metallgehäuse • superflach • 1 Zoll (2,54cm) • durchgeschleif-ter Bus • TEAC Laufwerk
1 Jahr Garantie
komplett anschlussfertig **249,-**
incl. Amigafarbene Blende

Basislaufwerke

1 Jahr Garantie

TEAC FD 135 FN 3,5" 1 MB superslimline **218,-**
1,6 MB Diskchange **+10,-**
Amigafarbene Blende **25,-**
3,5" Gehäuse **25,-**
5,25" Gehäuse **25,-**
Gehäuse für "Gemischtes Doppel" **65,-**

Bootselector

19,90

Amiga Eprommer

• Für A 500/1000
• Expansionsportanschluß
• Für EPROM's 2764-27011 (8K-128K)
Alle A-Typen und CMOS-Typen
• Funktionen:
LEERTEST LADEN VON DISK
VERGLEICHEN SPEICHERN AUS DISK
AUSLESEN HEXDUMP
BRENNEN
• vier Programmieralgorithmen
50ms/Byte - Superschnell 64K-1,5 min
• Programm zum Generieren und Brennen von Kickstarts direkt von Diskette oder aus ROM
• Mit Software + Gehäuse **225,-**

Meß- und Steuerinterface

• 8 ADC-Kanäle 0-2,55V in 0,01V Stufe
• 1 DAC-Kanäle 0-2,55V in 0,01V Stufe
Genauigkeit: 1,5 LSB
• 8 frei programmierbare TTL-I/O Kanäle
• Mit Gehäuse, Anschlüsse auf Schraubklemmen
• interne Referenzspannung
• Expansionsanschluß
• Einfache Programmierung in Basic möglich
Multitasking tauglich
• incl. DEMO-Software auf 3,5" Diskette **239,-**

Trackanzeige

Für DFD-DF3 einstellbar • für alle Laufwerke (3,5"/5,25") • Laufwerksbus durchgeschleift • mit Gehäuse
Super ALCOMPPreis **79,-**

Amiga - Harddisks

komplett anschlussfertig
Platte 20 MB A 2000 **798,-**
30 MB A 2000 **898,-**
40 MB A 2000 **1148,-**
65 MB A 2000 **1498,-**
Platte A 500/A 1000
20 MB **998,-**
30 MB **1098,-**
40 MB **1348,-**
65 MB **1598,-**

für den Selbstbau

HD-Interface A 2000 **198,-**
HD-Interface A 500/A 1000 **249,-**
für verschiedene XT-Harddisk-Controller

Vokabeltrainer

2500 englisch-deutsche Vokabeln incl. Hilfsatz • Merkfunktion • komfortabler Editor zur Vokabelverwaltung • Wörterbuch zum Dateiendurchsuchen **59,-**



Jetzt: Alle Laufwerke mit amigafarbener Blende!



Selbstbootende Harddisk für Amiga mit/bhne PC-Karte!

Die Amiga-Festplatte von ALCOMP
• Startet beim Einschalten/RESET ohne Bootdiskette! • Als Einbau-Festplatte für den "Amiga 2000" • Als Externe Einheit für den "Amiga 500" und 1000 mit Gehäuse, eigenem Schalt-Netzteil und Erweiterungsanschluß • Erhältlich mit 20, 30, 40 und 65 Megabyte • Kopiert 1 Megabyte in unter 4 Sekunden • Speichert schneller als "1.2-Ramdisk" • Läuft mit "FastFileSystem" • Mit intelligenter Installationssoftware
Für den Selbstbau:
Harddisk-Interface incl. Steuersoftware • Anschluß mit Slot für Harddisk-Controller



Kickstartumschaltung

Bauen Sie die anderen Kickstart-Versionen in Ihren Amiga 500 • Einfacher Einbau ohne Löten • für Original-Kickstart-ROM und 2 zusätzliche Versionen auf EPROM • EPROM-Programmierservice auf Anfrage

SuperALCOMPPreis **59,-**

Kickstartversion auf EPROM's **120,-**

Userport + Experimentierkarte für Expansionport

Mit Lochraster und 2 x 6522 Ports
Leer **59,-**
komplett aufgebaut **89,-**

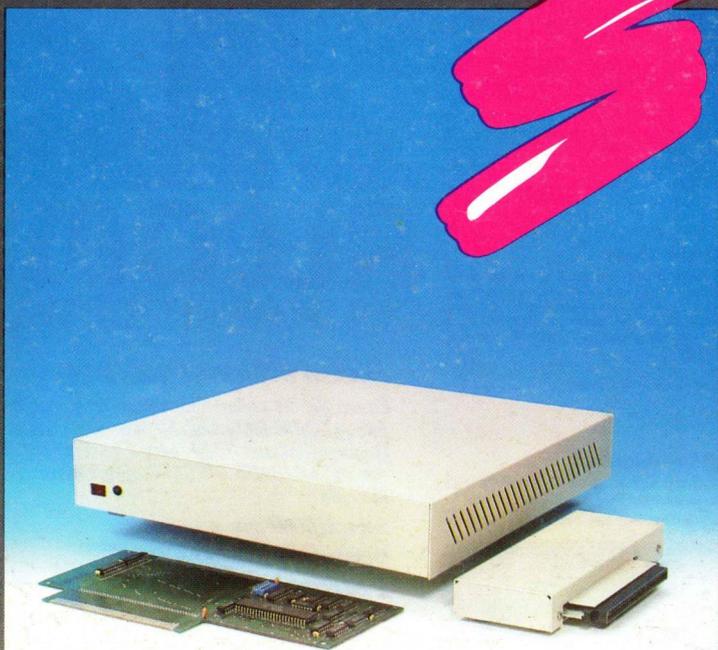
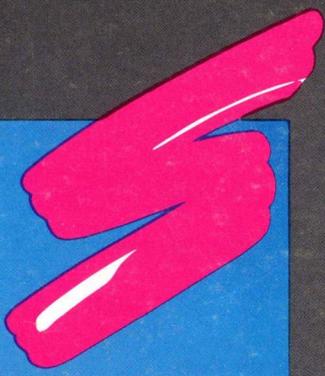
Wir suchen ständig Hardware-Entwicklungen. Wir garantieren gute Umsatzprovisionen und ehrliche Abrechnung

kostenloses Info anfordern!!!

Bestellung und Versand

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

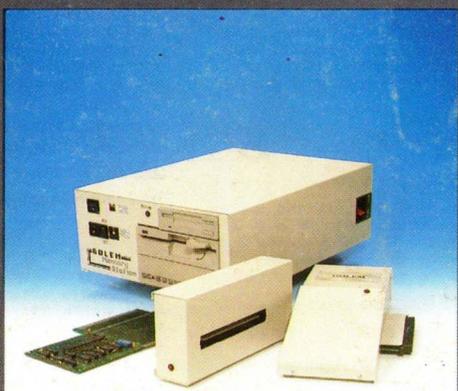
Nachnahmeversand NN-Spesen 10,- DM b. Vorkasse 5,- DM. Auslandsbestellungen: Nachnahmeversand NN-Spesen 15,- DM b. Vorkasse 10,- DM. Wir liefern Ihnen auf Ihre Rechnung und Gefahr zu den Verkaufs- und Lieferbedingungen des Elektronikgewerbes.
Postgiroamt Köln
(BLZ 370 100 50) 275 54-509



HD 3000

externes Festplattensystem für alle Amiga
 überragende Leistungsmerkmale wie Datentransferrate bis **327K/sec** ● NEC Qualitätsplatten ● formschönes Gehäuse als Monitorunterbau ● eigenes Netzteil ● Lüfter ● sehr komfortables Format-Menue ● defekte Zylinder werden erkannt, gesperrt und ersetzt, so erhalten Sie auch bei Harderrors immer die volle Plattenkapazität und brauchen nicht nach Fehlerrate Ihre Partitionen zu wählen

20 Megabyte	998.-
30 Megabyte	1098.-
40 Megabyte	1498.-
60 Megabyte	1698.-



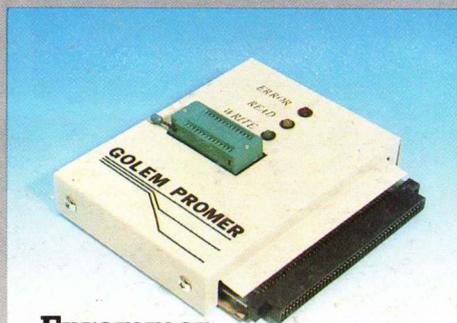
Golem Memory Station

externes Festplattensystem für alle Amiga
 technisch mit den gleichen überragenden Leistungsmerkmalen wie HD 3000 ● zusätzlich kann eine weitere Festplatte oder ein 5,25 Zoll Laufwerk und ein 3,5 Zoll Laufwerk eingebaut werden ● Elektronik mit zwei Trackdisplays für die Laufwerke ist bereits vorhanden ● Im Harddisk Interface ist ein 2 Megabyte dynamik Ram Controller integriert
 Nutzen Sie die Möglichkeit sehr preiswert Ram und Floppies nachzurüsten

20MB	1398.-
30MB	1498.-
40MB	1898.-
60MB	2098.-
Nachrüstätze zur Memory Station	
2MB Rambank	1098.-
3,5 Zoll Laufwerk NEC	
incl. Kabelsatz	200.-
5,25 Zoll Laufwerk NEC	
incl. Kabelsatz	220.-



(02 31)
81 83 25-27



Eprommer

Eprombrenner für alle gängigen Eproms bis zum Megabit-Eprom leistungsfähiger integrierter Monitor ● komfortables Menue sehr schnell z.B. 27512 (64KB) in 15 Sekunden

Golem Eprommer incl. Brennsoftware u. Monitor **249.-**

GOLEM

HARDWARE

AKTUELL

Amiga 2000 B

mit 1 MB RAM
 incl. Farbmonitor 1084 Stereo
 PC XT Karte m. 5,25 Zoll Drive
 sofort lieferbar **2998.-**

Amiga 2000 B

mit 1 MB Ram
 incl. 14 Zoll schw./weiß Monitor
 20 MB Autoboot-Festplatte
 Textverarbeitung Word Perfect 5.0 in Deutsch
 sofort lieferbar **2798.-**

8 MB für A 2000

2 MB bestückte Einsteckkarte ● 8 MB gesockelt ● bestückbar mit Megabit DRam autokonfigurierend **1298.-**

12 K Karte für Amiga 500

Einsteckkarte mit 512 K Ram und Akkugepufferter Uhr **329.-**

Trackdisplay Amiga 2000

Display zur aktuellen Spur und Kopfanzeige für DF0: u. DF1: **89.-**

Hardware Virus Protector

Extern für alle Amiga Laufwerke
 Schützt sicher Ihre Disk vor unbemerktem Schreibzugriff **39.-**