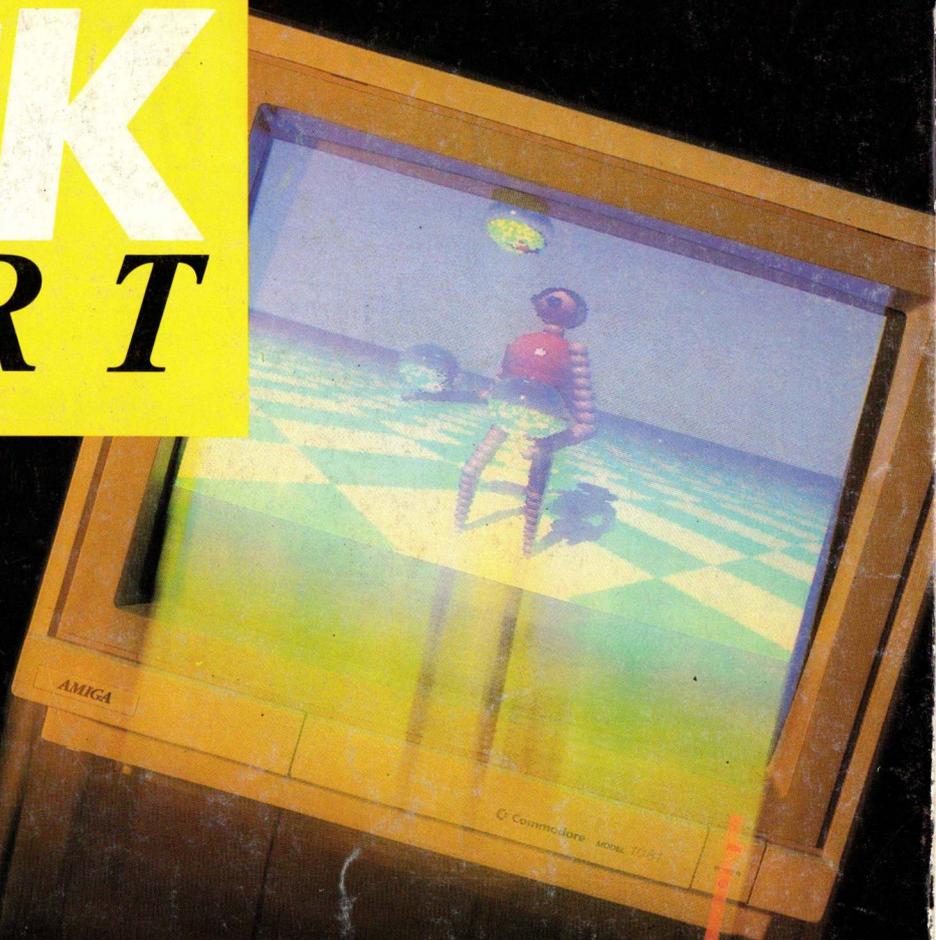


KICK START



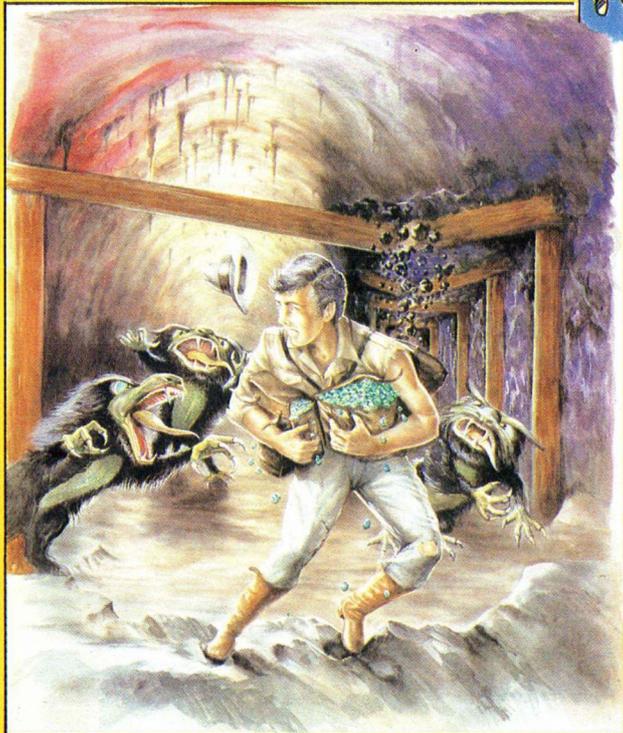
Test : Sculpt 3D · Apprentice Animator · Digipaint · Festplatten
Grundlagen : Intuition · Assembler · Tips & Tricks : zu Depaint 2
Listings : Movie Maker · C Shell · Vokable Craft · Neue Spiele

BILLIG-SPIELE FÜR AMIGA und ST MÜSSEN NICHT »BILLIG«



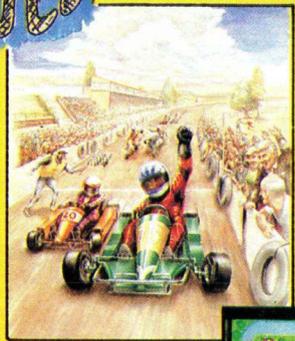
CITY DEFENCE
von Uwe Kühner

Eine schier endlose Zahl tödlicher Laserstrahlen aus dem Weltall kann nur gestoppt werden, wenn Sie zum richtigen Zeitpunkt ihre Bodenraketen abschies- sen, bevor die Städte zerstört werden. 1/2 Spieler; Maussteuerung. Lieferbar für AMIGA



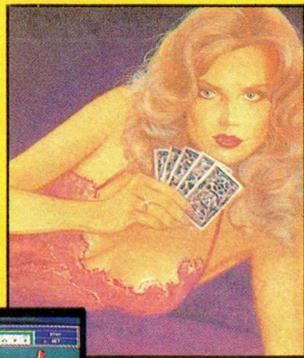
EMERALD MINE
von Klaus Heinz und Volker Wertich

Ein Riesen-Arcade-Adventure mit über 100 Le- veln und 2-Spieler-Option (gleichzeitig im Teamwork!); Joysticksteuerung. "Die flotte Grafik mit witzigen Monster-Sprites und vor allem der Sound gefallen mir sehr gut: Die fetzige Titelmusik und die digitalisierten Sound- Effekte sind exzellent. Das Verblüffendste an diesem tollen Spiel ist aber der sehr niedrige Preis: Wer hier nicht zuschlägt, ist selber schuld." (HAPPY COMPUTER)
Lieferbar für AMIGA und ST



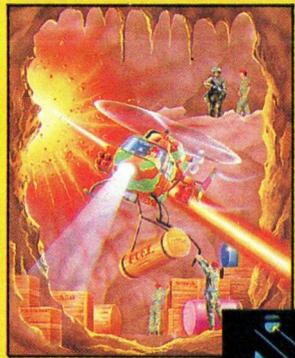
GOKART RACING
von Anco

Ein exzellentes Autorennen, bei dem ein echter Champion gefordert wird. Bei unterschiedlichsten Witterungsbedingungen und Strecken müssen Sie vorher Reifen und Übersetzung wählen. 8 Kurse, 1/2 Spieler (gleichzeitig); Joysticksteuerung. Lieferbar für AMIGA und ST



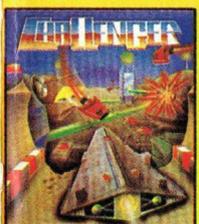
STRIP POKER
von Artworx

Verbringen Sie einen schönen Abend mit einer Runde Strip Poker. Mit Susi und Melissa stehen Ihnen 2 bildschöne, aber spielstarke Partnerinnen zur Verfügung. Bequeme Spielsteuerung mit der Maus. Lieferbar für AMIGA und ST

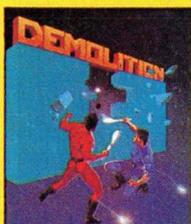


FORTRESS UNDERGROUND
von Andreas von Lepel

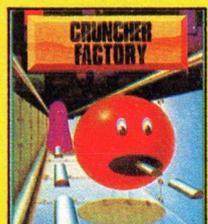
In einer gigantischen unterirdischen Höhle (640 Bildschirm groß) müssen Sie ein feindliches Kraft- werk aufspüren und eliminieren. Aber der Weg dorthin ist lang und gefährlich. Joysticksteuerung. Lieferbar für AMIGA



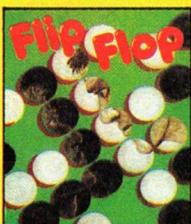
Lieferbar für AMIGA



Lieferbar für AMIGA



Lieferbar für AMIGA



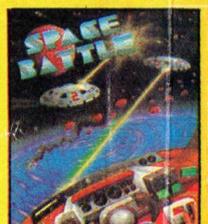
Lieferbar für AMIGA/ST



Lieferbar für ST



Lieferbar für AMIGA



Lieferbar für AMIGA



Lieferbar für ST



Lieferbar für ST



Lieferbar für AMIGA/ST

KINGSOFT-Spiele gibt's überall da, wo es gute Software gibt.

SPITZEN-SOFTWARE
...natürlich von



MADE IN GERMANY

SENSATIONELL

je Spiel nur DM

29.⁹⁵

unverb. Preisempfehlung

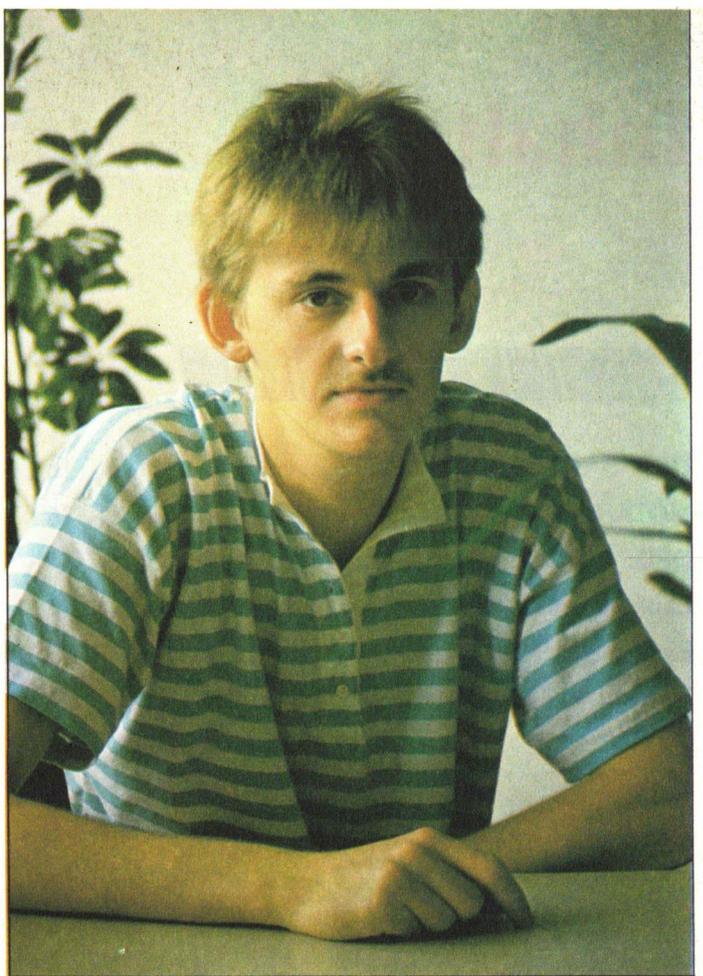
Bei uns gibt's jeden Monat Neuerscheinungen für Ihren AMIGA und ST - am besten kostenlos den aktuellen Katalog anfordern!

Alle Spiele selbstverständlich mit aus- führlicher deutscher Anleitung, Super- Grafik und fantastischen Digisound- Effekten.



F. Schäfer · Schnackebusch 4 · 5106 Roetgen
☎ 02408/5119 (nicht aufgeben!) Fax 02408-5213

EDITORIAL



Kopierschutz auf Kosten des Anwenders

Liebe Leserin, lieber Leser,

ein leidiges Thema auf dem Softwaremarkt ist der Kopierschutz. Bei Personal Computern, und zu diesen wird der Amiga bekanntlich gezählt, bewegt dies schon seit geraumer Zeit die Anwender.

Die Problematik ist, daß Softwarehäuser wegen der Raubkopierer zu Recht um ihre Verkaufszahlen fürchten und deshalb gezwungen sind ihre Produkte vor unbefugter Vervielfältigung zu schützen.

Leider erweist sich ein solcher Schutz in der Praxis eher für die ehrlichen User als für die Raubkopierer als hinderlich, denn meist hat die Crackerszene schon nach kurzer Zeit zugeschlagen und den Schutz entfernt.

Diejenigen, die das Produkt käuflich erworben haben, plagen sich aber weiterhin mit dem Schutz herum. Ich spreche von plagen, weil inzwischen eine Vielzahl verschiedener, in meinen Augen sehr hinderliche, Kopierschutzverfahren eingesetzt werden. So zum Beispiel diejenigen, die das Floppylaufwerk quälen und somit die Lebenserwartung des Laufwerks und der Originaldiskette nicht gerade erhöhen. Weiterhin kann man solche Programme in der Regel nicht von einer Festplatte betreiben.

Eine andere Art ist der sogenannte 'Dongle'-Stecker, der meist im Joystickport plaziert wird. Zwar hindert er in keiner Weise die Programmbedienung und erlaubt auch ein Portieren des Programms auf die Festplatte, doch was ist, wenn man zwei 'Dongle-Programme' gleichzeitig betreiben möchte – ein krasser Widerspruch zum Multitasking Betriebssystem des AMIGA.

Eine andere, derzeit beliebte Methode ist das Abfragen eines bestimmten Wortes aus der Anleitung. Dies ist eine ganz akzeptable Lösung, denn man muß dazu nur einmal im Handbuch blättern, doch glaube ich kaum, daß eine solche Lösung im professionellen Bereich, in dem der AMIGA es ohnehin schon schwer genug hat, willkommen ist.

Man muß hier allerdings eine starke Unterscheidung zwischen Anwender- und Spielesoftware treffen. Bei letzterer ist eine Passwort-Abfrage durchaus als akzeptabel anzusehen. Bei der Anwendersoftware jedoch halte ich immer noch ein gutes Handbuch und vor allem einen guten Service für die beste Methode, die Anwender vom Raubkopiergebrauch abzuhalten. Solch ein 'guter Dienst am Kunden' beinhaltet unter anderem einen Update-Service, mit dem der registrierte Benutzer die Möglichkeit erhält, neue, verbesserte und erweiterte Versionen zu beziehen. Sogar Anwenderzeitschriften und Clubs gibt es schon.

Natürlich ist hier ein qualitativ hochwertiges Programm vorausgesetzt, denn nicht selten sind die Preise der momentan verfügbaren AMIGA Anwendersoftware zu hoch für das, was sie bieten. Vielleicht sollte generell mehr Zeit in die Software als in die Kopierschutz-Entwicklung gesteckt werden.

Ihr

H. Schneider

Inhalt:

SOFTWARE

DIGI PAINT	16
MODULA-2 von TDI	36
DFÜ: ONLINE und A-TALK Plus	64
Lauf Bursche: DISNEY 3D ANIMATOR	70
Lichtspiele: SCULPT 3D	76
Endlich: VIZAWRITE DESKTOP ist da!	92



VIZAWRITE DESKTOP ist endlich da

Wohl kaum ein Programm wurde vor derart langer Zeit angekündigt und mit so vielen Vorschußlorbeeren versehen wie das Textverarbeitungsprogramm VIZAWRITE. Der Test soll klären, ob sich die lange Wartezeit gelohnt hat, und was das Programm wirklich leistet.



GRAFIK und ANIMATION vom FEINSTEN:

Das Erstellen von räumlichen Figuren und Körpern nach dem RAY-TRACER-Verfahren ist mit SCULPT 3D kein Problem. Dagegen ist das Erstellen dreidimensionaler bewegter Körper mit dem DISNEY ANIMATOR (APPRENTICE). Nicht ganz so einfach, aber die Animationen sind überzeugend. DIGI PAINT ist nicht irgendein Malprogramm, sondern besonders dafür ausgelegt, digitalisierte s/w-Bilder nachträglich mit Farbe zu versehen.

HARDWARE

Festplatten von C LTD, XEBEX und MOUTAIN . 67

SPEICHERRIESEN

Festplatten haben einen besonderen Reiz für den Computeranwender, da sie einerseits ein enormes Speicherpotential und andererseits eine schnelle Zugriffszeit besitzen. Wir haben uns für Sie umgesehen und Festplatten unter die Lupe genommen.

GRUNDLAGEN

C-Kurs	13
Einführung in INTUITION, Teil 5	29
Hilfe die VIREN kommen!!!	22
MEGA PATCH für interne Mega-Erweiterungen ..	29
NEU: Der ASSEMBLER Kurs beginnt!	49
Freie Fahrt: AmigaBASIC Kurs, Teil 3	51
CLI: Der Command Line Interpreter, Teil 5	96
Tips & Tricks zu DPaint II	106



Hilfe, die VIREN kommen

Der erste Virus auf dem Amiga ist aufgetaucht. Unser Artikel soll Ihnen einen kleinen Einblick geben und Sie in die Lage versetzen, Ihren Rechner zu 'kurieren'.

LISTING

- Der VOKABELLEHRER hilft beim lästigen Lernen 40
- Luxuriöse KOMMANDO-OBERFLÄCHE:
Eine Hilfe für genervte CLI-Benutzer 56
- Erstellen Sie sich Ihre eigenen Filme
mit dem MOVIEMAKER..... 79



MOVIEMAKER zum Abtippen

Beliebige geometrische Figuren werden mit diesem Programm-Listing in Bewegung versetzt. Die Einstellungen werden einfach mit der Maus vorgenommen und dann kann die Sequenz aufgenommen werden. Anschließend werden die Werte geändert, und schon geht es weiter. Am Ende der Arbeiten kann der ganze Film in Ruhe angesehen werden.

RUBRIKEN:

Editorial.....	3
NEWS	6
Messebericht	9
Für einen guten Zweck: FUSSBALLTURNIER mit bekannten Firmen	12
PUBLIC DOMAIN SERVICE	74
Einkaufsführer	89

SPIELE

TERRORPODS	99
LURKING HORROR + STATIONFALL	100
FANTASIE III.....	102
MISSION ELEVATOR.....	104
LOUNGE LIZARD	105



SPIELE

Gleich zwei Adventure-Spiele gibt es von INFOCOM: STATIONFALL Und LURKING HORROR. MISSION ELEVATOR ist die AMIGA-Version des bekannten Actionspiels. Nach BARBARIAN hat Pynosis mit TERRORPODS ein neues spannendes Strategiespiel herausgebracht. FANTASIE III ist ein mausgesteuertes Grafik-Adventure, bei dem allerlei bösertige Wesen bekämpft werden müssen. Das letzte Spiel ist LOUNGE LIZARD, das nicht ganz jugendfrei ist, denn der Held der Handlung hat nur eines im Kopf: Frauen...



Kleinanzeigen	103
Bücher	110
LESERBRIEFE	112
Inserentenverzeichnis	113
Impressum.....	114
Vorschau	114

News

FACC, der Floppy- beschleuniger!

Wer jetzt eine Hardware-Erweiterung erwartet, der die Floppy auf Trab bringt, wird enttäuscht sein. Es handelt sich um ein reines Softwareprodukt aus dem Hause ASDG Incorporated. FACC kann man am einfachsten mit einer automatischen RAM-Disk vergleichen, oder, wer es genauer wissen will, FACC legt einen Cache Speicher an (Zwischenspeicher). Über diesen Speicher laufen alle Daten der Laufwerke und werden dort auch gespeichert. Wird auf das Laufwerk zurückgegriffen, beispielsweise ein erneutes Auflisten des Verzeichnisses wird zunächst im Cache Speicher überprüft, ob sich die Daten für das Verzeichnis noch dort befinden. Wenn ja, listet das System das Verzeichnis, ohne einen Laufwerkszugriff durchzuführen. Selbiges gilt auch für Programme. Voraussetzung ist jedoch ein genügend großer Cache Speicher. Die Dimensionierung erfolgt in einem Fenster, indem der Speicher beliebig groß gewählt werden kann, zusätzlich werden noch einige allgemeine Informationen angezeigt. Ein sinnvolles Arbeiten mit FACC ist jedoch nur anzuraten, wenn der Arbeitsspeicher mindestens 1 MByte beträgt. Bei kleineren Speicherdimensionen bringt FACC nicht den gewünschten Erfolg, da der Cache klein gehalten werden muß. Der AmigaDOS Befehl 'Addbuffers' ist mit FACC zu vergleichen, bringt aber bei weitem nicht den Komfort von FACC. Nicht nur die Möglichkeit der Nutzung des FastRam bei FACC ist von Vorteil, auch andere Kriterien sind ausschlaggebend. Bezüglich kann man FACC bei folgenden Händlern:

IM, Tel. 069/7071102
DTM, Tel. 06121/560084
und bei gut sortierten Fachhändlern
Preis: 79,- DM (AK)

GOLDRUNNER für Amiga



Microdeals Renner GOLDRUNNER ist ab sofort auch für den Amiga erhältlich. Die Umsetzung vom ATARI ST zum Amiga ist hervorragend gelungen. Der Spieler muß, wie so oft, wieder einmal die gute alte Erde retten. Die bösen Tritons gehen diesmal der Menschheit an den Kragen. Dazu muß der Weg durch gefährliche Ringwelten freigeschossen werden. Das Ballerspiel zeichnet sich durch das enorm schnelle Scrolling und die

gute Animation der Sprites aus. Dabei kann das Raumschiff nach Oben sowie Unten fliegen. Der Sound besticht nicht sonderlich. Eine Hintergrund-Musik und Explosionen sind zu vernehmen. Gespielt wird mit Maus oder Trackball, der Joystick ist überflüssig.

Anbieter: PDC, Tel. 06172/24748
und gut sortierte Fachhändler (AK)

Der Winterpelz für die Maus!



Wie jeder weiß, geht es auf die kalte Jahreszeit zu. Auch die Maus des

Amigas fängt an zu frieren. Deshalb haben findige Entwickler ein Maushaus auf den Markt gebracht. Es handelt sich dabei um einen Pelz, der über die Maus gestülpt wird. Vorbei ist das Frieren in der kalten Jahreszeit. Die Maus läßt sich weiterhin gut bedienen, Schwierigkeiten beim Verschieben, sowie das Drücken der Knöpfe, treten nicht auf. Erhältlich ist der Pelz für 19,95 DM.
PDC, Tel. 06172/24748

C64 Emulator

Ein Produkt, das sicher von vielen lange erwartet wurde, soll nun in Kürze lieferbar sein. Es handelt sich um GO-64! Ein Commodore 64 Emulator auf dem Amiga.

Der in Kürze lieferbare Emulator besteht aus zwei Teilen. Der Hardwareteil soll es dem Anwender ermöglichen, die C64 Laufwerke und Drucker über den gewohnten seriellen Port anzuschließen. Der Hersteller gibt folgende Features für den Emulator an:

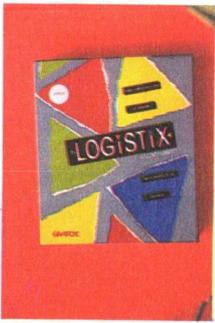
- Fast alle Basic- bzw. Assembler-Programme sind lauffähig.
- Der serielle Port des C64 ist voll funktionsfähig.
- Joystick bzw. Maus sind verwendbar.
- Der Ziffernblock des Amiga kann benutzt werden.
- Alle Graphikmöglichkeiten des C64 sind vorhanden.
- RGB und Composite Monitore werden unterstützt.
- Die benötigte Hardware ist im Lieferumfang enthalten.
- C64 Geräte sind voll funktionsfähig (z. B. 1541).

Der voraussichtliche Verkaufspreis in Deutschland wird bei 199,- DM liegen.

(GC)

LOGISTIX in deutscher Ausführung!

Die integrierte Datenbank ist ab sofort in deutscher Ausführung erhältlich. Einen ausführlichen Test von der englischsprachigen Version wurde in KICKSTART 8 (Juli/August) veröffentlicht. Trotzdem noch ein paar Informationen des Produktes: Tabellenkalkulation für numerische Projektionen und Analysen, Termin- und Netzplanung für die zeitabhängige Planung von Vorgängen und Ressourcen, luxuriöse Datenbank,



an. Der Controller entspricht der Version, die auch für die CLtd. Festplatten beliefert wird. Der Anwender erhält nun die Möglichkeit, mit eigenen Platten bzw. beliebigen SCSI Geräten (z. B. CD-ROM!) zu experimentieren. Der Controller unterstützt mehrere Geräte und kostet in der Grundversion 495,- DM. Der für den Betrieb am Amiga 1000 nötige Host Adapter schlägt mit weiteren 349,- DM zu Buche. Versionen für den A500 bzw. A2000 sind auf Anfrage erhältlich.

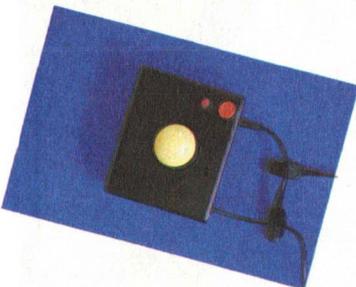
Intelligent Memory Frankfurt
Tel. 0 69 / 707 11 02 (GC)

Ein neues Gesicht

Grafikdarstellung zur Veranschaulichung, Trendanalyse und vieles mehr. PDC, Tel. 0 62 72 / 2 47 48
Preis: 340,- DM (AK)

Mausersatz gewünscht? Trackball!

Wer kennt die Probleme nicht? Der Bewegungsfreiraum der Maus reicht nicht aus. Zettel, Stifte, Disketten usw. liegen im Weg, auf dem zu kleinen Schreibtisch. Abhilfe könnte hier ein Trackball schaffen, der im Prinzip nur eine umgedrehte Maus darstellt. Der Vorteil eines Trackballs besteht darin, daß er einen festen Platz auf dem Schreibtisch einnimmt. Mit



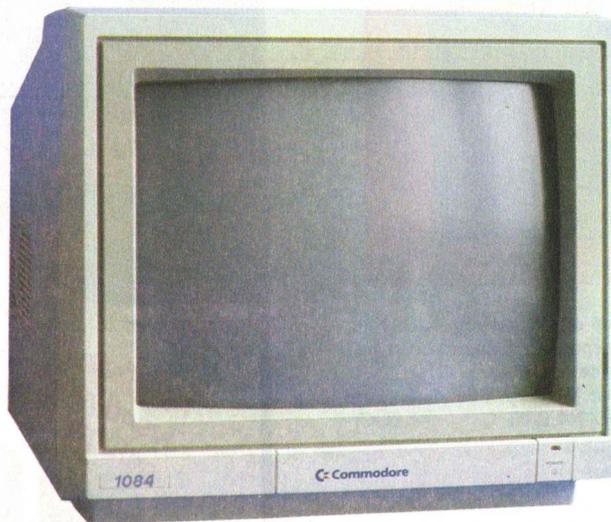
Hilfe einer Billardkugel wird der Mauszeiger über den Bildschirm bewegt; die Umstellung von Maus auf Trackball ist gewöhnungsbedürftig, doch nach einer gewissen Einarbeitungszeit treten keine Probleme mehr auf. Beziehen kann man den Trackball bei nachstehenden Händlern:

PDC, Tel. 0 62 72 / 2 47 48
IM, Tel. 0 69 / 707 11 02
DTM, Tel. 0 61 21 / 56 00 84
Preis: 79,90 DM

(AK)

SCSI Controller

Für alle Leute, die gerne basteln, bietet IM jetzt einen SCSI Controller



Seit kurzem gibt es einen neuen Monitor für den Amiga. Er trägt die Typenbezeichnung 1084. Der Monitor zeichnet sich durch ein brillantes Bild und eine mattierte Bildröhre aus. Das Arbeiten an diesem Bildschirm ist nun auch mit einer Textverarbeitung auf längere Zeit möglich. Erste Exemplare sind ab 1.10.

bei einigen ausgewählten Fachhändlern erhältlich. Mitgeliefert werden Anschlußkabel für den Amiga und den C128. Auch ein Schaltplan befindet sich in der Packung. Der Preis liegt bei ca. 950 DM. Bei Commodore Fachhändlern.

(GC)

Football Fortunes

Eine eigenwillige Variante bietet das englische Softwarehaus CDS Software LTD an. Es handelt sich dabei um ein 'Zwitter-Spiel'. Der Computer übernimmt die Schiedsrichterrolle,

während die Spieler auf einem mitgelieferten Spielbrett Figuren ziehen müssen. Das Fußball Simulationspiel ist auf den Rechner angewiesen und umgekehrt. Ein eigenständiges



Spielen ist nicht möglich. Der Computer übernimmt dabei das Würfeln der Spieler, die Berechnung der Ergebnisse, Ausgabe der Tabelle und verschiedener Ereigniskarten etc. Das Hauptgeschehen findet auf dem Spielbrett statt. Im Lieferumfang befindet sich eine Diskette, das Spielbrett, Geld, Spielfiguren, spezielle Fußballkarten und eine Anleitungsbroschüre. Ziel des 'Computerbrettspiels' ist es, seine Fußballmannschaft am Ende der Meisterschaft ganz oben zu wissen. Nachteilig bei diesem ungewohnten Spielprinzip ist lediglich die englische Ausführung. Beziehen kann man das Spiel bei:

IM, Tel. 0 69 / 707 11 02
DTM, Tel. 0 61 21 / 56 00 84
und bei gut sortierten Fachhändlern (AK)

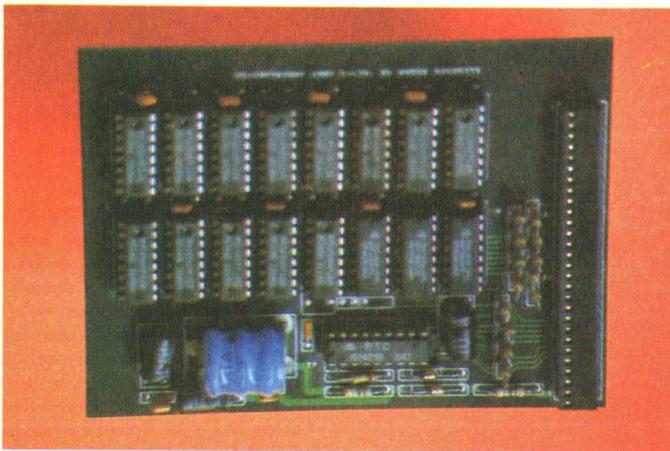
Chart Construction Set

Klaus Juris Computer Grafik liefert ein interessantes Paket an Bildern und Brushes für DPaint. Der Anwender erhält hiermit eine Bilderbibliothek, die es erlaubt, in kürzester Zeit eindrucksvolle Repräsentationsgrafiken zu erstellen. Die Diskette enthält zahlreiche Bausteine, mit denen z. B. Balken- oder Kuchendiagramme erstellt werden können. Ebenso sind zahlreiche Landkarten, Schriftarten, Flaggen und andere Objekte enthalten. Zum Betrieb ist allerdings ein DPaint Voraussetzung. Die Diskette kostet 89,- DM. Klaus Juris Computer Grafik, Bahnhofstr. 106, 6392 Neu Ansbach, Tel. 0 60 81 / 85 90

(GC)

NEWS

Speichererweiterung



Die Firma IM in Frankfurt hat den Preis für ihre 512 KByte Erweiterung für den A500 auf 249,- DM gesenkt. Diese Erweiterung wird, wie das Original, von unten in das Gehäuse eingesteckt und erweitert den

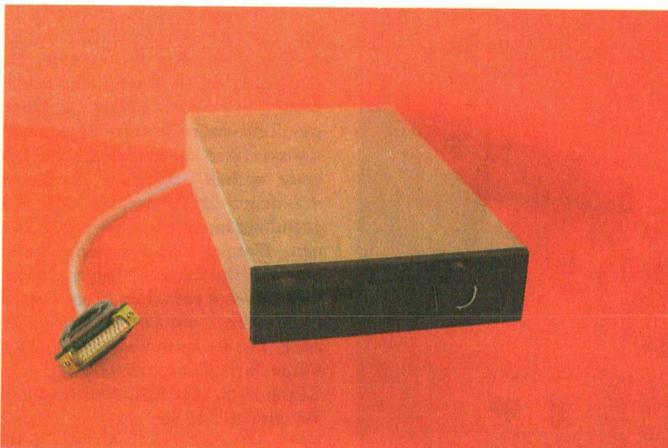
A500 auf ein Megabyte. Auch eine batteriegepufferte Echtzeituhr ist auf der Platine enthalten. Intelligent Memory, Ffm
Tel. 0 69 / 707 11 02 (GC)

Universell einsetzbare 5,25"-Floppy

Neu auf dem Markt erschienen, ist eine 5,25-Zoll-Diskettenstation von PDC, Homburg. Farblich nicht ganz auf den AMIGA abgestimmt (Gehäuse weiß, Front schwarz) kann sie im Alltagsbetrieb überzeugen. Weder als Zweit- oder Dritt(!)-Laufwerk am AMIGA noch an SIDECAR oder BRIDGEBOARD versagte das Laufwerk seinen Dienst. Eine Besonderheit ist der Schalter für den 40- oder 80-Track-Betrieb. Dadurch wird es möglich, das Laufwerk als normales AMIGA-Drive einzusetzen und MS-DOS-Standardformat zu lesen. Im AMI-

GA-Betrieb wird das Diskchange-Signal nicht unterstützt. Der Floppy-Bus ist nicht durchgeschleift, wodurch der Betrieb nur als Letztes Glied in der Laufwerkskette zulässig ist. Die Geräusentwicklung ist angenehm gering, etwas störend wirkt das Flackern der Betriebs-LED bei einem Diskettenzugriff. Der Preis der Floppy beziffert sich auf 557,- DM. Zu beziehen ist das Laufwerk bei:

PDC, Bad Homburg
Tel. 0 61 72 / 247 48 (chk)



AMIGA **KICK START**

KICK START

PREMIERE
Die neue Dimension für Ihren AMIGA

...ung in Intuition ... Hardware für ...
...op Publishing auf dem A...

KICK START

Für Ihre Bestellung verwenden Sie bitte die beigefügte Bestellkarte.

KICK START

...00C - TV-Modulator - Textamat
...LI - Fonts - C - Basic - Sortieren
...per - Farbeditor - Brandneue Spiele

KICK START

...Video Scape 3D - DIGA - Farbdrucker - Hardwa
...Intuition - Basic - CLI - C - Kurs - MS DC
...Listings - BAM Copy - PUZZLE - ... - Neue Spiel

Die sollten Sie haben... wenn nicht, dann können Sie jetzt nachbestellen!

KICK START

AKTUELL - AKTUELL - AKTUELL

The 10th PERSONAL COMPUTER WORLD SHOW

London, 23. - 27. September 1987



Das COMMODORE-VILLAGE bei der diesjährigen PCW wurde vom Chelsea Fußball Team und seinem Manager John Hollins feierlich eröffnet. Wie Commodore bekanntgab, wurde mit dem Club von Chelsea ein Werbevertrag für drei Jahre abgeschlossen. Commodore zahlt dafür die Rekordsumme von 3.75 Mio DM - die bisher höchste Summe für einen Werbevertrag im britischen Fußball.

Damit ist neben Bayern München und Dynamo Kiew bereits der dritte europäische Traditionsverein mit Commodore verbunden. Aber nicht nur die Fußballer, viele Sportfreunde haben von der neuen Werbestrategie Commodores profitiert. Sehr große Summen sind besonders in den Amateursport wie z. B. Leichtathletik und Radfahren geflossen.

Profitiert hat natürlich auch Commodore selbst. Wie auf der PCW bekanntgegeben wurde, ist der Vorjahresverlust von 127,9 Mio US\$ nicht nur aufgefangen, sondern in einen Nettogewinn von 28,6 Mio US\$ verwandelt worden. Dies ist natürlich noch kein Grund in überschwenglichen Jubel auszubrechen. Den Löwenanteil daran hat vor allem der deutsche Markt eingebracht.

Entsprechend war auch die Bereitschaft der im COMMODORE VILLAGE vertretenen Firmen mit uns zusammenzuarbeiten. Im Moment schwimmen wir noch in einer Flut von Prospekten, Pressemappen, Vorankündigungen, Pressemitteilungen, Testexemplaren und eigenen Notizen. Bis wir Land sehen, werden noch ein paar Tage vergehen. Der Redaktionsschluß war auch schon vor zwei Wochen, aber wir haben es immerhin geschafft für dieses Heft noch eine Übersicht der wichtigsten Neuvorstellungen zusammenzutragen. Ausführlichere Informationen und erste Erfahrungsberichte zu den Produkten bringen wir aber der nächsten Ausgabe.

AACKOSOFT

Aackosoft hat für dieses und nächstes Jahr insgesamt 17 Amigatitel angekündigt. Die ersten sind Flight Deck, Battle Chopper und Hype. Hype ist erst das 97. Weltraumschießspiel dieser Saison und unterscheidet sich nur unwesentlich von seinen Vorgängern. Flight Deck, ein Flugzeugträger im Einsatz gegen Terroristen und Battle Chopper, eine Hubschraubersimulation, bringen dagegen neue Ideen auf den Amiga-Spielemarkt. Nur für MS-DOS Rechner sind vorläufig Police Academy 2 - das Spiel zum Film - und Indy 500 - das klassische amerikanische Autorennen um die 500 Meilen von Indianapolis - erhältlich. Nicht nur fahrerisches Geschick, sondern auch Cleverness und Strategie sind gefragt, denn Anzahl und Zeitpunkt der Boxenstopps haben einen großen Einfluß auf den Ausgang des Rennens.

AEGIS DEVELOPMENT INC.

Aegis präsentierte die PAL-Version von Aegis Videoscope 3D - komplett mit deutschem Handbuch. Ebenfalls neu als PAL-Version Aegis Video Titler und Aegis Draw Plus. Alle registrierten Benutzer können die neuen Versionen gegen Erstattung der Versandgebühr kostenlos umtauschen. Die Firma SOFTWARELAND aus Zürich übernimmt übrigens in Kürze eine Exklusivvertretung von AEGIS in Europa.

ACTIVISION

Mit Thexder, Police Quest, 3D Helicopter Simulator hat sich Activision auf der PCW gemeldet. Thexder ist ein weiteres Weltraumschießspiel.

Ziel ist es, einen Computer zu zerstören, der irgendwelche Monster produziert. Wo die Programmierer diese originellen Ideen immer nur her haben? Bei Police Quest wurden die realen Erfahrungen eines amerikanischen Polizisten bei seinem Einsatz als amerikanischer Rauschgiftfahnder verarbeitet. Das genaue Gegenteil eines Musterknaben ist Leisuresuit Larry. Seine Abenteuer in Las Vegas sind nur für Erwachsene. Spielen, feiern, tanzen und flirten führen den Benutzer zu „...the most fun you can have with both hands on the keyboard!...“ (siehe Spieleteil)

Die besondere Option des angekündigten Hubschrauber-Simulator ist die Zwei-Spieler Option im Combat-Modus. Ob er sich ansonsten wesentlich von anderen, angekündigten Produkten unterscheidet, war nicht zu beurteilen. Activision hat von allen angekündigten Produkten noch keine spielbaren Versionen vorgelegt.

Ebenfalls angekündigt, jedoch nur für MS DOS: Star Rank Boxing II - mit verbesserter Animation.

INFOCOM Adventures sind Kultspiele. Die Firma ist inzwischen im Besitz von ACTIVISION. Drei neue Abenteuer werden demnächst das Licht der Kaufhäuser erblicken:

Bejond Zork

- die Fortsetzung des Millionenhits

Plundered Hearts

- ein spezielles Adventure für Frauen

North and Bert

- acht kurze Geschichten

BUROCARE LTD

auf dem Burocare-Stand im Commodore Village wurden erstmals die Elproma Data Switches vorgestellt. Damit können bis zu 12 Peripheriegeräten verbunden werden. Ebenfalls präsentiert wurde Artisan, eine Graphik & Video Arbeitsstation für Amiga 2000. Das Besondere daran für Genlock-Fans ist, daß man nicht nur Text auf Bilder, sondern beides beliebig gemischt auf den Bildschirm bringen kann.

CAMERON

Der bereits für andere Rechner sehr bekannte Handy Scanner ist jetzt auch als Hardwarescanner für Amiga 500/1000 erhältlich.



AKTUELL - AKTUELL - AKTUELL

CATSPAW - NOAH SYSTEMS

Mit Snobola4+ wurde eine an der Universität von San Antonio entwickelte Sprache jetzt auch für MS-Dos Rechner vorgestellt. Sie soll laut Aussagen ihrer Entwickler leicht zu lernen und elegant zu programmieren sein. Ob sie hält, was versprochen wird, muß sich allerdings noch zeigen.

ELECTRONIC ARTS

die Produktpalette von Electronic Arts war wie immer beeindruckend. Direkt erhältlich für den Amiga war jedoch keine der Neuankündigungen. Diesen Winter werden noch erscheinen: Formula 1 Racing und Tau Ceti. Das Autorennen wurde bereits auf der CES in Chicago unter dem Arbeitstitel Ferrari Grand Prix vorgestellt, Tau Ceti ist die Umsetzung des bereits bekannten Spiels für den Amiga.

EPYX

Auch bei Epyx wurden die auf der CES präsentierten Titel nochmals vorgestellt. Neu war dabei Impossible Mission II. Epyx wurde von US Gold präsentiert und war diesmal nicht mit einem eigenen Stand vertreten.

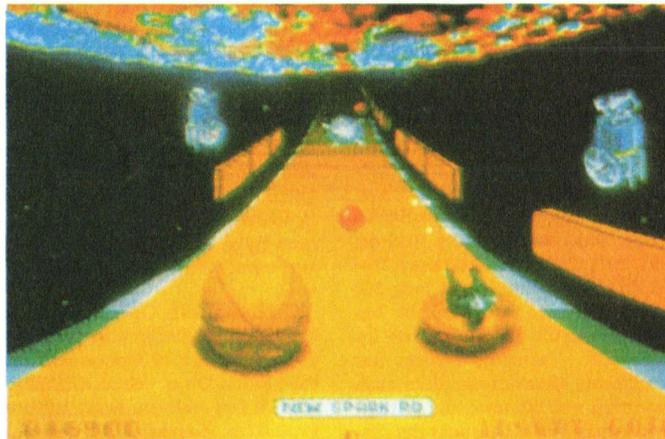
HAITEX RESOURCES

Memory Plus ist eine Multifunktionskarte für den Amiga 2000. Im Produkt-Überblick wird sie wie folgt beschrieben: 1 parallel port standard, 1 optional / 1 serial port / Control logic & socket for 68881 math coprocessor / Logic & sockets for 1 megabyte of ram using 256k d-ram / support software. Alle Informationen dazu gibt es momentan nur in Englisch. Ebenfalls von Haitex vorgestellt wurde Haicalc, ein Kalkula-

tionssystem. Es soll laut Herstellerangaben Multi-Tasking und Multi-Windowing voll unterstützen.

HBMLE (HABE Systems & HB Marketing)

Ein Textverarbeitungssystem wird von HBWL angeboten - Sunword. Auch hier ist der Hersteller überzeugt etwas ganz besonderes in puncto Benutzerfreundlichkeit und angebotenen Funktionen geschaffen zu haben. Welche Vorzüge all die neu vorgestellten Systeme zur Textverarbeitung haben und für welchen Anwenderkreis sie geeignet sind, werden wir nach einem Praxistest verraten. Richtig rund ging es bei der Präsentation von Firepower, dem neuesten Spiel von MicroIllusions. Näheres dazu war bereits in unserer letzten Ausgabe zu lesen.



LEVEL 9

Nach dem Erfolg von Knight Orc hatte sich Level 9 entschlossen, das

neueste Adventure auf einem eigenen Stand vorzustellen. Gnome Ranger ist die Geschichte von Ingrid, der Gnomdame, die von ihrer Familie in die Wildnis geschickt wird - zum Pech der dort hausenden Monster.

MASTERTRONIC

Mastertronic kündigt für den Amiga drei Spiele an: Ninja Mission, Feud und Space Ranger. Mehr war noch nicht zu erfahren.

MELBOURNE HOUSEK

Ebenfalls drei Spiele werden demnächst von Melbourne House erscheinen. Roadwars ist das Rennen zweier Kugeln auf dem Space-Highway um den Mond von Armageddon; bei Metropolis begeben Sie sich auf die Suche nach den verschwundenen Bewohnern der von Robotern kontrollierten Stadt. Bestens bekannt ist Rockford, der Comic-Held. Er durchstöbert die Caverns of Graymar auf der Suche nach den Äpfeln der ewigen Jugend.



MICRO APL LTD

Amiga APL - stellte die Umsetzung dieser Sprache für den Amiga vor. Zielgruppe sind Anwender, die Programme für Wirtschaft und Verwaltung schreiben. Der Einführungspreis wurde mit ca. 320 DM angegeben.

MICRODEAL

Irgendwie war drei diesmal die magische Zahl im Spielegeschäft. MicroDeal wurde somit seinen Verpflichtungen gerecht. Die drei Neuvorstellungen: Airball - das von Atari ST bekannte Labyrinthspiel mit dem Ball, der immer wieder aufgepumpt werden will; Amiga Soccer - das erste Fußballspiel für den Amiga - Erscheinungstermin: Winter - und Fright Night - ein Horrorspiel nur für Erwachsene. MicroDeal hat aber nicht alle neuen Projekte auf der Messe vorgestellt. Deshalb haben wir John Symes und seinem Team einen Besuch abgestattet. Mehr darüber im nächsten Heft.

MIRRORSOFT

präsentierte auf seinem Stand noch vier weitere Firmen: PSS, Spectrum-Holobytem Mindscape und FTL Games. Die Strategiespiele-Spezialisten PSS wurden von Mirrorsoft aufgekauft. Spectrum-Holobyte ist eine Tochtergesellschaft von Mirrorsoft. Mindscape vertreibt seine Titel in Großbritannien exklusiv bei Mirrorsoft, FTL hat sogar alle Europarechte abgeben. In der Bundesrepublik werden die Mindscape-Spiele bei Rushware vertrieben.

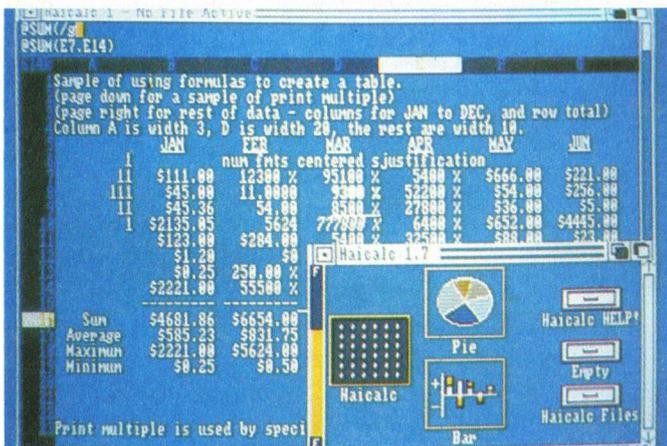
Mirrorsoft:
Bermuda Project - Adventure
Strike Force Harrier - Flugsimulator

PSS:
Power Struggle - Simulation der Politik der Supermächte
Mindscape Shadowgate, Nachfolger von Uninvited King of Chicago (von Cinemaware)

Spectrum-Holobyte:
Gato, der U-Boot-Simulator

KUMA

Kuma hat fast die ganze Produktpalette vom Atari ST jetzt auf den Amiga übertragen oder ist noch dabei. Dies sind: K-Seka, K-Spread2, K-Graph2, K-Word2, K-Comm2, K-Spell, K-Data, K-Roget und K-Minstrel. Erhältlich für den PC ist auch das K-Midi Interface Board.



AKTUELL - AKTUELL - AKTUELL

Für den PC noch angekündigt: Fighting Falcon. PT-109.

Annals of Rome (PSS) sowie der Fleet Street Editor, das Art Studio und die Executive Picture Show.

NINE TILES

Netzwerke jetzt auch für den Amiga gab es bei Nine Tiles zu bestaunen. CatenaNet werden wir demnächst einer genaueren Prüfung unterziehen.

NUCLEUS SOFTWARE SYSTEMS

GEFIX – ist ein supervisory control and data aquisition (SCADA) system für den Amiga 2000

OCEAN

insgesamt 19 neue Titel hat Ocean auf der PCW vorgestellt. Drei davon werden demnächst für den Amiga verfügbar sein. Army Moves, ECO und Wizball. ECO – ist eine Art Spiel des Lebens. Sie beginnen im Stadium der Amöbe und machen den Weg bis zum Menschen, falls Sie nicht vorher das Zeitliche segnen. Wizball – Wiz und seine phantastische Katze retten Wizworld. Ein Actionspiel um Magie und schnelle Reaktionen. Army Moves ist eine Simulation des taktischen Vorgehens der Landstreitkräfte in einem Verteidigungsfall. Ein Nachfolgespiel mit dem Titel Navy Moves ist bereits geplant. Übrigens – in einer der nächsten Ausgaben starten wir mit Ocean einen großen Wettbewerb, in dem es einen echten Spielhallenautomaten zu gewinnen gibt.

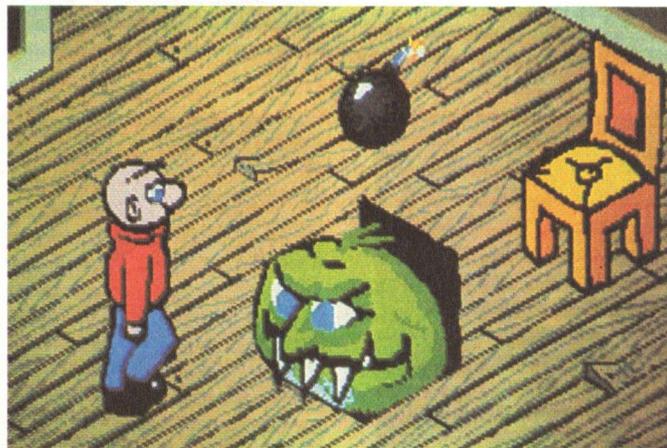
PRECISION SOFTWARE

Nach Superbase personal gibt es jetzt Superbase professional. Mehr Möglichkeiten und verbesserte Handhabung geben diesem Programm

durchaus Chancen im Bereich der professionellen Anwendung. Mit Digipic Frame Grabber kann man Videobilder im IFF-Format abspeichern und somit für zahlreiche Malprogramm nutzen.

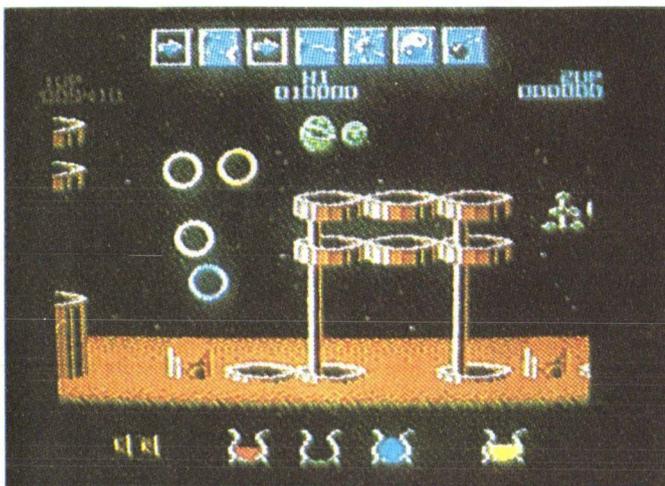
PROSPERO

Prospero Pascal for GEM und Prospero Fortran for GEM laufen jetzt auf jedem IBM oder kompatiblen Rechner. Die Pakete beinhalten u. a. Compiler, Linker, Symbolic Debugger etc. Insgesamt 750 Seiten Dokumentation werden ebenfalls mitgeliefert. Allein die Erstellung dieser Dokumentation hat übrigens beinahe ein Jahr gedauert.



PSYGNOSIS

Freunde der Eule werden diesmal mit dem neuesten Produkt aus dem Hause Psygnosis sicher zufrieden sein. Terrorpods ist ein Strategie- und Actionspiel von sehr großer Komplexität. Der Spieler muß nicht nur sich selbst und den Asteroiden Colian beschützen. Zusätzlich gilt es Handel zu treiben, um genügend Gewinn zu erwirtschaften, damit die Verteidigungsanlagen sich die teuren Waffen auch leisten können. Graphik, Sound und SCROLLING hinterlassen einen sehr guten Eindruck. (siehe Spielteil)



ROBTEK

Mit einem großen Verkaufsstand war diesmal auch Robtek anwesend. Unter den zahlreichen Neuankündigungen sind die 2.0 Megabyte RAM Erweiterung, Disk Library – ein Diskettenverwaltungsprogramm – und

Ein ebenfalls sehr interessantes Strategiespiel für MS Dos Rechner wird bereits ausgeliefert – Tracker. Das schöne dabei ist, daß der Computer erst anfängt Sie zu bekämpfen, wenn Sie ihn angreifen.

SENTINEL

Kaum ein anderes Textverarbeitungsprogramm wurde so gespannt erwartet wie Wrod Perfect Amiga. In London vorgestellt, wird es jetzt zeigen müssen, ob es hält, was sich die Käufer von Sentinel-Produkten versprechen.

STRATEGIC SIMULATIONS INC (SSI)

Nur ein einzelnes Fact-sheet war von SSI zu bekommen. Letzte Konfrontation mit dem dunklen LORD heißt das Schlußadventure der Trilogie Phantasie. Nikademos, der dunkle Lord droht in Phantasie II die Welt zu erobern. (siehe Spielteil)

SYSTEMS ARCHITECTS

Ein Architekturprogramm für Amiga 2000 stellte Systems Architects vor. Graphics Systems ist kein außergewöhnlicher Name, die zahlreichen Möglichkeiten, die dieses Programm dem professionellen Anwender bietet, sind es jedoch allemal.

TAURUS

Extensor, die Olympiade des dritten Jahrtausends auf dem Mars sicher erwähnenswert. Deutsche Distributoren wurden aber bei der PCW noch nicht genannt. Wir hoffen, daß wir darüber bald mehr berichten können.

Aquisition 1.3 – die neueste Version der Datenbank; XCAD-Designer – ein schnelles CAD für Amiga und City-Desk – ein weiteres Desktop Publishing Programm waren die Renner bei Taurus. Die Fähigkeiten von Aquisition haben wir bereits in der letzten Kickstart beschrieben. Besprechungen von XCAD-Designer und City-Desk werden noch folgen.

RAINBIRD

Die erfolgreiche Serie der Adventures aus dem Lande Kerovnia wird mit Jinxter fortgesetzt. Das Nachfolgedventure von Guild of Thieves beeindruckt wieder mit einer witzigen Spielidee und seiner guten Graphik. Carrier Command ist ein taktisches Kampfspiel im Stil von Balnace of Power. Es wird aber noch etwas Dauern bis die Amiga-Version fertig ist (Anfang 1988).

TRIANGLE

Ein Bonbon am Rande war Triangle TV – diese Firma präsentierte professionelle Anwendungen mit dem Amiga im TV-Bereich.

TYNESOFT

Alphabetisch, aber bestimmt nicht was die Qualität betrifft, am Ende steht Tynesoft. Recht überraschend für die Fachwelt präsentierte die mit MicroValue verbundene Firma zwei Neuheiten für das Weihnachtsgeschäft, von denen bisher noch nicht einmal gerücheweise etwas zu hören war. Winter Olympics 88 holt den Medaillenregen in Calgary auf den heimischen Monitor und Formula One Grand Prix wird eventuell doch noch das erste Autorennen für den Amiga, wenn sich die Konkurrenz mit den seit längerem angekündigten Rennspielen weiter so viel Zeit läßt.

Ein Actionspiel im Zeichenstil eines Comicheftes ist Dick Special. Die Sprites sind das größte, was der Amiga-Besitzer bisher auf seinem Rechner gesehen hat. Die vorgeführte Demo-Version war noch aus einem sehr frühen Stadium, so daß weitere Aussagen im Moment noch nicht möglich sind. Der bisherige Eindruck verspricht eine interessante Variante. Ein Sampler mit drei Adventures von Level 9 wird unter dem Titel Time and Magic angeboten. Stil und Graphik von Level 9 sind dabei unverkennbar.

Für einen guten Zweck Fußballturnier mit bekannten Computerfirmen

Daß Computerfirmen nicht nur mit Bits und Bytes jonglieren können, sondern auch Fußball spielen können, bewies das von der Firma Preh ausgetragene Turnier. Um so erfreulicher war, daß die gesamte Aktion einem guten Zweck diente, der Lebenshilfe.

In zwei Gruppen kämpften die, vorwiegend aus Hobbyfußballern bestehenden Mannschaften, um den Einzug ins Finale. Dabei ging es durchweg fair zu, die Schiedsrichter konnten sich freuen. Der einzige Verletzte, auf Seiten COMMODORES, griff nach kurzem Arztbesuch, wenn

Das Finale

Nach Meinung der Zuschauer ging ATARI als Favorit in diese Begegnung. Dies resultierte auch daraus, daß diese Mannschaft neben den sonstigen Spielern zwei erfahrene Bundesligaspieler in ihren Reihen hatten.

terstützung fanden die vermeintlichen Außenseiter aus ‚unverständlichen‘ Gründen in der Anhängerschar von COMMODORE.

In der Mitte der ersten Halbzeit fiel durch ein klassisches Abstaubertor das 1:0 für ATARI. Zur Freude



Die Endspielteilnehmer vor dem entscheidenden Spiel.



Von links nach rechts: H. Schneider, H. Engel, M. Fabian + M. Mayer (Verantwortlich für unsere Titelbilder), G. Wiesner, T. Hübner, M. Nerding und unten rechts C. Lippert.



Unser Torwart wurde schon beim Einschießen hart geprüft.

Der Hintergrund

Vor drei Jahren wurde diese Veranstaltung zu Gunsten der Lebenshilfe ins Leben gerufen. Nach dem Leitsatz ‚Menschen helfen Menschen‘ traten 8 Mannschaften bekannter Computerfirmen an, um einen Wanderpokal auszuspielden. Durch Spenden und Verlosungen kamen als Reinerlös bis zum Abend dieses Tages 5000 Dm zusammen, der der Lebenshilfe für geistig Behinderte in Bad Neustadt/Saale zufließt.

Das Turnier

An dieser Veranstaltung beteiligten sich acht Mannschaften:
PREH (Bad Neustadt, der Vorjahressieger)
COMMODORE (Frankfurt)
STAR (Eschborn)
SIEMENS (Würzburg)
HEWLETT PACKARD (Bad Homburg)
ATARI (Rauheim)
VOGEL-VERLAG (Würzburg)
MERLIN Computer (Eschborn)

auch nicht mit dem Fuße, so doch mit lautstarker Stimme, kurze Zeit später wieder in das Spielgeschehen ein.

Aus dieser recht stürmischen vorrunde gingen die Mannschaften von SIEMENS und ATARI als Gruppensieger hervor und bestritten das Finale. Besser dran waren da die unterschiedlichen Mannschaften, darunter unser Team, die sich, anstatt noch weitere 30 Minuten auf dem Platz kämpfen zu müssen, genüßlich mit kühlen Getränken am Spielfeldrand von den Strapazen erholten.

Die KICKSTART-Redaktion, die einen großen Teil der MERLIN-Computer Mannschaft stellte, schlug sich tapfer, insbesondere unser Tormann, der so manche gute Chance verrietete. Bei dieser Gelegenheit stellen wir uns auch einmal bildlich vor.

Zum einen EX-Nationalspieler Lothar Emmerich und Horst Trimhold, ehemaliger Spieler vom FSV Frankfurt, die beide zur Zeit beim FC Rhein Main mit Bernd Hölzenbein & Co ihrer Leidenschaft nachgehen. Nicht ganz fair, denn dadurch kann eine gewisse Überlegenheit gegenüber den übrigen Mannschaften nicht verleugnet werden.

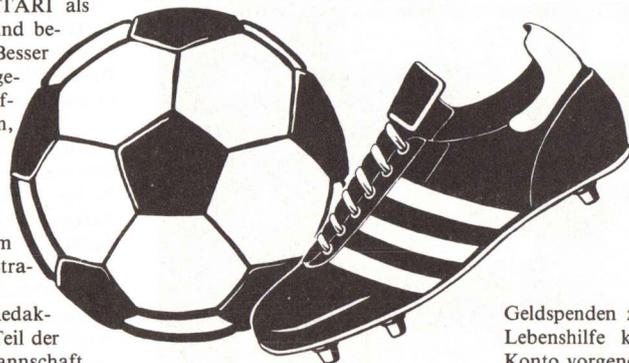
Es bleibt zu hoffen, daß nächstes Jahr Commodore nicht seinen Wer-

der Zuschauer gelang SIEMENS kurz vor der Pause durch einen Gewaltschuß der Ausgleich, (in der Fachsprache könnte man den Schuß als Langwortpoke direkt unter die Basepage bezeichnen).

In der zweiten Halbzeit kontrollierte ATARI das Spiel und schmutzte noch zwei Bälle zum Endstand von 3:1 in das Tor des wacker stehenden Torwarts.

Die Mannschaft von COMMODORE hat für das nächste Jahr schon jetzt Revanche angekündigt und will ATARI auf eigenem Platz den Titel streitig machen. Bis dann

(HS)



partner FC Bayern München zur Hilfe holt

Den bekanntesten Namen trotzend stand das Publikum hinter dem Team von SIEMENS. Weitere Un-

Geldspenden zur Unterstützung der Lebenshilfe können auf folgendes Konto vorgenommen werden.

Sparkasse Bad Neustadt,
Kto.-Nr. 4242 (BLZ 793 530 90)
Stichwort:
„PREH-Fußball-Turnier 5.9.1987“

C-KURS TEIL 3

Der dritte Teil des C-Kurses beschäftigt sich mit Kontrollstrukturen. Kontrollstrukturen sind Sprachelemente mit deren Hilfe man ein Programm von bestimmten Einflüssen abhängig machen kann. Dies können äußere Ereignisse sein, wie z.B. ein bestimmter Tastendruck oder auch innere, wenn z.B. eine bestimmte Uhrzeit erreicht ist. Die Sprachelemente, die wir hier besprechen wollen, erlauben es Ihnen Programmteile in Abhängigkeit von solchen Ereignissen abzuarbeiten. Ein typisches Beispiel ist die Reaktion des Amiga Dos Editors ED, wenn nach dem Abändern eines Textes die Tasten <ESC> Q gedrückt werden. In diesem Fall gibt das Programm folgende Meldung aus: Edits will be lost. Type y to confirm. Das Programm gibt Ihnen die Möglichkeit den nun folgenden Programmablauf von außen zu kontrollieren. Aus der Sicht des Programmes (des Programmierers) sieht eine solche Entscheidungsmöglichkeit folgendermaßen aus:

```
wenn Ereignis
dann mache ...
  ( Programmteil für
    Ereignis eingetroffen )
andernfalls ...
  ( Programmteil für
    Ereignis nicht
    eingetroffen )
```

Ein Sprachelement, das eine solche Entscheidung zuläßt ist die 'if' Anweisung, ein in fast allen Sprachen vorhandenes Element. Programmierer, die von Basic oder Pascal umsteigen, werden starke Parallelitäten feststellen.

Basic:

```
...
50 if Passwort$ = "geheim" then
    goto 70
60 new
70 Rem Programmstart
...
```

oder Pascal:

```
...
if flag = true then begin
    ...
end;
...
```

Der Syntax für die 'if'Anweisung in C entspricht fast der der Sprache Pascal. In einer etwas formalen Schreibweise sieht die 'if' Anweisung in C folgendermaßen aus:

```
if(Ausdruck)
  Befehl A
else
  Befehl B
```

Wie Sie erkennen, verfügt auch C über ein 'else'. Diese Sprachkonstruktion erlaubt das oben abgedruckte 'andernfalls'. Das 'else' ist, wie in vielen anderen Sprachen auch, ein optionales Sprachelement. Dies bedeutet, daß die 'else' Anweisung nicht zwingend auf ein 'if' folgen muß. Wir werden dies gleich noch näher besprechen.

Gehen wir zuerst einmal auf die einfachste Form einer Programmkonstruktion ein, in der die 'if' Anweisung verwendet wird. Nehmen wir an, wir wollen feststellen, wann der Wert der Variablen Namens "n" die Zahl fünf erreicht. Dieses Ereignis

soll auf dem Bildschirm ausgegeben werden. Ein Programmteil, der diese Aufgabe erfüllt, ist hier abgedruckt:

```
if(n = 5)
  printf("n ist jetzt 5 n");
```

Innerhalb der nach 'if' folgenden Klammern erfolgt eine Operation, die von 'if' bewertet wird. Liefert diese Operation den logischen Wert wahr, wird die Anweisung nach der Klammer ausgeführt.

Dem Programmierer stehen folgende Möglichkeiten zur Verfügung:

a < b	a kleiner b
a > b	a größer b
a <= b	a kleiner gleich b
a >= b	a größer gleich b
a = b	a gleich b
a != b	a ungleich b

Eine Besonderheit ist die Schreibweise des 'gleich' Operators mit '='. C unterscheidet hier durch das doppelte Gleichheitszeichen von der Wertzuweisung durch das einfache Gleichheitszeichen.

Wenden wir uns nun einer Konstruktion zu, die den optionalen 'else' Teil enthält. Wir erweitern dazu die Anforderungen an das obige Programmstück und fordern nun eine zusätzliche Meldung, wenn der Wert der Variablen "n" ungleich fünf ist. Schreiben wir zunächst einmal einen Programmteil, der nur die 'if' Anweisung verwendet:

```
if(n = 5)
  printf("n ist jetzt fünf\n");
if(n != 5)
  printf("n ist nicht fünf\n");
```

Obwohl es klar scheint, das nicht beide Bedingungen auf einmal erfüllt werden können, muß der Rechner beide Vergleiche ausführen, um den Inhalt der runden Klammern mit 'if' bewerten zu können. Eine solche Situation innerhalb eines Programmes kann mit einer 'if' - 'else' Konstruktion eleganter gelöst werden:

```
if(n == s)
printf("n ist 5\n");
else
printf("n ist nicht 5\n");
```

Diese Art der Programmierung sieht nicht nur eleganter aus, sie spart auch noch Zeit, wenn nämlich die erste Bedingung erfüllt ist und somit weitere Abfragen entfallen.

Eine besondere Art der 'if-else' Programmierung ist die sogenannte 'if-else' Kette. In ihr werden beliebig viele 'if' Anweisungen geschachtelt. Folgendes Beispiel soll diese Technik einmal verdeutlichen.

```
main()
{
char c;

c = getchar();
if(c == '0')
printf("Null\n");
else if(c == '1')
printf("Eins\n");
else if(c == '2')
printf("Zwei\n");
else if(c == '3')
printf("Drei\n");
else if(c == '4')
printf("Vier\n");
else if(c == '5')
printf("Fünf\n");
else if(c == '6')
printf("Sechs\n");
else if(c == '7')
printf("Sieben\n");
else if(c == '8')
printf("Acht\n");
else if(c == '9')
printf(„,Neun\n");
}
```

Wenn Sie das Programm kompiliert haben und starten, müssen Sie eine beliebige Taste und <Return> drücken. Das Programm liest diese Taste über die Funktion getchar() ein. Anschließend wird beim Durchlauf der 'if-else' Kette überprüft, ob

das eingegebene Zeichen eine Ziffer von 0 bis 9 war. Das Ergebnis wird als Wort durch die Funktion printf() ausgegeben.

Dieses Programm hat jedoch noch einen kleinen Fehler. Gibt der Anwender keine Ziffer ein, erhält er keine Reaktion von Seiten des Programmes. Abhilfe schafft ein weiteres 'else', das hinter dem letzten 'else if' eingefügt wird:

```
else if(c == 9)
printf("Neun\n");
else
printf("Keine Ziffer\n");
}
```

Man kann also das letzte 'else' verwenden, um falsche Eingaben abzufangen oder Fehlermeldungen auszugeben. Auch kann man über solche Konstruktionen bestimmte Unterprogramme aufrufen:

```
printf(" ... Hauptmenü ...
n");
printf(" n");
printf(" <i> für
Unterprogramm1\n");
printf(" <2> für
Unterprogramm2\n");
printf(" <3> für
Unterprogramm3\n");
printf("eine andere Taste für
Ende\n");

c = getchar();
if(c == '1')
unterprogramm1();
else if(c == '2')
unterprogramm2();
else if(c == '3')
unterprogramm3();
else
{
printf("Vielen Dank und
Tschüß");
break;
}
} /* Ende for() */
} /* Ende main() */
```

Die hier aufgetauchte Anweisung 'break' hat die Aufgabe die durch for(;;) als endlos programmierte Schleife abubrechen. Die 'break' Anweisung unterbricht den momentan laufenden Programmteil. Im Beispiel oben veranlaßt die 'break' Anweisung das Programm, mit dem ersten Befehl nach der for(;;) {...} Schleife

fortzufahren. (An dieser Stelle steht im Beispiel oben der Kommentar 'Ende for ()').

Eine weitere Möglichkeit eine Auswahl aus mehreren zu treffen ist die 'switch' Anweisung. Pascalprogrammierer werden diese Konstruktion folgendermaßen in Erinnerung haben:

```
Pascal
case c of
'1': writeLn('Eins');
'2': ...
end
```

Der Aufbau einer solchen Konstruktion in C sieht so aus:

```
switch(c)
{
case '1':
Aktion für c = 1
case '2': Aktion für c = 2
default: Aktion für c nicht
1 oder 2
}
```

Die Option 'default' verhält sich analog zum oben besprochenen 'else' am Ende einer 'if-else' Kette. Sie fängt den Programmablauf ab, für den Fall, daß kein 'case' ausgeführt wurde.

Nun unterscheidet sich die 'switch' Anweisung in einigen Fällen von den 'if-else' Ketten. Im Gegensatz zu der Konstruktion mit 'if' und 'else', bei der der Programmierer die Variable auch mit Werten vergleichen kann, die erst während der Programmausführung festgelegt werden (z.B. andere Variablen), arbeitet die 'switch' Anweisung nur mit Konstanten. D.h. die Werte nach denen der Programmierer seine Entscheidungen im Programm treffen will, müssen vor dem Übersetzen feststehen. Schauen wir uns das obige Beispiel noch einmal in der Programmierung mit 'switch' an:

```
main()
{
char c;
c = getchar()
switch(c)
{
case '0': printf("Null\n");
break;
case '1': printf("Eins\n");
```

```

        break;
    case '2': printf("Zwei\n");
              break;
    case '3': printf("Drei\n");
              break;
    case '4': printf("Vier\n");
              break;
    case '5': printf("Fünf\n");
              break;
    case '6': printf("Sechs\n");
              break;
    case '7': printf("Sieben\n");
              break;
    case '8': printf("Acht\n");
              break;
    case '9': printf("Neun\n");
              break;
    default: printf("Keine
                  Ziffer\n");
              break;
} /* Ende switch */
} /* Ende main */

```

Das explizite 'break' nach jedem 'printf' veranlaßt das Programm (sofort die 'switch' Anweisung zu verlassen) nachdem das erste 'case' zutrifft. Man kann jedoch das 'break' auch mit Absicht weglassen. Ein solches Programm könnte folgermaßen aussehen:

```

main()
{
    char c;

    c = getchar();
    switch(c)
    {
        case '0':
        case '1':
        case '2':
        case '3':
        case '4':
        case '5':
        case '6':
        case '7':
        case '8':
        case '9':
            printf("0-9\n");
            break;
        default:
            printf("Keine Ziffer\n");
            break;
    }
}

```

Wie Sie sehen, können so verschiedene Fälle zusammengefaßt werden. Wichtig ist jedoch auch hier das explizite 'break' am Ende eines jeweiligen Entscheidungspfades. Dies liegt darin begründet, daß die durch 'case' definierten Marken nur als Textmarkierungen für den Compiler zu sehen sind. D.h. wenn die Variable den Wert '2' hat, ist mit der Programmausführung ab der Textmarke 'case '2':' fortzufahren: Dies schließt jedoch auch alle Anwendungen hinter späteren Marken ein. Ein analoges Beispiel ist die

on a goto 50,60,70

Anweisung in Basic. Auch dort würde nach einer Auswahl der Zeile 50 mit 'on a goto' die Zeile 60 mitabgearbeitet werden. Soll dies vermieden werden, muß in Basic am Ende der Zeile z.B. ein 'goto 80' stehen, das die Programmausführung an dieser Stelle fortsetzt.

Befassen wir uns noch einmal mit der 'break' Anweisung, sowie dem äußerst seltenen 'continue'.

Eine 'break' Anweisung verläßt immer die innerste Schleife, die sich gerade in der Bearbeitung befindet, also die Schleife in der auch die Anweisung steht:

```

for(;;)
    for(i = 1; i < 10; i++)
    {
        c = getchar();
        if(c == 'e') break;
    }

```

Das 'break' wird in diesem Beispiel die innere 'for' Schleife beenden, wenn der Anwender 'e' eingibt. Die äußere 'for(;;)' Schleife bleibt jedoch unbeeinflusst, so daß dieses Programm kein Ende finden wird.

Die 'break' Anweisung arbeitet mit folgenden Konstruktionen zusammen:

```

for(...)
{
    ...
    break;
    ...
}

```

while (...)

```

{
    ...
    break;
    ...
}

```

```

do
{
    ...
    break;
    ...
}while(...)

```

```

switch(...)
{
    ...
    case ... : ...
                break;
    ...
}

```

Die seltenere 'continue' Anweisung bewirkt das Gegenteil der 'break' Anweisung. Anstatt eine Schleife zu verlassen, wird diese neu gestartet. Praktisch heißt dies, daß 'continue' das Programm an der Stelle fortführt an der die Schleife beginnt in der ein 'continue' angefallen wurde. Das folgende Beispiel liest fünf Zahlen ein. Ist eine dieser Zahlen größer als zehn, beginnt der Ablauf von vorne.

```

for (i = 0; i < 5; i++)
{
    if(a[i] > 10)
        continue;
    ...
}

```

Die 'continue' Anweisung arbeitet jedoch nur mit Schleifen zusammen. D.h. es können alle oben aufgelisteten Fälle auch für 'continue' auftreten, außer in dem Fall mit 'switch'.

Sie haben sicher die Variablenbezeichnung '[i]' im obigen Beispiel bemerkt. Diese Art der Variablen nennt man Arrays. Diejenigen unter Ihnen die schon in Basic oder Pascal Erfahrung haben, werden diese Variablenart sicher schon kennen. Wir werden dieses Thema in der nächsten Folge besprechen. Bis dahin wünsche ich Ihnen viel Erfolg.

DIGIPAIN^T

Malprogramme sind zweifellos eine der Domänen des Amiga, der zu allen grafischen Anwendungen ja prädestiniert ist. Eines dieser Programme, das die Redaktion in letzter Zeit erreichte, nennt sich Digipaint. Es bietet genug interessante Features, um es einmal etwas eingehender zu behandeln.

Bei Digipaint handelt es sich um ein recht neues Malprogramm, das im HAM-Modus (also mit 4096 Farben) arbeitet und sowohl in der Auflösung 320★200 als auch 320★400 (Interlaced) gefahren werden kann. Wie schon der Name suggeriert, bietet es besondere Möglichkeiten zum Bearbeiten digitalisierter Bilder. Zum Lieferumfang von Digipaint gehört außer einer Diskette ein knapp 60-seitiges Handbuch in englischer Sprache, das sämtliche Optionen des Programms knapp, aber ausreichend und verständlich diskutiert und außerdem noch mit ein paar Anleitungen aufwartet, mit deren Hilfe man die Anwendung verschiedener Funktionen von Digipaint schnell und anschaulich erlernt.

Startet man Digipaint, so erscheint nach kurzer Ladezeit zuerst einmal ein Fenster, das den Anwender dazu



Bild 1:
Lady in
Schwarz-weiß...

auffordert, ein bestimmtes Wort aus einer zufällig gewählten Zeile und Seite des Handbuchs einzugeben. Diese Art von Kopierschutz erscheint recht fragwürdig, und sie erweist sich für Anwender, die ihr Handbuch bereits eingehend studiert und daher nicht unbedingt griffbereit haben, sogar als lästig.

Die Funktionen von Digipaint

Das eigentliche Programm meldet sich mit einem schwarzen Screen und der sogenannten Toolbox, die das untere Drittel des Bildschirms einnimmt. Sie besteht aus zwei Menueleisten und einer Ansammlung von

Gadgets, welche dem Anwender verschiedene Funktionen offerieren. Neben den üblichen Standardfunktionen eines Malprogramms, als da wären verschiedene Brushes, gefüllte Kreise und Rechtecke, Linien und auch ein Spraydoseneffekt, findet man hier eine Palette mit 16 Farben, die Digipaint ohne fließenden Übergang darstellen kann, was ein besonderes Feature im HAM-Modus darstellt, denn normalerweise können hierin keine scharfen Übergänge von einer Farbe zur anderen dargestellt werden. Dies gilt auch bei Digipaint für die 4080 verbleibenden Farben. Die Farben in der Palette sind übrigens frei wählbar. Desweiteren findet man drei Colormenues, auf denen die möglichen Farbschattierungen zu sehen sind und aus denen man eine gewünschte Farbe direkt anwählen kann, und drei RGB-Schieber, mittels derer man seine Zeichenfarbe exakt justieren kann.

Auf der Menueleiste finden sich anfangs verschiedene Gadgets, mit denen man die Funktionen PickColor, Undo, Again, CopyColor, Clear und Fill anwählen kann. Mittels PickColor wählt man seine Zeichenfarbe, CopyColor dient zum Kopieren einer beliebigen Farbe in die Palette.

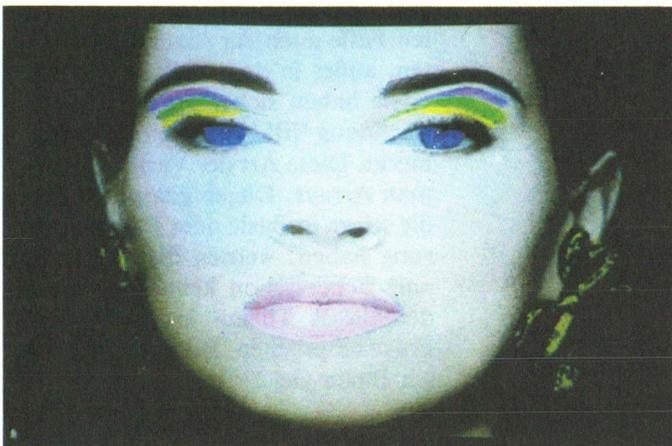


Bild 2:
...und geschminkt in
vielen Farben

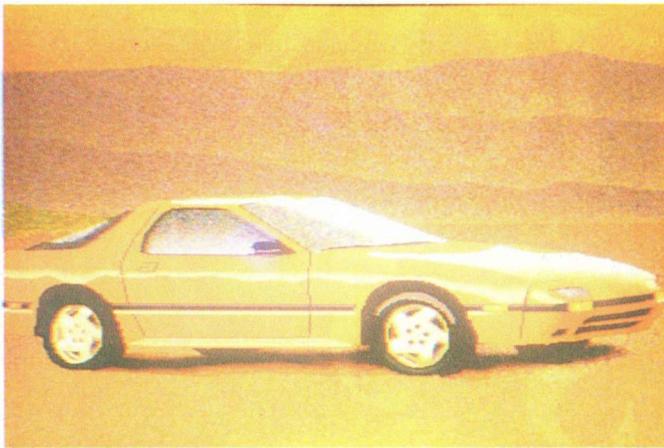


Bild 3: Shadings in vielen Variationen

Again erweist sich als sehr nützlich, wenn man eine Operation wiederholen möchte, nachdem man die Zeichenfarbe gewechselt hat. Die anderen Funktionen erklären sich mit ihren Namen von selbst. Mit einem Klick auf den rechten Mausknopf offenbart sich dem Anwender ein Menü mit den Optionen Picture, Brush, Effect, Mode und Prefs. Mittels des Picture-Menues lassen sich Bilder laden, speichern und ausdrucken. Außerdem steht die Option Swap zur Verfügung, die es ermöglicht, ein zweites, im Hintergrund liegendes Bild zu bearbeiten. Brush dient zum Laden und speichern von IFF-Brushes, die auch mit einem anderen Malprogramm erstellt worden sein können. Effects bietet dem Anwender Optionen wie Bildvergrößerung, Bildverkleinerung, Spiegelungen und Weichzeichnen eines Bildes. Das interessanteste dieser Menues stellt jedoch das Mode-Menue dar.

Schattierungen und Überlagerungen

Seine große Stärke spielt Digipaint beim Einsatz verschiedener Fillmodi, von denen nicht weniger als 12 Stück existieren, aus. Sie erlauben die mannigfaltigsten Arten der Farbüberlagerung, indem auf verschiedene Art und Weise bereits gesetzte Farben und neu zu zeichnende miteinander verknüpft werden. So ist es zum Beispiel möglich, ein digitalisiertes Schwarzweißbild nachträglich einzufärben, wobei die Schattierungen des Originalbildes erhalten bleiben. Auf diese Art können faszinierend verfremdete oder auch sehr natürlich wirkende Bilder erstellt werden, was ganz dem Anwender überlassen bleibt. Es gibt die Möglichkeit, Farbwerte von Hinter-

grund- und Zeichenfarbe logisch zu verknüpfen, man kann sich aber auch einfach Mischfarben erzeugen lassen, die ganz nach Wunsch mehr zum Hintergrund oder mehr zur Zeichenfarbe tendieren. Desweiteren lassen sich Schattierungen in verschiedenen Variationen erzeugen, womit man beispielsweise sehr realistische Kugelfeffekte erzeugen kann. Auch der Effekt der Verwischung läßt sich stufenlos einstellen, wodurch Schattierungen – je nach Einstellung – entweder scharf abgegrenzt oder aber als fließender Übergang dargestellt werden. Alles in allem sind die Möglichkeiten, die Digipaint mittels seiner Fillmodi bereitstellt, derart mannigfaltig, daß sie im einzelnen hier nicht genau erläutert werden können. In dieser Hinsicht ist Digipaint anderen Malprogrammen deutlich überlegen.

Obwohl Digipaint nur in der niedrigen horizontalen Auflösung von 320 Pixeln arbeitet, ist es möglich, mit anderen Malprogrammen im Hires-Modus erstellte IFF-Bilder einzuladen. In diesem Falle konvertiert Digipaint diese Bilder, so daß sie ohne Verzerrungen auf dem Bildschirm erscheinen und nachbearbeitet werden kön-

nen. Dies ist sowohl mit 16- als auch mit 32farbigen Bildern möglich.

Digipaint kann sicher nicht mit derart vielen Features aufwarten, wie beispielsweise DeluxePaint das kann, aber das entspricht sicher auch nicht dem Gedanken, der hinter Digipaint steht. Wie schon zuvor erwähnt, liegen die besonderen Stärken von Digipaint in den vielfältigen Fillmodi, die sich besonders beim Bearbeiten von digitalisierten Bildern als nützlich erweisen. Hier hat Digipaint eindeutig die Nase vorn. Aber auch bei „normalen“ Bildern sind diese Möglichkeiten nicht zu verachten, und es ist schon etwas anderes, mit 4096 Farben arbeiten zu können, denn hier ergeben sich Möglichkeiten der Bildgestaltung, die mit anderen Programmen nicht oder nur mit sehr viel Mühe zu verwirklichen sind. Außerdem erweist sich Digipaint als sehr einfach und komfortabel zu bedienen, was bei der Benutzung viel Freude macht. Anwender, die gerne einmal ein Malprogramm benutzen, sollten sich Digipaint daher auf jeden Fall ansehen. Recht erfreulich ist bei diesem Programm der – relativ – niedrige Preis: Digipaint wird empfehlenermaßen für ca. DM 198,- im Handel zu erhalten sein.

(WD)

Übersicht der Vor- und Nachteile

+ HAM-Darstellung
+ außergewöhnliche Fillmodi
+ verarbeitet auch fremde Bildformate

– etwas eingeschränkte Funktionsvielfalt

Anbieter: DTM, IM, PDC

Preis: ca. 198,- DM

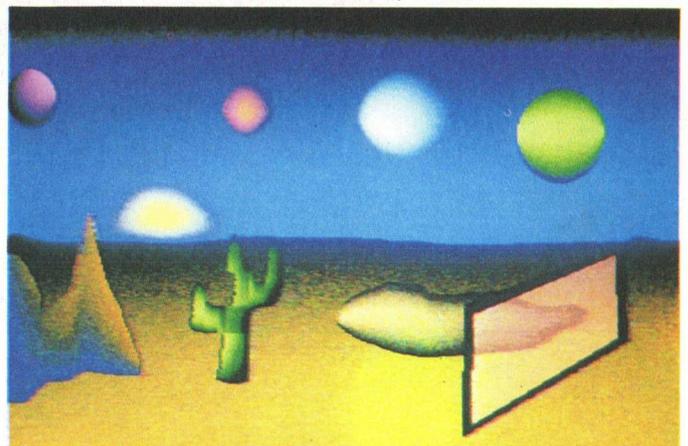


Bild 4: Solche Farbverläufe gibt es nur im HAM-Mode

INTUITION- KURS 5

Requester

Requester sind eine weitere Intuitionskomponente, man kann diese als eine Art Menue bezeichnen, jedoch nicht im herkömmlichen Sinne, oder gar im Vergleich zu Pull-Down Menues. Nein, Requester sind Felder in denen im Normalfall verschiedene Gadgets plaziert werden, man kann sie auch als Dialogfelder bezeichnen, welche dann vom Benutzer angewählt werden können. Eine gewisse Ähnlichkeit mit Fenstern ist durchaus gegeben, aber Requester kann oder sollte man eher als eine bestimmte Weiterentwicklung von Fenstern sehen, in Bezug auf Dialogfelder versteht sich, ansonsten kommen Requester nicht an die Möglichkeiten von Fenstern heran.



Rückblick

Auch zu Beginn von diesem Kurs teil möchte ich mit einem kleinen Rückblick starten. In der letzten Ausgabe von KICKSTART (Ausgabe 10) ging ich ausführlich auf die Programmierung von Pull-Down Menues ein. Davor waren die Kursthemen Screens, Windows und Gadgets. Die Handhabung und die verschiedenen Möglichkeiten der Programmierung waren in diesen Kapiteln der Inhalt. Dabei wurde zum Teil auf die Programmiersprache AmigaBASIC zurückgegriffen, hauptsächlich wurde aber 'C' genutzt. Das Verwenden dieser Sprache liegt klar auf der Hand, alle Intuitionskomponenten können in dieser Sprache in allen von Intuition bereitgestellten Eigenschaften programmiert werden. In AmigaBASIC ist das nicht der Fall. Die Requester Programmierung wurde in dem Befehlssatz gar nicht implementiert. Komplizierte Wege müssen zur Nutzung von Requestern In AmigaBASIC begangen werden. Der Intuition Kurs 5 greift nun Requester auf, deren Einsatzmöglichkeiten sowie deren Programmierung ich erklären werde.

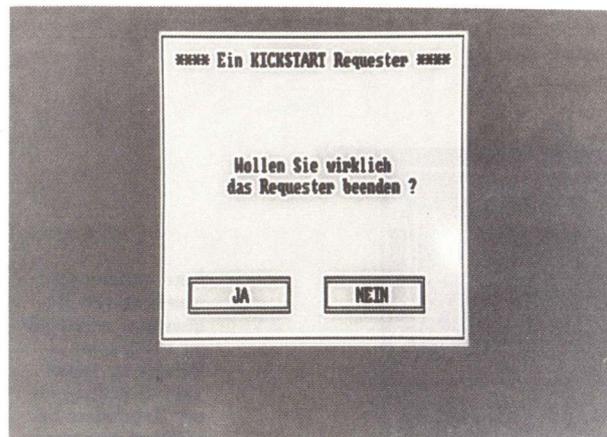
Alles über Requester

Vorneweg, Intuition unterscheidet zwei Arten von Requestern, zum einen die sogenannten Autorequester. Das Betriebssystem nutzt diese Art von Requestern zur Ausgabe von Fehlermeldungen. Sicherlich hat Ihr Amiga schon einmal in der linken oberen Ecke ein kleines Fenster geöffnet, in dem er Sie aufgefordert hat den Schreibschutz zu entfernen, oder eine bestimmte Diskette in das Laufwerk zu legen. Unterhalb der Meldung waren immer zwei Gadgets plaziert, die 'Cancel' oder 'Retry' beinhalteten. Mit 'Cancel' konnten Sie die Meldung unterbrechen und der Rechner brach die Durchführung seiner momentanen Aufgabe ab. Mit 'Retry' prüfte der Rechner ob der Fehler behoben ist, wenn ja, führt er die jeweilige Aufgabe weiter aus, wenn der Fehler nicht behoben wurde, wird das selbe Requester (Systemrequester) erneut erzeugt. Die kleinen

'Retry' und 'Cancel' Felder sind übrigens Gadgets des Typs Boolean.

Software Error, Finish all Disk activity...

Solche Systemrequester sind in der Regel recht harmlos, anders Verhält es sich mit einem anderen. Erscheint ein Requester in der linken, oberen Ecke mit der Meldung 'Software Error, Finish all Disk Activity...' sind die Folgen schon schlimmer. Beim Anwählen des 'Cancel' Gadgets stürzt der Rechner ohne Erbarmen ab und es muß neu gestartet werden. Durch auswählen des 'Retry' Gadgets verschwindet das Requester kurzzeitig und erscheint dann wieder. Bild 1



Der am meisten gefürchtete System Requester, hier in einer kleinen Spielerei. Es wurden dazu einige Tasks zum Absturz 'überredet'.

zeigt ein Requesterwirrwarr auf dem Bildschirm, hierbei wurden einige Tasks zum Absturz bewegt. Doch zur Begründung: Diese Meldung wird vom System ausgegeben, wenn ein Task seinen Geist aufgegeben hat, was soviel bedeutet, daß ein irreparabler Fehler der Software diesen Task zum Absturz gebracht hat. Man muß jedoch nicht verzweifeln, hat man das Programm von der Workbench aus gestartet, also durch zweimaliges Anklicken mit dem Mauszeiger, kann der Benutzer die Meldung erst einmal ignorieren und ein anderes Programm starten, oder eine andere Aufgabe durchführen. Das liegt daran, wie schon erwähnt, daß nur ein Task nicht mehr seine Arbeit verrichtet, andere Tasks können aber durchaus noch genutzt werden. Um die Sache besser zu verstehen sollte man wissen, daß alle Programme, die von der

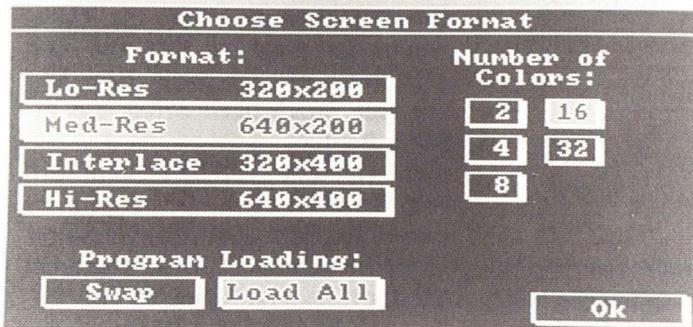
Workbench aus gestartet werden als Task laufen. Von der CLI Oberfläche (Befehlseingabe über die Tastatur) ist dies nicht der Fall. Die Ausgabe eines solchen Requesters läßt nur eine Möglichkeit offen, das erneute Booten.

Die zweite Art von Requestern wird vom Betriebssystem überhaupt nicht verwendet, wird aber von vielen Programmen gebraucht. Das Programm DPaint II benutzt Requester zum Laden von Bildern oder Brushes, aber auch zum Einstellen der Bildschirmauflösung, siehe dazu Bild 2 und 3. Es handelt sich hierbei um eine Art Fenster, in dem Gadgets und Bilder plaziert werden können. Klar, das kann in einem Fenster auch geschehen, aber Requester haben einen besonderen Vorteil, in Hinblick zum Betriebssystem. Wird ein Requester aufgerufen, unterbricht das Pro-

gramm den Ablauf und wartet bis eine Eingabe gemacht ist, erst dann wird das Programm fortgesetzt. In einem Requester muß immer eine Eingabe erfolgen. Bei einem Pull-Down Menue braucht nicht unbedingt eine Eingabe gemacht zu werden. Requester sind beweglicher als Menues, sie können überall auf dem Bildschirm oder in einem Fenster plaziert werden, demnach direkt neben dem betroffenen Objekt. Kleine Unstimmigkeiten treten auf, wie sie auch bei Pull-Down Menues des öfteren vorkommen.

Double Menue Requester

Schließlich existiert noch eine dritte Art von Requestern, das Double

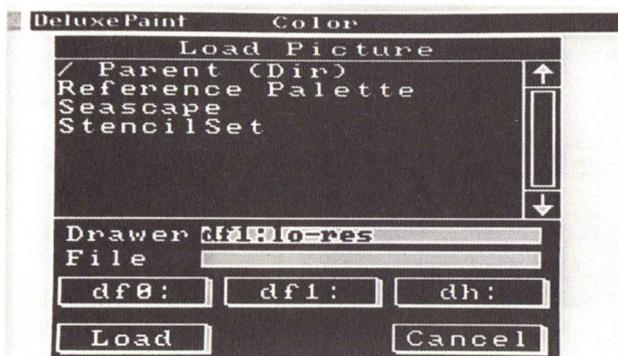


Die Anwendung eines Requesters im Programm DPaint II zeigt das Auswahlmenü der Bildschirmauflösung.

Menue Requester. Dieses ist in allen Belangen identisch mit allen anderen Requestern, bis auf eine Ausnahme. Es kann nur mit einem doppelten Anwählen der Maus aufgerufen werden. Ist ein solches Requester ausgegeben, besitzt es Priorität gegenüber allen anderen Requestern.

Requester Programmierung

Da in AmigaBASIC keinerlei Befehle zu Requestern implementiert sind, gehe ich ohne Umschweife gleich



Eine weitere Anwendung von Requestern, wiederum im Programm DPaint II. Diese Verwirklichung eines Requesters bezeichnet man gerne als Programm-Auswahl Box.

Die Requester Struktur:

```
struct Requester
{
    struct Requester *OlderRequester,
    SHORT LeftEdge;
    SHORT TopEdge;
    SHORT Width;
    SHORT Height;
    SHORT RelLeft;
    SHORT RelTop;
    struct Gadget *ReqGadget;
    struct Border *ReqBorder;
    struct IntuiText *ReqText;
    USHORT Flags;
    UBYTE Backfill;
    struct ClipRect ReqCRect;
    struct BitMap *ImageBMap;
    struct BitMap ReqBMap;
};
```

auf die Programmierung in C über. Eines soll aber noch gesagt werden, Requester können in AmigaBASIC erstellt werden, jedoch müssen Sie hierzu die Systemaufrufe nutzen (nichts anderes wird auch in C gemacht).

Auch an dieser Stelle möchte ich wieder ein wenig Allgemeinwissen über C verbreiten. C ist eine Modulsprache, d.h. daß Programme aus einzelnen Modulen bestehen. Dabei unterscheidet man grundsätzlich zwei verschiedene Modultypen. Das wichtigere Modul ist sicher das 'main()'-Modul. Auch als Hauptmodul bezeichnet. Ein C-Programm beginnt immer mit dem ersten Befehl dieses Moduls. Andere Module, meistens Unterprogramme, können dann vom Hauptmodul angesprochen werden. Den Umfang eines Moduls wird immer von einem Klammerspaar bestimmt. Ein typischer Modulaufbau sieht folgendermaßen aus: Modulname() {...}

Ein weiteres Merkmal eines C Programmes ist der sogenannte Programmkopf, dieser existiert aber nicht immer. Der Programmkopf dient dazu Dateien einzubinden, Variablen zu definieren oder allgemeine Zuweisungen zu machen.

Doch beginnen wir mit der Programmierung von Requestern. Um ein Requester zu erzeugen, müssen folgende Schritte beachtet werden: 1. es muß eine Requester Struktur erstellt werden. Strukturen sind Sammlungen von verschiedenen Variablen, die zusammen eine ergeben. Das Aussehen und die Aufgaben eines Requesters werden von einer solchen Struktur bestimmt. Die Requester Struktur ist nebenstehend abgedruckt. Die Programmierung eines Requesters habe ich in Listing 1 nachvollzogen. Eine ausführliche Erklärung von Listing 1 habe ich am Ende des Artikels abgedruckt.

Die Programmierung eines Autorequesters ist sehr einfach und kurz, dementsprechend sind auch die Möglichkeiten begrenzt. Autorequester werden immer in der linken, oberen Ecke des Bildschirms ausgegeben. Außerdem erlauben sie nur das Erstellen von zwei Boolean Gadgets. Listing 2 ruft ein Autorequester auf, Sie werden sehen wie einfach die Erstellung eines solchen Requesters gegenüber eines herkömmlichen ist. Ebenfalls

am Ende des Artikels wird auch dieses Listing noch näher erläutert.

Bevor ich auf die Erklärungen der Listings eingehen möchte, erkläre ich noch verschiedene Systemfunktionen, die sich mit Requestern befassen.

Requester Funktionen

Um ein Requester aufzurufen, also in einem Fenster sichtbar zu machen, muß die Funktion mit dem Syntax

```
Request(Requester,Window)
```

herangezogen werden. Die Funktion holt das Requester aus dem 'Untergrund' hervor. 'Requester' ist hierbei ein Zeiger auf die jeweilige Requester Struktur, 'Window' ein Zeiger auf das zugrundeliegende Fenster. Ein solches Requester verschwindet wieder von der Anzeige wenn die Funktion mit dem Syntax

```
EndRequest(Requester,Window)
```

aufgerufen wird. Die Variablen innerhalb der Klammern besitzen die selbe Bedeutung wie bei der obigen Funktion beschrieben.

Eine weitere Funktion ruft das sogenannte Double Menue Requester auf. Der Syntax dazu lautet:

```
SetDMRequest(Window,Requester)
```

Um ein solches wieder zu schließen wird die Funktion mit dem Syntax

```
ClearDMRequest(Window,Requester)
```

benötigt. Die Variablen 'Window' sind hierbei wieder Zeiger auf das jeweilige Fenster, 'Requester' auf die Requester Struktur.

Ein Autorequester wird auch als einfaches 'Ja-Nein Requester' bezeichnet. Der Syntax zum Aufrufen des Requesters lautet:

```
AutoRequest(Window,Bodytext,Positivetext,Negativetext, Positiveflags, Negativeflags,Width,Height)
```

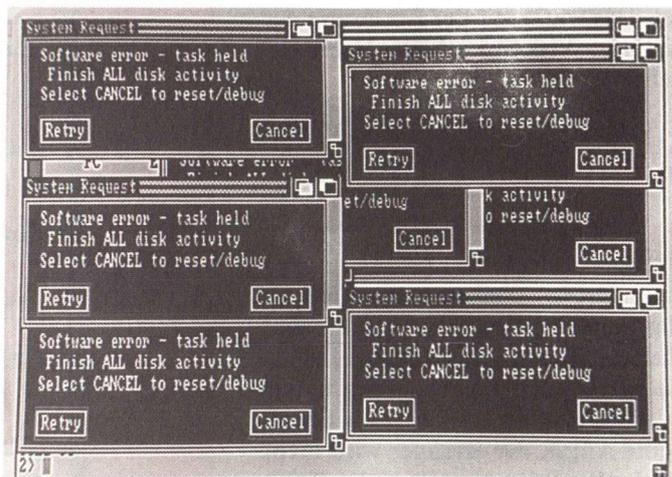
Die nötigen Festlegungen werden

bei diesem Requester gleich in der Funktion bestimmt. Dabei haben die Variablen folgende Bedeutung: 'Window' ist ein Zeiger auf eine Window Struktur, 'Bodytext', 'Positivetext' und 'Negativetext' beinhalten Zeiger auf IntuiText Strukturen (Strukturen die die Ausgabeart des Textes bestimmen), 'Positiveflags' und 'Negativeflags' stehen für IDCMPort. 'Width' und 'Height' geben die Dimensionen des Requesters an. Angaben zur linken, oberen Anfangskoordinate müssen nicht vorgenommen werden, da, wie schon erwähnt, ein Autorequester immer in der linken, oberen Ecke ausgegeben wird. Diese Funktion liefert entweder einen TRUE oder FALSE Wert zurück.

Programmerklärung

Beginnen wir mit Listing 1. Es stellt ein Requester mit zwei Gadgets und einem Informationstext. Wie es fast immer in den Listings des Intuitionsurses der Fall war, werden auch hier zwei Dateien eingebunden. Zum einen die Datei 'types.h', die Variablen-Definitionen enthält, zum anderen die Datei 'intuition.h'. Diese besitzt schon einen höheren Informationsanteil, hier sind alle Intuitionsstrukturen abgelegt und die Definition der verschiedenen Flags. Selbstverständlich enthält sie auch die Struktur des Requesters. In den Zeilennummern 17-19 werden einzelne Strukturen mit einem Zeiger versehen. Die Zeiger werden im Hauptprogramm benötigt. Zeile 23 definiert

die Library Nummer (29), es können auch höhere Nummern eingesetzt werden. Als nächstes folgt die NewWindow Struktur. Das Fenster, welches im Programm verwendet wird, bekommt in der Struktur seine Aufgabe und oder Aussehen verliehen. Nach 'SHORT paare1[]' werden die Koordinaten für den zu zeichnenden Rand festgelegt. Diese Aufgabe übernimmt die Border Struktur, die in Zeile 46 beginnt. Weiter geht es wieder mit Koordinaten, diesmal soll ein Rand um ein Gadget gezeichnet werden. Die nachfolgende Border Struktur übernimmt diese Aufgabe. Das gleiche Spiel beginnt noch einmal ab Zeilennummer 80, das zweite Gadget bekommt seinen Rand. Die nächsten Programmschritte sind die IntuiText Strukturen, die für den Text in den Gadget verantwortlich sind. Die Definitionen in Zeile 102 und 118 sind nicht weiter von Bedeutung, sie dienen nur der besseren Orientierung. Doch es folgen zunächst die beiden Gadget Strukturen, die ab der Zeile 122 beginnen. Nach Beendigung der Gadget Strukturen beginnen in Zeile 134 noch vier weitere IntuiText Strukturen, die für das Requester bestimmt sind. Jetzt beginnt die Requester Struktur, die, wie Sie wissen, für den globalen Requesterapparat verantwortlich ist. Mit der Requester Struktur ist der Programmkopf beendet. Es beginnt das Hauptprogramm ab Zeile 168. Zwei weitere Variablen werden als ULONG bzw. USHORT Wert deklariert, sie werden später zur Ereignisabfrage benötigt. Den selben Verwendungszweck erfüllt der Zeiger in Zeile 178. Der nächste Programmschritt erfolgt in



Listing 1 bewirkt das abgebildete Requester.

Zeile 182. Hier beginnt das Öffnen der Intuition Library. Bevor diese Library nicht geöffnet ist geht überhaupt nichts. Um Funktionen von Intuition zu nutzen muß sie immer erst geöffnet werden, das Selbe gilt selbstverständlich auch für andere Libraries. In Zeile 188 beginnt das Öffnen des Fensters, dazu sollte man wissen, wenn das Öffnen des Fensters sowohl der Library nicht erfolgreich war, daß das Programm seinen Ablauf unterbricht und aussteigt. In Zeile 196 wird das eigentliche Requester erst geöffnet. Nach Öffnen verzweigt das Programm in eine Endlosschleife, die immer wieder bestimmte Ereignisse abfragt. Registriert das Programm, daß ein Gadget angewählt wurde, wird das Unterprogramm angesprungen, das in Zeile 223 beginnt. Hier wird selektiert und herausgefunden welches Gadget angewählt wurde und die entsprechenden Aktionen werden vorgenommen, entweder wird das Programm beendet oder es wird ein Text ausgegeben. Damit ist das Programm auch schon beendet.

Gehen wir jetzt auf Listing 2 über. Auch hier werden wieder die gleichen Dateien eingebunden, die Library Version festgelegt. Zeile 22 und 23 rüsten zwei Strukturen mit Zeigern aus. In Zeile 30 beginnt die IntuiText Struktur für das Autorequester, die beiden folgenden IntuiText Strukturen sind für den Text der zwei Boolean Gadget verantwortlich. In Zeile 51 beginnt schon das Hauptprogramm. Wie immer wird zuerst die Intuition Library geöffnet. Danach erfolgt schon der Aufruf des Autorequesters. Zeile 68 enthält lediglich noch das Schließen der Intuitionlibrary und das Programm ist beendet. Das war's mit der Erklärung der Listings. Viel Spaß beim Nachvollziehen.

Wie geht's weiter?

Ein großer Teil von Intuition ist abgehandelt, aber noch längst nicht alles. Im nächsten Kurs bespreche ich die gefürchteten Alerts, oder besser bekannt als Guru Meditation Error. Keine Angst, der Rechner wird nicht bei der Programmierung in das Reich

der Gurus einkehren, Alerts können auch harmlos sein, zumindest bildet der im nächsten Kurs keine Gefahr.

(AK)

Nachtrag

In Ausgabe 10 der KICKSTART wurde leider ein AmigaBASIC Listing vergessen. Deshalb reichen wir den Abdruck in dieser Ausgabe nach.

```

1  /*****
2  * Das Aufrufen eines Requesters *
3  * Autor: Andreas Kramer *
4  * Copyright KICKSTART 11.87 *
5  * Erstellt auf dem Lattice C Compiler *
6  *****/
7
8  /* Zwei Dateien werden eingebunden */
9
10 #include <exec/types.h>
11 #include <intuition/intuition.h>
12
13 /* Zuweisung von verschiedenen Strukturen mit einem *
14 * Zeiger. Die zeiger werden später im Hauptprogramm *
15 * benötigt. */
16
17 struct IntuitionBase *IntuitionBase;
18 struct IntuiMessage *Nachricht;
19 struct Window *Window;
20
21 /* Variable Definition */
22
23 #define INTUITION_REV 29
24
25 /* Die NewWindow Struktur */
26
27 struct NewWindow Fenster =
28 {
29     170, 25, 300, 150, 0, 0,
30     MOUSEBUTTONS : GADGETDOWN,
31     0, NULL, NULL, 0, NULL, NULL, 0, 0, 0, 0,
32     WBENCHSCREEN
33 };
34
35 /* Festlegung der Koordinaten für den Rand *
36 * des Requesters */
37
38 SHORT paarel[] =
39 {
40     1, 143, 290, 143,
41     290, 1, 1, 1, 1, 143
42 };
43
44 /* Die Border Struktur erzeugt einen Rahmen */
45
46 struct Border rand1 =
47 {
48     1, 1, 3, 2, JAM1, 5,
49     (SHORT *) paarel, NULL
50 };
51
52 /* Eine weitere Festlegung von Koordinaten *
53 * diesmal für die Gadgets */
54
55 SHORT paare3[] =
56 {
57     -2, -2, -2, 14,
58     94, 14, 94, -2, -2, -2
59 };
60
61 /* Eine weitere Border Struktur */
62
63 struct Border rand3 =
64 {
65     -1, -1, 1, 0, JAM1, 5,
66     (SHORT *) paare3, NULL
67 };
68
69 /* Dritte Koordinaten Bestimmung *
70 * wieder für die Gadgets */
71
72 SHORT paare2[] =
73 {
74     0, 0, 0, 12,
75     92, 12, 92, 0, 0, 0
76 };
77
78 /* Dritte Border Struktur */
79
80 struct Border rand2 =
81 {
82     -1, -1, 3, 2, JAM1, 5,
83     (SHORT *) paare2, &rand3
84 };
85
86 /*****
87 * Die IntuiText Strukturen für die zwei Gadgets *
88 *****/
89
90 struct IntuiText text3 =
91 {
92     1, 0, JAM1, 20, 2, 0, " NEIN ", 0
93 };
94
95 struct IntuiText text2 =
96 {
97     1, 0, JAM1, 30, 2, 0, " JA ", 0

```

```

98  };
99
100 /* Variablen Definition fur die Gadget Struktur */
101
102 #define neingadget 2
103
104 /* Es werden zwei Gadgets im Requester erzeugt *
105 * hierzu die beiden Gadget Strukturen */
106
107 struct Gadget gadget2 =
108 {
109     NULL, 160, 119, 90, 11,
110     GADGHCOMP, RELVERIFY : GADGIMMEDIATE, /* Nein Gadget */
111     BOOLGADGET : REQGADGET,
112     (APTR)&rand2, NULL, &text3,
113     0,0,neingadget,0
114 };
115
116 /* Variablen Definition */
117
118 #define jagadget 1
119
120 /* Zweite Gadget Struktur */
121
122 struct Gadget gadget1 =
123 {
124     &gadget2, 30, 119, 90, 11,
125     GADGHCOMP, /* Ja Gadget */
126     GADGIMMEDIATE : RELVERIFY,
127     BOOLGADGET : REQGADGET,
128     (APTR)&rand2, NULL, &text2,
129     0,0,jagadget,0
130 };
131
132 /* IntuiText Strukturen fur das Requester */
133
134 struct IntuiText TextR3 =
135 {
136     1,0,JAM1,15,10,0, "**** Ein KICKSTART Requester ****",0
137 };
138
139 struct IntuiText TextR2 =
140 {
141     1,0,JAM1,15,70,0, " das Requester beenden ? ".&TextR3
142 };
143
144 struct IntuiText TextR1 =
145 {
146     1,0,JAM1,15,60,0, " Wollen Sie wirklich ".&TextR2
147 };
148
149 /******
150 * Die Requester Struktur
151 ******/
152
153 struct Requester Requester1 =
154 {
155     NULL,
156     0, 0, 300,150,
157     0, 0,
158     &gadget1,
159     &rand1, &TextR1,
160     0, 2, 0,
161     NULL, NULL
162 };
163
164 /******
165 * Beginn des Hauptprogrammes
166 ******/
167
168 main()
169 {
170     /* Zuweisung von zwei Variablen als ULONG und USHORT *
171     * Wert. Werden spater zur Ereignisabfrage benötigt.*/
172
173     ULONG NachrichtenArt;
174     USHORT code;
175
176     /* Zusätzlicher Zeiger auf eine Struktur */
177
178     struct Message *GetMsg();
179
180     /* Offnen der Intuition Library */
181
182     IntuitionBase=(struct IntuitionBase *)
183     OpenLibrary("intuition.library", INTUITION_REV);
184     if(IntuitionBase==NULL)exit(FALSE);
185
186     /* Offnen des Fensters */
187
188     if((Window=(struct Window *)
189     OpenWindow(&Fenster))!=NULL)
190     exit(FALSE);
191
192     /******
193     * Der Aufruf des Requesters
194     ******/
195
196     Request(&Requester1,Window);
197
198     /* Die Programmschleife fragt bestimmte Ereignisse ab */
199
200     for(;;)
201     {
202         if(Nachricht=(struct IntuiMessage *)
203         GetMsg(Window->UserPort))
204         {
205             NachrichtenArt=Nachricht ->Class;
206             code=Nachricht->Code;
207             ReplyMsg(Nachricht);
208             switch (NachrichtenArt)
209             {
210                 case GADGETDOWN : Re_aktion(Nachricht, Window);
211                                 break;
212             } /* switch */
213         } /* if */
214     } /* for */
215     return(0);
216 } /* main */
217
218 /******
219 * Unterprogramm zum Gadget abfragen und zum beenden des *
220 * Requesters
221 ******/
222
223 Re_aktion(Meldung, Fenster1)
224
225 struct Window *Fenster1;
226 {
227     struct Gadget *JNGadget;
228     int gad;
229
230     JNGadget=(struct Gadget *) Nachricht->IAddress;
231     gad=JNGadget->GadgetID;
232
233     switch(gad)
234     {
235         case neingadget : printf("'' NEIN ''");
236                         break;
237
238         case jagadget : EndRequest(&Requester1,Window);
239                         CloseWindow(Window);
240                         CloseLibrary(IntuitionBase);
241                         printf("Requester geschlossen\n");
242                         exit(TRUE);
243                         break;
244     }
245     return (0);
246 } /* Programmende */
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

65
66 /* Schließen der Intuition Library */
67
68 CloseLibrary(IntuitionBase);
69 return(0);
70 }

1 REM Beispielprogramm ueber Menueprogrammierung
2
3 REM Erstellung der Menues
4 MENU 1,0,1."Menue 1"
5 MENU 1,1,1."Punkt 1"
6 MENU 1,2,1."Punkt 2"
7 MENU 1,3,1."Quit "
8
9 REM Ausschalten der BASIC Menues
10 MENU 2,0,0.""
11 MENU 3,0,0.""
12 MENU 4,0,0.""
13
14 REM Abfrage ob ein Menu angewaehlt wurde
15 ON MENU GOSUB Menueauswahl
16 MENU ON
17
18 REM Endlosschleife
19 WHILE -1
20 SLEEP
21 WEND
22
23 REM Menuepunkt Selektierung
24 Menueauswahl:
25 m=MENU(0)
26 t=MENU(1)
27
28 REM Sprung in das jeweilige Subprogramm
29 IF t=1 THEN GOSUB Menuepunkt1
30 IF t=2 THEN GOSUB Menuepunkt2
31 IF t=3 THEN GOTO Menuepunkt3
32 RETURN
33
34 REM Die verschiedenen Subprogramme
35 Menuepunkt1:
36 PRINT "Menuepunkt 1 wurde angewaehlt"
37 RETURN
38 Menuepunkt2:
39 PRINT "Menuepunkt 2 wurde angewaehlt"
40 RETURN
41 Menuepunkt3:
42 MENU RESET
43
44
45
46

```

Tabellarische Übersicht

Die Variablen der Requester Struktur haben folgende Bedeutung.
Die Erklärung der einzelnen Variablen:

OlderRequester	Diese Variable zeigt auf ein Requester, welches vor diesem ausgegeben wurde.
LeftEdge, TopEdge	Legen Sie die Variablen fest, wenn das Requester relativ zu der oberen, linken Ecke eines Fensters ausgegeben werden soll.
Width, Height	Die Variablen beschreiben die Größe des rechteckigen Requesters.
RelLeft, RelTop	Initialisieren Sie die Variablen, wenn das Requester relativ zu der momentanen Position des Zeigers ausgegeben werden soll. Hierbei müssen Sie den POINTREL Flag setzen.
ReqGadget	Ein Zeiger auf das erste Gadget in der Liste.
ReqBorder	Ein Zeiger auf eine Border Struktur, welche um und innerhalb des Requesters einen Rand zeichnet.
ReqText	Ein Zeiger auf eine IntuiText Struktur, welche einen Text für das Requester beinhaltet.
Flags	Es können die untenstehenden Flags gesetzt werden.
POINTREL	bewirkt, daß das Requester relativ zum Zeiger ausgegeben wird.
PREDRAWN	wird gesetzt, wenn eine BitMap Struktur oder ImageBMap Struktur für das Requester bestimmt ist.
Intuition benutzt die nachstehenden Flags:	
REQOFFWINDOW	Veranlaßt Intuition das Requester sofort zu aktivieren.
REQACTIVE	Dieses Requester befindet sich immer im aktiven Zustand.
SYSREQUEST	Diese Flag wird gesetzt, wenn es sich um ein System Requester handelt.
BackFill	Farbnummer für das Ausfüllen des Requesters.
ReqCRect, ReqBMap	Diese Variablen werden gebraucht, falls ein Bild (Image) für das Requester bestimmt ist.
ImageBMap	Zeiger auf ein Bit-Map für das Requester.

★ ACHTUNG ★ ACHTUNG ★ ACHTUNG ★ ACHTUNG ★

An alle Amiga Programmierer

Wir suchen **Amiga Programme** aus allen Anwendungsgebieten (Spiele, Grafik, Sound usw.), für die Vermarktung im gesamten Bundesgebiet.

Wir übernehmen die Kosten für den Vertrieb (Werbung, Verpackung, Versand usw.).

WIR BIETEN IHNEN EINE UMSATZBETEILIGUNG BIS ZU 50 %!

Wenn Sie Interesse haben, wenden Sie sich noch heute an uns:

A. Wardenga **Software 2000**
Lange Str. 51 · 2320 Plön/Holstein

AMIGA - Software	Public Domain Disketten	1 bis 92 1 bis 11 1 bis 39 1 bis 40	Inhalt: Spiele, Bilder, Video-Clips, Musik Anwendungen, Programmiersprachen Textverarbeitung, DFO und Demos der bekanntesten TOP-Programme
		Fish Disk Auge 4000 Faug Panorama	
		Jede PD-Diskette 6,50 DM	
		Aktuelle Spiele auf Anfrage	
		A. Fischer - 05257 / 4347	
		A 794 HÖVELHOF	

usw. usw. - Fordern Sie mit Freiumschlag unsere Liste an! Im Computer-Center oder bei uns zu obigen, unverbindlich empfohlenen Preisen + DM 3,- bei Vorkasse oder DM 4,70 bei Nachnahme

ASTROLOGISCHES KOSMOGRAMM

Nach Eingabe von Namen, Geburtsort (geografischer Lage) und Geburtszeit werden errechnet: Sternzeit, Ascendent, Medium Coeli, Gestirne im Tierkreis, Häuser nach Dr. Koch/Schäck (Horoskop-Daten mit Ephemeriden. Außer dem Bildschirmdisplay kann Ausdruck auf 2 DIN A4-Seiten erfolgen: davon 1/2 Seite allgemeines Persönlichkeitsbild mit Partnerschaftskriterien und 1/2 Seite Tierkreisdiagramm (Horoskop). Alle Planeten mit Sonne und Mond. Für alle Berufs- und Hobby-Astrologen eine unentbehrliche Arbeitserleichterung. **75,-**

BIOKURVEN

Zur Trendbestimmung der Bio-rhythmen und des seelisch-/geistig-/körperlichen Gleichgewichts mit Druck des Kurvendiagramms von oben nach unten in beliebiger Länge.

In der rechten Blathälfte das Diagramm, links eine Auswertung des Gesamtpotentials für jeden Tag. Werte für bestimmte Tage auch auf dem Bildschirm. Ausführliche Beschreibung der wissenschaftlichen Grundlagen.

Ideal für Partnervergleiche.

58,-

Prg. für alle AMIGA-Modelle

— Exzellent in Struktur, Grafik, Sound — alle Prg. in Deutsch —

GELD

Man wählt mit der Maus unter 25 Rechenroutinen in den Bereichen: Anlage — Kapital — Vermögensbildung — Rentensparen — Rendite — Lasten — Zinsen/Zinseszinsen — Kredit — Hypotheken — Laufzeit — Amortisation — Ratenzahlung — Wertverlust — Nominal und Effektivzinsen — Ausdruck vollständiger Tilgungsraten — Diskontierung — Devisen/Sorten — Konvertierung **98,-**

KALORIEN-POLIZEI — Nach Eingabe von Größe, Gewicht, Geschlecht, Arbeitsleistung erfolgt Bedarfsrechnung und Vergleich m. d. tatsächlichen Ernährung (Fett, Eiweiß, Kohlehydrate), Idealgewicht, Vitalstoff, auf Wunsch Ausdruck **58,-**
ADRESSEN **66,-**
BIBLIOTHEK **86,-**
LAGERARTIKEL **86,-**



I. Dinkler

Am Schneiderhaus 17 · D-5760 Arnsberg 1
Tel. 0 29 32/3 29 47

PHILGERMA IHR SPEZIALIST FÜR AMIGA COMPUTERSPRACHEN

True BASIC (True Basic) — Modernes strukturiertes Basic m. Grafik. Handbuch 500 S. DM 398,00

AC/BASIC Compiler (absoft) — Dieser Compiler paßt zu dem vorhandenen Amiga Basic Interpreter und ist bis 50 mal schneller DM 398,00

AZTEC C68k/am-p Professional (Manx) — Neueste Version 3.4 dieses bekannten C-Compilers. Er umfaßt optimierenden C-Compiler, Assembler, Linker, Bibliotheken und Beispiele. Unterstützung des 68020 und des 68881 Prozessors. Hervorragendes engl. Handbuch 400 S. DM 448,00

AZTEC C68k/am-d Developer (Manx) — Zusätzlich Debugger, Make, Diff, Grep usw. DM 648,00

AZTEC C68k/am-c Commercial (Manx) — Zusätzlich Z(vi)Editor, Quellcode Bibliotheken ... DM 1098,00

AC/FORTRAN77 (absoft) — ANSI X3.9-78 Standard Fortran 77 Compiler für 68000 Prozessoren. Zusätzliche Optionen. Fließkommaarithmetik 16 Stellen nach IEEE Standard. Overlays, virtuelle Arrays und Debugger. Handbuch 300 S. DM 598,00

AC/FORTRAN77-68020/68881 (absoft) DM 1198,00

TEXTVERARBEITUNG MIT DEM AMIGA

WORDPerfect Textverarbeitung engl. ... DM 790,00
Wizawrite Textverarbeitung. DM 198,00
Textomat Textverarbeitung. DM 98,00
Page Setter Desktopprogramm. DM 378,00
Publisher 1000 Desktopprogramm. DM 498,00
Instant Music Kompositionsprogramm. DM 79,00
Sonix 1.4 Musikprogramm. DM 198,00

NATÜRLICH HABEN WIR AUCH SPIELE

Barbarian Abenteuerpiel DM 69,00; The Guild of Thieves DM 69,00; Karate Kid II DM 69,00; Gold Runner Geschicklichkeitsspiel DM 69,00; Faery Tale DM 119,00; Uninvited DM 79,00; Defender of the Crown DM 89,00; Sinbad Abenteuerpiel DM 89,00; Deja Vu Grafisches Krimispiel DM 89,00; Marble Madness Geschicklichkeitsspiel DM 69,00; Starglider Geschicklichkeitsspiel DM 69,00; Pawn Text + Grafik-adventure DM 69,00; Terrorpods Abent. DM 69,00; Archon Geschicklichk. DM 69,00; Archon II DM 69,00; Chessmaster 2000 DM 99,00; Bureaucrazy Infoc. Textadventure DM 89,00; Emerald Mine DM 29,00; Portal SF DM 99,00; Bard's Tale DM 119,00; Flight II Sublogic **DM 99,00**

MODULA II Standard (TDI) — Diese umfangreiche Modula Implementierung vereint die Vorteile von Pascal mit maschinennahen Sprachelementen. Compiler mit AmigaDOS Einbindung DM 248,00

MODULA II Developer (TDI) — Zusätzlich symbolischer File Decoder, Cross Referencer, Modula CLI, Utilities für IFF und ILBN DM 398,00

MODULA II Commercial (TDI) — Zusätzlich alle Modula Module im Quellcode DM 648,00

K-SEKA Assembler (KUMA) DM 168,00

GRABBIT Screen Dump Programm DM 68,00

Superbase Dateiverwaltung deutsch DM 248,00

dBMAN (Versasoft) — Datenbank DM 398,00

Logistix Tabellenkalk. deutsch DM 398,00

Deluxe Paint II Grafikprogramm DM 278,00

Deluxe Video Construction neu V1.2 DM 278,00

Sculpt 3D Animationsprogramm DM 198,00

VideoScape 3D Animationsprogramm DM 398,00

NEWIO Leiterplattenentflechtungspr. DM 498,00

Aegis Draw plus CAD Programm DM 578,00

AUSZUG AUS UNSERER HARDWARE-LISTE

Einzelaufwerk 3'5 720K mit Bus DM 398,00
Doppelaufwerk 3'5 2 * 720K DM 798,00
Einzelaufwerk 5 1/2 40/80 Spuren DM 548,00
Speichererweiterung extern 2MB-RAM DM 998,00
Harddisk 20 MB Amiga 500/1000 DM 1298,00
Harddisk 20 MB Amiga 500/1000 schnell DM 1698,00
Harddisk 40 MB Amiga 500/1000 schnell DM 2598,00
Digitalisier-System DIGI-VIEW V2.0 DM 398,00
10 Disketten 3'5 2DD 1a Qualität DM 29,00

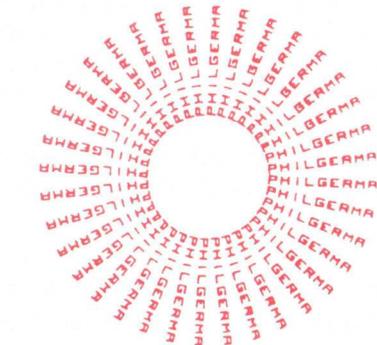
Fordern Sie unsere umfangreiche **Preisliste** an. Händler bitte Händlerliste anfordern. Bestellungen bitte an:

PHILGERMA GmbH, Ungererstraße 42, 8000 München 40, Tel: 0 89 / 39 55 51

Bei Bestellungen unter DM 200 beträgt der Versandkostenanteil DM 4,80. Nachnahme DM 3,20. Lieferung ins Ausland nur gegen Vorkasse (Überweisung o. Eurocheck) + DM 20 Versandkosten.

Besuchen Sie unseren Softwareladen in der Ungererstraße 19.

Sie können alle Produkte anschauen und testen.



MCC PASCAL (Metacomco) — Pascal Compiler ISO 7185 Standard. Single Pass Compiler, schnell u. effizient. Die AmigaDOS Routinen können voll im Pascal eingebunden werden. MCC Pascal Prog. können mit MCC Assembler oder Lattice C geflinkt werden. Handbuch 200 Seiten DM 198,00

MCC ASSEMBLER (Metacomco) — Professioneller Makro Assembler, der den vollen Motorola 68000 Instruction Set unterstützt. Mit Editor, Linker und AmigaDOS-Routinen DM 168,00

LATTICE C (Lattice) — Bewährter C-Compiler der USA-Firma Lattice, Standardprodukt in der IBM-Welt. Kompatibel auf vielen Rechnern, gut für professionelle Entwicklungen. Kerningham/Ritchie Standard. Fließkommaarithmetik mit 16 Stellen Genauigkeit. Die neue Version 3.10 enthält Assembler, Linker und Text Management. Ausführliches engl. Handbuch 300 S. ... DM 448,00

CAMBRIDGE LISP (Metacomco) — Interpreter und Compiler mit dem Sprachumfang, den man von Großrechnern gewöhnt ist. Volle Real-Arithmetik 16 MByte Adressraum. Handbuch 330 S. DM 448,00

TOOLKIT (METACOMCO) — Sammlung von wichtigen Utilities: Pipes, Librarian, Disassembler, Enlarge, Browse und Aux CLI DM 118,00

SHELL (METACOMCO) — Erweiterung des CLI von Metacomco, dem Entwickler des AmigaDos. UNIX ähnliche Kommandos mit Anleitung ... DM 148,00

Die bis jetzt arglosen Amigabesitzer sollten aufhorchen und diesen Artikel mit Aufmerksamkeit lesen. Der erste Computervirus auf dem Amiga ist im Umlauf. Wir wollen Ihnen helfen, diesen Virus zu erkennen und so eventuellen Schaden von Ihren Disketten fernzuhalten.

Grundsätzliches

Was ist ein Virus? Unter einem Virus versteht man ein Programm, das erstens sich selbst verbreitet (vergleichbar mit einem 'echten' biologischen Virus) und zweitens zum Ausbruch einer Krankheit führt. Das diese Krankheit nicht Grippe oder etwas ähnliches ist, sondern Datenverlust und Systemabsturz sind, müßte jeder wissen. Der hier beschriebene Virus ist nur eine Form der vielen Möglichkeiten, ein solches Programm zu schreiben. Es ist jedoch das erste Programm dieser Art, das bis jetzt aufgetreten ist. Die beschriebenen Maßnahmen gelten daher auch nur in beschränktem Maß für eventuell vorhandene andersartige Viren. Sie geben dem Programmierer jedoch einen Einblick in die Arbeitsweise solcher Programme und erlauben es somit, entsprechend geänderte Prüf-routinen für andere Viren, als den hier vorgestellten, zu entwerfen.

Funktionsweise

Der hier besprochene Virus ist in der Bootspur der Diskette abgelegt. Die Bootspur enthält bei einer gesunden Diskette Informationen, die dem Starten des Amiga Dos dienen. Ist eine Diskette infiziert, so enthält sie außerdem das Virusprogramm an dieser Stelle der Diskette. Dies bedeutet für den Systemstart mit einer solchen Diskette zweierlei. Erstens wird, nachdem der Rechner zum Einlegen der Workbench auffordert, der Virus geladen und automatisch!! gestartet. Zweitens lädt der, nun fest im System installierte Virus, auf normale Weise das Amiga Dos nach. Diese Vorgänge spielen sich so schnell ab, das der Anwender von alledem nichts merkt.

Die Vorgehensweise des Virusprogramms ist dabei folgende:

1. Der Virus reserviert sich einen kleinen Speicherbereich von 1024 Bytes ab der Adresse 7ec00 für sich selbst.

Hilfe!!!

die



VIREN

kommen

2. Der Virus kopiert sich selbst in diesen Speicherbereich.
3. Der Virus setzt einen, vom System unberührten Vektor auf den Programmumfang im reservierten Ram.
4. Das Amiga Dos wird nun ganz normal gestartet.

Bis zum jetzigen Zeitpunkt ist eigentlich noch nichts 'Schlimmes' passiert. Die Diskette mit der Sie gestartet haben, ist 'verseucht'. Das Hauptziel eines Virusprogramms jedoch ist die eigene Vermehrung. Hier liegt auch schon die Tücke des hier behandelten Virus. Denn so unberührt ist der unter Punkt 3. beschriebene Vektor nur in einer Weise. Er wird vom Betriebssystem nach einem Rest bzw. nach dem Einschalten des Rechners zwar nicht belegt, jedoch abgefragt und bei einem von Null verschiedenen Wert auch angesprungen. Direkt nach dem Einschalten ist dies für uns nicht wichtig, da der komplette Speicher getestet und gelöscht wird (inklusive dieses Vektors).

Betrachten wir jedoch die Vorgänge nach einem Reset (Amiga Amiga CTRL). In diesem Fall überprüft eine kleine Routine den Zustand des Systems mit Hilfe mehrerer Prüfverfahren (Prüfsummen und Adressplausibilitätskontrollen) und startet bei einem positiven Abschluß das System sofort, d.h. ohne erneutes Initialisieren des Speichers und beim A 1000, dem Laden der Kickstartsoftware von Diskette.

In diesem Verhalten des Systems liegt nun beim Befehl, durch den hier besprochenen Virus der Grund, da der oben beschriebene Vektor bei dem nun weggefallenen Systemtest nicht mehr auf null gesetzt wird.

Jetzt wirds ernst

Das System kommt nun an die Stelle, an welcher der oben beschriebene Vektor überprüft wird. Da die Virusroutine hier ihren Startwert abgelegt hat, verzweigt das System zum Virus. Der Anwender muß nun mit großen Unannehmlichkeiten rechnen. Während auf dem Bildschirm zum Einlegen der Workbench durch das altbekannte Bild aufgefordert wird, lauert im Rechner der Virus, um sich nach Einlegen einer nicht schreibgeschützten Diskette in deren Bootsektor zu kopieren. An dieser Stelle schließt sich der Teufelskreis und eine weitere Diskette ist infiziert.

Erste Folgen

Ungeachtet dessen, zu dem der Virus fähig ist, werden allein durch den Kopiervorgang Disketten zerstört, die wichtige Daten auf der Bootspur abgelegt haben, wie z.B. einige teure Originaldisketten, wie das Spiel Barbarian.

Erkennen

Da der hier vorliegende Virustyp in der Bootspur einer Diskette abgelegt ist, kann man ihn an dieser Stelle einfach aufspüren. Einen Diskettenmonitor sollten Sie jedoch besitzen. Haben Sie keinen, wir haben einen für Sie auf der Public Domain Diskette Nr. 7. Stellen sie bei Ihrem Diskettenmonitor folgende Werte ein:

Spur (Cylinder/Track)	0
Seite (Head)	0
Sektor (Sector)	1

Wenn Sie nun das Anzeigenformat auf ASCII einstellen (Buchstaben-darstellung), können Sie auf einer verseuchten Diskette den Text aus Abbildung 1 wiederfinden. Bitte überprüfen Sie auf diese Weise alle Ihre Disketten auf das Vorhandensein des Virus.

Behandeln

Finden Sie bei sich eine oder mehrere erkrankte Disketten, so können Sie diese noch retten. Nehmen Sie eine unbefallene Workbench und schalten Sie den Rechner aus. Warten Sie ca. 20 Sekunden um sicher zu gehen, daß der Speicher Ihres Rechners seinen Inhalt verliert. Schalten Sie den Rechner wieder ein und starten Sie das System mit dieser Workbench. Nun können Sie aus dem CLI mit Hilfe des 'Install' Befehls die Bootspur auf der versuchten Diskette überschreiben und so den Virus vernichten. Vorgehensweise:

1. CLI anklicken
2. copy c:install to ram: eingeben
3. cd ram: eingeben
4. Die kranke Diskette in dfO: einlegen
5. install dfO: eingeben
6. Mit dem Diskettenmonitor den Erfolg überprüfen.

Waren Ihre Disketten von dem Virus befallen und Sie haben sie nicht ohne Aufwand 'geheilt', sind sie bestimmt an einer vorbeugenden Maß-

nahme in bezug auf resetfeste Viren interessiert.

Vorbeugen

Da der Virus über einen Vektor aktiv wird und dieser dazu die Startadresse des Virusprogramms enthalten muß, kann ein kleines Programm z.B. diesen Vektor abfragen und gegebenenfalls Alarm schlagen.

Im System existieren drei solcher Vektoren. Sie haben die Namen ColdCapture, CoolCapture und WarmCapture. Listing 1 zeigt ein kleines Maschinensprachprogramm, das diese drei Vektoren in die Register d0, d1 und d2 lädt. Anschließend werden diese Vektoren durch ein logisches Oder im Register d0 verknüpft und dieser Wert zurückgegeben. Rufen Sie dieses Programm mit 'run checkvirus' auf, gibt es einen Zahlenwert aus, wenn einer der drei Vektoren belegt ist. Sind alle drei Vektoren null, wird kein Wert ausgegeben. Kopieren Sie das Programm Checkvirus in das c: Verzeichnis Ihrer Workbench und fügen sie am Anfang der 'startup-sequence' folgende zwei Zeilen ein:

run checkvirus wait 5

Sie erhalten nun nach jedem Systemstart Meldung über einen eventuell gesetzten Vektor und somit

auch über einen vorhandenen Virus. Das Programm Checkvirus ist mit dem Seka Assembler erstellt und kann auch von dort aus (writeobjekt) auf Diskette geschrieben werden. Haben Sie keinen Seka Assembler, reicht auch der abgedruckte Basic-lader aus.

Abschließendes

Arbeiten Sie nur mit original Software, sind Sie relativ sicher vor solchen Killerprogrammen, die auch nicht davor zurückschrecken, einmal schnell die Festplatte zu löschen. Die größte Verbreitung findet ein Virus wohl über den Tausch von raubkopierter Software. Gerade Spiele sind ein geradezu idealer Nährboden für ein solches Programm. Wie oft werden bei Freunden die Spiele nur kurz gestartet und dann kopiert. Doch gerade dieses 'nur anstarten', meist durch einen Reset, reicht dem hier vorgestellten Virusprogramm bereits, um sich auszubreiten. Wir bekamen den Hinweis auf die Existenz eines solchen Virus von einem Leser, bei welchem sich der Virus bereits auf die bevorzugte Workbench kopiert hatte.

Ich möchte mich auch im Namen aller Leser bei diesem Herrn noch einmal bedanken und wünsche Ihnen viel Erfolg bei der Jagd auf den Virus.

(GC)

```

0300: 01800800 8605FFFE 01800D51 8A05FFFE
0310: 01800000 FFFFFFFF 002FFFFA FFCC6772
0320: 61706869 63732E6C 69627261 72790064
0330: 6F732E6C 69627261 72790000 20536F6D
0340: 65746869 6E672077 6F6E6465 7266756C
0350: 20686173 20686170 70656E65 64D2AA2D
0360: 17596F75 7220414D 49474120 69732061
0370: 6C697665 20212121 BEA04113 616E642C
0380: 20657665 6E206265 74746572 2E2E2E50
0390: 50041F53 6F6D6520 6F662079 6F757220
03A0: 6469736B 73206172 6520696E 66656374
03B0: 65646E32 5A0E6279 20612056 49525553
03C0: 20212121 8C783216 416E6F74 68657220
03D0: 6D617374 65727069 65636520 6F668232
03E0: 32165468 65204D65 67612D4D 69676874
03F0: 79205343 41202121 DC6E0000 4EF90000
0400: 00004121 53434121 53434121 53434121
0410: 53434121 53434121 53434121 53434121

```

```

.....Q....
...../....gr
aphics.library.d
os.library.. Som
ething wonderful
has happened..-
.Your AMIGA is a
live !!!..A.and,
even better...P
P..Some of your
disks are infect
edn2Z.by a VIRUS
!!!.x2.Another
masterpiece of.2
2.The Mega-Might
y SCA !!.n..N...
..A! SCA! SCA! SCA!
SCA! SCA! SCA! SCA!

```

```

start:      move.l      $00000004,a6      ; ExecBase nach a6
           move.l      $2a(a6),d0      ; ColdCapture nach d0
           move.l      $2e(a6),d1      ; CoolCapture nach d1
           move.l      $32(a6),d2      ; WarmCapture nach d2
           or.l        d1,d0
           or.l        d2,d0          ; Werte odern
           rts                          ; und an run ubergeben

```

```

OPEN "Checkvirus" FOR OUTPUT AS #1
FOR n=1 TO 88
  READ i
  j=j+i
  PRINT #1,CHR$(i);
NEXT n
IF j<>3375 THEN PRINT "Fehler in Datas"
CLOSE #1
END

```

```

DATA &h00,&h00,&h03,&hf3, &h00,&h00,&h00,&h00
DATA &h00,&h00,&h00,&h02, &h00,&h00,&h00,&h00
DATA &h00,&h00,&h00,&h01, &h00,&h00,&h00,&h07
DATA &h00,&h00,&h00,&h01, &h00,&h00,&h03,&he9
DATA &h00,&h00,&h00,&h07, &h2c,&h79,&h00,&h00
DATA &h00,&h04,&h20,&h2e, &h00,&h2a,&h22,&h2e
DATA &h00,&h2e,&h24,&h2e, &h00,&h32,&h80,&h81
DATA &h80,&h82,&h4e,&h75, &h4a,&hac,&he2,&hb2
DATA &h00,&h00,&h03,&hec, &h00,&h00,&h00,&h00
DATA &h00,&h00,&h03,&hf2, &h00,&h00,&h03,&heb
DATA &h00,&h00,&h00,&h01, &h00,&h00,&h03,&hf2

```

Listing 2:

DAS AMIGA PROJEKT

DER AUDIODIGITIZER DER LUXUSKLASSE

8 Monate Entwicklungszeit stecken in diesem SAMPLER!
Hier einige Features von DSound in Stichworten:

- Erzeugen von SOUNDS im STANDARD-FORMAT
- Erzeugen von SOUNDS im IFF-FORMAT
- Erzeugen von Instrumenten im IFF-FORMAT
- Erzeugen von perkussiven SONIX-INSTRUMENTS (Pauke, Becken etc.)
- Erzeugen von nichtperkussiven SONIX-INSTRUMENTS (Trompete etc.)
- ECHO- und HALLEFFEKTE in STEREO mit fertigen SOUND-SAMPLES, DIGITIZER direkt als ECHO-GERÄT benutzen (ohne zu sampeln!)
- EFFEKT-BOARD mit FM u. AM-Modulationsmöglichkeiten
- PLAYBACK-Geschwindigkeiten während der Wiedergabe ändern.
- SAMPLING direkt auf DISK (anstatt ins RAM, SUPER-LONG-PLAY-DISK)
- Komplett in ASSEMBLER programmiert.



Viele weitere Möglichkeiten!
Skeptiker fordern unsere DEMO-DISK an. (Schutzgebühr DM 10,-)

DE LUXE SOUND V.2.2. PLUS

DSOUND V.2.2. PLUS für AMIGA 1000 komplettes Gerät im Gehäuse mit Anleitung, Steuersoftware und Demo-Sounds
nur 198,- DM

DSOUND V.2.2. PLUS für AMIGA 500 / AMIGA 2000 technische Einzelheiten und Lieferumfang wie bei A 1000. (siehe Abbildung)
nur 228,- DM

DSOUND 2.2 DEMO-DISK mit der Original-Anleitung und DEMO-SOUNDS
nur 10,- DM

MIC 600 passendes Dynamikmikrofon für DSound-Sampler
nur 25,- DM

AK 2 passendes 2m Adapterkabel "DIN-5-STEREO auf CINCH"
nur 7,- DM

AMIGA-LAUFWERKE anschlussfertig an AMIGA 500/1000/2000 mit Metallgehäuse (kunststofflackiert), Busdurchführung und Driveabschalter!

3,5' Einzellauf nur 369,- DM
3,5' Doppellauf nur 698,- DM
5,25' Einzellauf nur 448,- DM
3,5/5,25' Kombi nur 795,- DM

EASY-TITLE Superkurzer Titel-maker! Lädt ILBM LO-RES, MED-RES, HI-RES INTERLACED, PAL-FORMAT, DIGI-VIEW und HAM-Bilder sowie SOUNDFILES im DSOUND/FUTURE SOUND-Format
nur 29,- DM

STRING RELACER sucht und ersetzt ASCII-Strings schnell und komfortabel. Unser neuester GIG!
nur 29,- DM

BOOT-TITEL erzeugt einen Titelvorspann in 4096 FARBEN !!! im Bootsektor mit diversen DPAINT Brushes.
nur 29,- DM

PILBM mit diesem Epson kompatiblen Druckertreiber können Sie z.Bsp. DPAINT-PICS verzerrungsfrei ausdrucken z.Bsp. Layouts in Hires!
nur 29,- DM

CPCLO wandelt PCLO 3.0 Layouts in ILBM Hi-Res-Pics (DPAINT)! Weiterverarbeitung mit DPAINT, Ausdruck ohne Plotter !! Ein nützliches Tool.
nur 29,- DM

Alle obigen Programme in Assembler!

Preise für Rechner, Drucker Zubehör etc. etc. auf Anfrage!!

Hagenauer computer
MÜNSTERSTRASSE 202
D-4700 HAMM 5

(0 23 81) 67 31 65

Die Lieferung erfolgt per Nachnahme zzgl. Versandkosten.

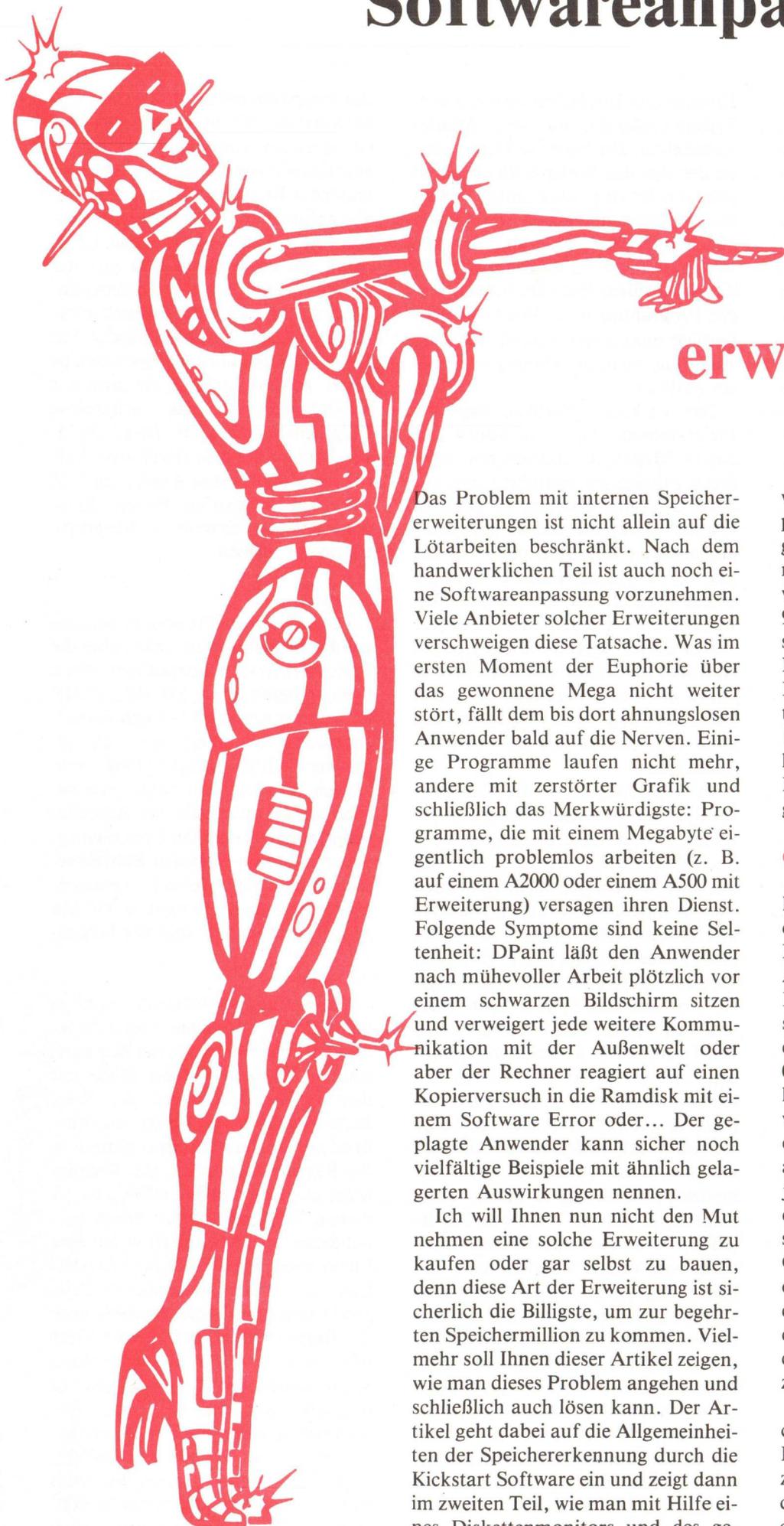
Händleranfragen willkommen

Softwareanpassung von

internen

Mega-

erweiterungen



Das Problem mit internen Speichererweiterungen ist nicht allein auf die Lötarbeiten beschränkt. Nach dem handwerklichen Teil ist auch noch eine Softwareanpassung vorzunehmen. Viele Anbieter solcher Erweiterungen verschweigen diese Tatsache. Was im ersten Moment der Euphorie über das gewonnene Mega nicht weiter stört, fällt dem bis dort ahnungslosen Anwender bald auf die Nerven. Einige Programme laufen nicht mehr, andere mit zerstörter Grafik und schließlich das Merkwürdigste: Programme, die mit einem Megabyte eigentlich problemlos arbeiten (z. B. auf einem A2000 oder einem A500 mit Erweiterung) versagen ihren Dienst. Folgende Symptome sind keine Seltenheit: DPaint läßt den Anwender nach mühevoller Arbeit plötzlich vor einem schwarzen Bildschirm sitzen und verweigert jede weitere Kommunikation mit der Außenwelt oder aber der Rechner reagiert auf einen Kopierversuch in die Ramdisk mit einem Software Error oder... Der geplagte Anwender kann sicher noch vielfältige Beispiele mit ähnlich gelagerten Auswirkungen nennen.

Ich will Ihnen nun nicht den Mut nehmen eine solche Erweiterung zu kaufen oder gar selbst zu bauen, denn diese Art der Erweiterung ist sicherlich die Billigste, um zur begehrten Speichermillion zu kommen. Vielmehr soll Ihnen dieser Artikel zeigen, wie man dieses Problem angehen und schließlich auch lösen kann. Der Artikel geht dabei auf die Allgemeinheiten der Speichererkennung durch die Kickstart Software ein und zeigt dann im zweiten Teil, wie man mit Hilfe eines Diskettenmonitors und des ge-

wonnenen Knowhows selbst eine Anpassung der Kickstartsoftware an die geänderten Speicherverhältnisse vornehmen kann. Ich werde dabei sowohl auf die Erweiterung aus Heft 9/87 eingehen als auch eine Anpassung der 'Konkurrenzerweiterung aus Heft 9/87 der 68000er beschreiben. Diese beiden Fälle sind stellvertretend für die Adressbereiche 080000 – 100000 und 100000 – 180000 anzusehen. Sie können so auch für gekaufte Erweiterungen dieser Art einfach eingesetzt werden.

Grundlagen

Das Betriebssystem des Amiga kennt zwei Arten von Arbeitsspeichern. Diese sind mit Chipmem und Fastmem benannt. Der erste Teil des Arbeitsspeichers ist das Chipmem. Der physikalische Adressbereich dieser Speicherart liegt von null bis einem halben Megabyte (000000 – 080000). Der Name Chipmem ist hierbei jedoch nicht willkürlich gewählt. Beide Arten von Arbeitsspeicher sind ja durch Speicherchips aufgebaut. Die ersten 512 KByte sind jedoch nur von den Spezialbausteinen des Amiga adressierbar. Diese Bausteine sind unter anderem für den Grafikaufbau, Sound und Diskettenoperationen zuständig. Die Tatsache, daß nur dieser Teil des Speichers von den Spezialchips angesprochen werden kann, hat zu der Speicherbezeichnung Chipmem geführt.

Jede weitere Art von Arbeitsspeicher wird als Fastmem bezeichnet. Der Prozessor des Amiga hat als einziger die Möglichkeit auf diesen Speicher zuzugreifen. Der Zugriff auf eine Speicherzelle im Fastmem kann

nicht durch die Spezialchips behindert werden. Dadurch ist dieser Teil des Arbeitsspeichers bei Zugriffen durch den Prozessor schneller als das Chipmem, auf das auch noch die anderen Bausteine zugreifen können. Auf Grund dieses Geschwindigkeitsvorteils wurde dieser Teil des Arbeitsspeichers von den Entwicklern des Amigasystems mit Fastmem bezeichnet.

Alle Programme die nun mit grafischen Objekten arbeiten, müssen diese innerhalb des Adressbereiches der Spezialbausteine ablegen. Das Betriebssystem trägt dieser Forderung Rechnung und stellt eine Routine bereit, die es dem sauberen Programmierer erlaubt Chipmem für solche Aufgaben explizit anzufordern. Dies geschieht über ein Flag, das beim Aufruf der Speicherreservierungsroutine den Wert 'MEMF_CHIP' erhält. Dieser Wert ist als Konstante in den jeweiligen Includedateien der verschiedenen Sprachen definiert. Wird dieses Flag nicht gesetzt, versucht das Betriebssystem zuerst Fastmem zu reservieren, da es bestrebt ist, möglichst viel Speicherplatz für Grafik freizuhalten.

Für Programme, die auf einem System mit 512 KByte entwickelt wurden, ist dieser Automatismus der erste Fallstrick. Während auf einem kleinen System das 'vergessene ?' Flag keine Auswirkungen zeigt, da das Programm sowieso in den dann nur vorhandenen unteren 512 KByte arbeitet, sieht dies auf einem Rechner mit mehr wie einem halben Megabyte anders aus. Hier liegt sowohl der Programmtext mit seinen statischen Variablen sowie aller, ohne das gesetzte Chipmemflag dynamisch angeforderter Speicher, außerhalb des Adressbereichs der oben erwähnten Bausteine. Für den reinen Programmcode ist dies nicht von Bedeutung, da dieser ja vom Prozessor ausgeführt wird (im Fastmem sogar schneller). Sind jedoch Grafikobjekte im Programmcode abgelegt oder Diskettenpuffer definiert, auf die diese Bausteine zugreifen müssen, läuft das Programm im günstigsten Fall nicht. Ein weniger günstiger Fall führt jedoch meist zum Systemabsturz.

Verzwick

Nun könnte man ja die zusätzliche

Erweiterung abschalten oder aus dem System entfernen, um dieses Manko zu beheben. Doch dies scheitert meist an der Art des Einbaus (huckepackgelötet oder zumindest einige Drähte fest verbunden). Ich werde jedoch gleich auf eine Änderung der Betriebssystemsoftware kommen, die dieses Problem löst. Da jedoch solche Programme in nächster Zeit hoffentlich aussterben werden, ist diese Änderung nicht der Hauptpunkt dieses Artikels.

Das wirkliche Problem liegt bei Programmen, die vermeindlich mit einem Megabyte kooperieren bzw. dieses mindestens benötigen, um mit allen Optionen arbeiten zu können. Diese Programme haben dies auch auf anderen Rechnern mit mehr als 512 Kbyte gezeigt, vielleicht sogar bei einem Freund, der eine externe Erweiterung oder einen A2000 besitzt.

Um dieses scheinbare Paradoxon zu lösen, muß man sich schon etwas näher mit den Vorgängen der Speichererkennung und Initialisierung nach dem Starten des Rechners beschäftigen.

In Abbildung 1. ist die Kaltstart-routine der Kickstartsoftware abgebildet. Dieses Unterprogramm wird nach dem Laden der Kickstart von Diskette und einigen vorausgehenden Testroutinen angesprungen. Die erste interessante Stelle beginnt ab Offset 01D6. Hier wird der Start- und Endbereich eines Adressbereichs festgelegt in dem das System nach einer externen Speichererweiterung sucht. Wird eine solche Erweiterung gefunden, wird ab Offset 01EE dieser Bereich gelöscht und später als Fastmem eingebunden. An dieser Stelle scheint es schon den ersten Konflikt mit der oben genannten Art von Speichererweiterung zu geben, denn diese liegen ja nicht in dem überprüften Adressbereich und werden daher auch von dieser Routine nicht erkannt. Eigentlich ist dies nicht weiter schlimm, da die Einbindung von solchem Speicher noch nach dem Systemstart mit Hilfe des Befehls AddMem eingefügt werden kann. Doch ist es Ihnen sicherlich aufgefallen, daß einige Erweiterungen doch vom System anerkannt werden müssen, da auf der Workbenchleiste ca. 900 000 Bytes frei gemeldet werden. Für dieses Verhalten des Betriebssystems ist

der Programmteil ab Offset 0208 verantwortlich. Er überprüft den Arbeitsspeicher von unten her bis an eine Grenze von 2 Megabyte auf vorhandenes Ram und merkt sich später die gefundene Obergrenze als Ausbauwert für das vorhandene Chipmem. An dieser Stelle tritt nun der größte Fehler bei der Speichereinbindung auf. Es wird nämlich ohne Rücksicht auf die physikalische 512 KByte Adressgrenze der Spezialchips dieser gefundene Wert als logischer Maximalwert für das vorhandene Chipmem eingetragen. Diese Tatsache können Sie z.B. durch den Aufruf des Programmes Avail vom CLI aus leicht überprüfen. Es wird in einem solchen System 1 Megabyte Chipmem melden.

Geleimt

Fordert nun ein Programm oder sogar das Betriebssystem selbst über die Speicherreservierungsroutinen einen Speicherbereich mit MEMF_CHIP an, kann es sein, daß es einen Bereich zugeordnet bekommt, der trotz gesetztem MEMF_CHIP Flag zwar logisch im Chipmem liegt, physikalisch jedoch außerhalb der Adressierungsmöglichkeiten der Spezialchips. Das Programm inclusive Betriebssystem werden so regelrecht verladen. Unvorhersagbare Ereignisse bis hin zum Systemabsturz sind die Folgen.

Abhilfe

Dieses eben beschriebene Problem läßt sich nun durch eine kleine Änderung im Programmcode der Kaltstart-routine beheben. An der Stelle mit dem Offset 020C wird der Wert 200000 als Startwert für das anschließend aufgerufene Unterprogramm in das Register a1 geladen. Der Rechner testet nun von Adresse 000000 bis zu diesem Wert auf kontinuierlich vorhandenen Speicher. Trifft er auf eine Lücke oder eine defekte Speicherstelle, so gibt er den bis zu diesem Zeitpunkt geprüften Adressbereich über das Register a3 zurück. Dieser Wert wird dann bei Offset 021A auf einen Minimalwert von 256 kByte getestet (minimal möglicher Speicher). Ändert man nun den oben angesprochenen Wert von 200000 auf 080000, prüft das Unterprogramm nur noch den Speicher von 000000 bis 080000. Dieser Wert entspricht nun genau

den physikalisch adressierbaren 512 KByte Chipmem. Um diese Änderung vorzunehmen muß man nun nicht gleich die gesamte Kickstartsoftware disassemblieren oder viel von 68000 Maschinencode verstehen. Solche kleinen nachträglichen Änderungen kann man leicht am fertigen Programm vornehmen. Der Profi nennt dies patchen (vom englischen flicken). Sie benötigen für die folgenden Schritte einen Diskettenmonitor (z.B. SmartDisk oder DiskZap).

1. Diskettenmonitor laden.
2. Kickstart Diskette V1.2 oder Rom Version in df0: einlegen.
3. Mit dem Diskettenmonitor die Spur 0, Sektor 2, Kopf 0 einlesen.
4. Da die Kickstartsoftware ab Sektor 1 auf der Diskette gespeichert ist, muß die Länge dieses ersten Blocks auf den vom Diskettenmonitor angezeigten Offset addiert werden. Die Länge eines Sektors ist genau 200. In der Praxis heißt das das das erste angezeigte Byte im Sektor 2 den Offset $0 + 200 = 200$ hat. Das Zweite $1 + 200 = 201$ usw. . Zum Vergleich ob Sie richtig sind, ist die angesprochene Stelle in Abbildung 2. noch einmal ausgedruckt. Die zu ändernde Stelle ist hier unterstrichen.
5. Ändern Sie den Wert entsprechend der Abbildung.
6. Schreiben Sie den geänderten Sektor auf Diskette zurück. Lassen Sie AUF KEINEN FALL die Prüfsumme neu berechnen, da diese im Gegensatz zu 'normalen' Disketten im Datenbereich liegt

und beim Zurückschreiben den Programmcode zerstören würde.



Die so geänderte Kickstart muß sich nach dem Einlegen ganz normal verhalten und die Workbench anfordern. Die Menüleiste der Workbench muß nach dieser Änderung ca. 450 000 Byte frei anzeigen. Die Erweiterung kann nun mit dem Programm Addmem auf korrekte Weise als Fastmem eingebunden werden.

Komfortabler

Nun ist es natürlich nicht sehr angenehm den AddMem Befehl nach jedem Hochfahren des Systems einzugeben. Auch sind manche Disketten schon so weit gefüllt, daß kein Platz für weitere Programme auf ihr ist, oder die Diskette ist gar durch einen Kopierschutz gesichert, der es nicht erlaubt zusätzliche Programme darauf zu überspielen. In einem solchen Fall greift nun der zweite Patch, den ich hier beschreiben möchte.

Dieser Patch wird ab dem Offset 01D6 und 01DC durchgeführt. Da an dieser Stelle der Suchbereich für Fastmem festgelegt ist, kann man die originalen Werte C00000 und DC0000 durch eigene ersetzen. Wählt man nun für diese Werte den Adressbereich der eingebauten Erweiterung, wird diese beim Hochfahren des Systems automatisch erkannt. Abbildung 3 zeigt die zu ändernden Werte. Es ist jeweils ein Beispiel für die Erweiterung aus Kickstart 9/87 und

68000er 9/87 angegeben. Es kann jedoch auch eine beliebige, an eigene Erweiterungen angepaßte Adressenkombination eingesetzt werden. Haben Sie die richtigen Adressen eingesetzt sollte sich der Rechner mit einem Megabyte melden. Das Programm Avail muß nun die richtige Speicherbelegung anzeigen. Dies ist in der hier besprochenen Ausbaustufe je 512 KByte Chipmem und 512 KByte Fastmem.

Einen kleinen Nachteil hat diese eben beschriebene Änderung der Suchgrenzen für das Fastmem. Dieser soll nicht verschwiegen werden. Speichererweiterungen die mit der original Kickstartsoftware sogenannte autokonfigurating Eigenschaften haben, werden nur an der Adresse C00000 korrekt erkannt. Ändern Sie nun wie im Absatz oben beschrieben diese Adresse auf die Startadresse Ihrer Speichererweiterung, wird eine andere, am Expansionsport angeschlossene, nicht mehr automatisch erkannt. D. h., Sie können nur einen Adressbereich durch die Kickstartsoftware automatisch einbinden lassen.

Schönheitsfehler

Die hier beschriebenen Änderungen haben bis jetzt leider einen Schönheitsfehler. Der Rechner verlangt nach jedem Reset nicht wie bisher die Work-

Die Kaltstartroutine der Systemsoftware enthält ab dem Offset 1CE den folgenden Code. Der Offset zählt ab der effektiven Adresse FC000 an die die Systemsoftware vom Bootrom geladen wird.

```

01CE FC01CE lea    400,a6      a6 gleich erste freie Ramadresse
01D2      suba.w #-276,a6    plus Länge der Sprungleiste
01D6      lea    C00000,a0  Suchbereich Fastmem
01DC      lea    DC0000,a1  Suchbereich Ende
01E2      lea    6(pc),a5    Rücksprungadresse nach a5
01E6      bra    FC061A      und Subroutine Suchen anspringen
01EA FC01EA move.l a4,d0      Ergebniss in a4 nach d0 kopieren
01EC      beq.s  FC0208      wenn kein Ram dann weiter mit FC0208
01EE      movea.l #C00000,a6  Anfangsadresse nach a6
01F4      suba.w #-276,a6    wie 01D2
01F8      move.l a4,d0      Löschen Ende
01FA      lea    C00000,a0  Löschen Anfang
0200      lea    6(pc),a5    Rücksprungadresse nach a5
0204      bra    FC0602      und Subroutine Löschen anspringen
0208 FC0208 lea    0,a0        Speicheruntergrenze
020C      lea    200000,a1   Testobergrenze
0212      lea    6(pc),a5    Rücksprungadresse nach a5
0216      bra    FC0592      Subroutine Speichergröße
021A FC021A cmpa.l #40000,a3     Ergebniss mit 256kb vergleichen
0220      bcs.s  FC0238      wenn kleiner dann Fehlerausgang
0222      move.l #0,0        Hilfeflag löschen
022A      move.l a3,d0        Speicherobergrenze nach d0
022C      lea    C0,a0        Stackbereich aussparen
0230      lea    E(pc),a5    Rücksprungadresse nach a5
0234      bra    FC0602      Subroutine Speicherlöschen anspringen
    
```

Abbildung 1:

0000:	4BFA0006	600003F4	41F80000	43F90020	K...~...A...C...
0010:	<u>00004BFA</u>	00066000	0372B7FC	00040000	..K...~...r.....
0020:	650E200B	41F800C0	4BFA000E	600003CC	e. .A...K...~...
0030:	303C00C0	6000037A	41F900DF	F000317C	0<...~...zA.....1!
0040:	7FFF0096	317C0200	0100317C	00000110	...1!...1!...1!
0050:	317C0888	018041EE	00544CEE	001C0222	1!...A..TL....."
0060:	7000323C	007D20C0	51C9FFFC	48EE001C	p.2<.> .Q...H...
0070:	022221CE	0004200E	46802D40	0026200C	..!... .F.-@.& .
0080:	6602200B	2E402D40	00360480	00001800	f. .@-@.6.....
0090:	2D40003A	2D4B003E	2D4C004E	61002E2A	-@...-K.>-L.Na..*
00A0:	6100029C	816E0128	43FA0020	30196700	a....n.(C... 0.g.
00B0:	008641F6	00002088	589042A8	00042148	..A... .X.B...!H
00C0:	00083019	1140000C	60E20142	000A0150	..0..@...~...B...P
00D0:	0008015E	0003017A	00090188	00040196	...^...z.....
00E0:	000101A4	0001016C	000201B2	000B01C2l.....
00F0:	000B01D2	000B01E2	000B01F2	000B0214
0100:	000F0000	090000FC	00A80600	0000024CL
0110:	002100BD	00FC0018	00000000	00014368	..!.....Ch
0120:	6970204D	656D6F72	79004661	7374204D	ip Memory.Fast M
0130:	656D6F72	790041FA	2C602D48	01302D48	emory.A.,~-H.0-H
0140:	01342D7C	00FC1CD0	01382D7C	0000FFFF	.4-!.....8-!....
0150:	013C3D7C	80000140	43EE0008	41FAFFA6	.<=!...@C...A...
0160:	700C32D8	51C8FFFC	204E43FA	16B82449	p.2.Q... NC...\$I
0170:	610011E8	3D400010	200C6724	41EE024C	a...=@... .g\$A..L
0180:	43FAFFA8	7400323C	0005200C	90880480	C...t.2<... ..
0190:	00001800	61001638	41F80400	7000600A	...a..8A...p.~
01A0:	41EE024C	203CFFFF	E800323C	00032448	A..L <....2<...\$H
01B0:	43FAFF6C	74F6D08B	90886100	1612224E	C...lt.....a... "N
01C0:	6100102E	41FA03AA	2248347C	00086006	a...A... "H4!...~
01D0:	47F00000	24CB3019	66F6302E	01280800	G...\$.O.f.O..(..
01E0:	00006732	41FA048E	327C0008	22C822C8	..g2A...2!...".
01F0:	2D7C00FC	08B2FFE4	2D7C42C0	4E75FDF0	-!.....-!B.Nu..

Der oben unterstrichene Wert ist in der ersten Zeile von 0020 in 0008 zu ändern.

Abb. 2:
So muß der zu ändernde Sektor aussehen

bench, sondern erneut die Kickstart. Dieses Problem ist in einer Prüfroutine der Systemsoftware verborgen. Dieses Unterprogramm überprüft die Summe alle Langworte im Kickstartbereich und testet das Ergebnis auf eine bestimmte Prüfsumme. Da die geänderte Kickstart jedoch eine andere Prüfsumme liefert, schließt das Programm auf eine beschädigte Systemsoftware und lädt deshalb diese von Diskette nach. Um die geänderte Kickstart in der Prüfsumme auf den verlangten Wert anzupassen ist erst einmal der Korrekturwert festzustellen. Diese Aufgabe kann jedoch der Rechner für uns übernehmen. Listing 1 und 1a enthalten jeweils ein Programm für den Seka Assembler und den Lattice C Compiler. Das Seka Programm ist nur innerhalb des Assemblers lauffähig (assemblieren mit A und dann G für go) und zeigt die neue Prüfsumme in Register d0 an. Das Programm in Lattice C gibt die Prüfsumme auf dem Bildschirm aus.

Haben Sie auf eine der beiden Möglichkeiten die neue Prüfsumme ermittelt, ist diese am Offset 0034 auf

der Diskette einzutragen. Der Offset bezieht sich ab dem Anfang von Spur 0, Sektor 1, Kopf 0. Dies ist der selbe Sektor in dem auch der erste Patch gemacht werden muß. Abbildung 3 zeigt noch einmal die richtige Stelle. Der Originalwert ist FFFFFFFF.

Die beschriebenen Änderungen gelten nur für die Version 1.2 und die Rom Version (die es für A1000 Besitzer auch auf Diskette gibt). Arbeiten Sie bitte nur an einer KOPIE Ihrer Original-Software. Alle Sektoren, die Sie mit Hilfe des Diskettenmonitors zurückschreiben, dürfen unter keinen Umständen eine neue Prüfsumme erhalten. Fragt der Diskettenmonitor 'recalculate checksum' ist dies immer zu verneinen.

Ansonsten kann ich Ihnen nur viel Erfolg bei eigenen Änderungen an der Kickstart wünschen. Für solche unter Ihnen, die sich die beschriebenen Änderungen nicht zutrauen oder deren Kickstart dann nicht mehr läuft, möchte ich auf die Public Domain Diskette 37 verweisen. Diese Diskette enthält ein Programm, das die beschriebenen Änderungen auto-

matisch auf der Diskette vornimmt. Desweiteren enthält diese Diskette ein Programm, das die ersten Sektoren der Kickstart von Diskette liest und in ein Format umwandelt, das von einem Disassembler verarbeitet werden kann. Auch das Checksum Programm ist auf dieser Diskette enthalten.

(GC)

```

0000: 11114EF9 00FC00D2 0000FFFF 002100A6 ..N.....!...
0010: 002100BD FFFFFFFF 65786563 2033332E .!.exec 33.
0020: 31383920 28333020 53657020 31393836 189 (30 Sep 1986
0030: 290D0A00 FFFFFFFF 0D0A0A41 4D494741 ).....AMIGA
0040: 20524F4D 204F7065 72617469 6E672053 ROM Operating S
0050: 79737465 6D20616E 64204C69 62726172 ystem and Librar
0060: 6965730D 0A436F70 79726967 68742028 ies..Copyright (
0070: 43292031 3938352C 20436F6D 6D6F646F C) 1985, Commodo
0080: 72652D41 6D696761 2C20496E 632E0D0A re-Amiga, Inc...
0090: 416C6C20 52696768 74732052 65736572 All Rights Reser
00A0: 7665642E 0D0A0000 65786563 2E6C6962 ved....exec.lib
00B0: 72617279 00004AFC 00FC00B6 00FC321E rary..J.....2.
00C0: 00210978 00FC00A8 00FC0018 00FC00D2 .!.x.....
00D0: 4E704FF9 00040000 203C0002 00005380 NpO....<...S.
00E0: 6EFC41FA FF1C43F9 00F00000 B1C9670E n.A...C.....g.
00F0: 4BFA000C 0C511111 66044EE9 000213FC K....Q..f.N....
0100: 000300BF E20113FC 000200BF E00149F9 .....I.
0110: 00DF0000 303C7FFF 3940009A 3940009C ....0<..9@..9@..
0120: 39400096 397C0200 0100397C 00000110 9@..9!...9!...
0130: 397C0444 0180307C 0008323C 002D43FA 9!.D..0!..2<.-C.
0140: 046C20C9 51C9FFF0 60002F5E 20380004 .l.Q...~/^8..
0150: 08000000 66782C40 D0AE0026 4680666E ....fx,@...&F.fn
0160: 720041EE 00227018 D25851C8 FFFC4641 r.A.."p..XQ...FA
0170: 665C202E 002A670C 20404BFA 000842AE f\..*g.@K...B.
0180: 002A4ED0 08790001 00BFE001 203AFE82 .*N..y.....:..
0190: B0AE0014 6638266E 003EB7FC 00080000 ....f8&n.>.....
01A0: 622CB7FC 00040000 6524286E 004E200C b,.....e$(n.N.
01B0: 67000086 B9FC00DC 00006212 B9FC00C4 g.....b.....
01C0: 0000650A 200C0280 0003FFFF 676A4DF8 ..e.....gJM.
01D0: 04009CFC FD8A41F9 00C00000 43F900DC .....A.....C...
01E0: 00004BFA 00066000 042A200C 671A2C7C ..K....~*..g..!
01F0: 00C00000 9CFCFD8A 200C41F9 00C00000 .....A.....

```

Hier müssen folgende Werte geändert werden :

Für die Erweiterung aus Kickstart Heft 9/87

```
0030: 290D0A00 XXXXXXXX 0D0A0A41 4D494741
```

An der Stelle XXXXXXXX ist die Zahl einzusetzen die das Prüfsummenprogramm liefert.

```
01D0: 04009CFC FD8A41F9 00080000 43F90010
```

```
01F0: 00080000 9CFCFD8A 200C41F9 00080000
```

Für die Erweiterung aus 68000er Heft 9/87

```
0030: 290D0A00 XXXXXXXX 0D0A0A41 4D494741
```

An der Stelle XXXXXXXX ist die Zahl einzusetzen die das Prüfsummenprogramm liefert.

```
01D0: 04009CFC FD8A41F9 00100000 43F90018
```

```
01F0: 00100000 9CFCFD8A 200C41F9 00100000
```

Abb. 3:
Die Änderungen für
automatische Er-
kennung

Für Lattice C :

```
checksum.c:
```

```
extern unsigned long checksum();
```

```
main()
```

```
{
  unsigned long cs;

  cs = checksum();
  printf("Checksumme der momentanen Kickstart ist : %x\n",cs);
  return(0);
}
```

```
cksum.asm:
```

```

CSECT    text
XDEF     _checksum

_checksum lea    $fc0000,a0
          move.l  #$10000,d1
          move.l  d1,d2
          swap   d2
          clr.l  d0

```

```

lab1     bra.s   lab2
          add.l  (a0)+,d0
          bcc.s  lab2
          addi.l #1,d0
lab2     dbra   d1,lab1
          dbra  d2,lab1
          not.l d0
          rts
          END

```

Anleitung : Beide Teile getrennt eintippen.
Dann 'Lc checksum' eingeben.
'Asm cksum.asm' eingeben.
'Blink from lib:c.o,checksum.o,cksum.o to checksum
lib lib:lc.lib,lib:amiga.lib'

Der Aufruf von Blink muß bei Ihnen in einer Zeile
erfolgen.
Checksum aufrufen. Das Ergebniss wird auf dem
Bildschirm angezeigt.

Für SEKA

```
checksum.s:
```

```

lea      $fc0000,a0
move.l  #$10000,d1
move.l  d1,d2
swap    d2
clr.l   d0
bra.s   ba
bo:     add.l  (a0)+,d0
        bcc.s  ba
        addi.l #1,d0
ba:     dbra  d1,bo
        dbra  d2,bo
        not.l d0
        illegal
    
```

Anleitung : Seka aufrufen. Workspace = 100
 <ESC> drücken und Listing eingeben.
 <ESC> drücken und 'A' <RETURN> eingeben.
 <RETURN> eingeben.
 'G' <RETURN> eingeben.
 <RETURN> eingeben.
 Das Ergebniss steht in d0.

ACHTUNG : Beide Programme berechnen die Prüfsumme der momentan geladenen Kickstart. Sie müssen also mit der geänderten Kickstart booten um auf das richtige Ergebniss zu kommen.

Listing 1 und 1a Das Prüfsummenprogramm.

PUBLIC DOMAIN SOFTWARE

180 Disketten für AMIGA

Kopiergebühr incl. Disc **DM 8,- je Disk**

zuzügl. Porto/Bearbeitungsgebühr
 DM 5,- je Lieferung bei Vorkasse
 unabhängig von der Anzahl der Disket-
 ten, DM 10,- bei Nachnahme.

PUBLIC DOMAIN-Liste
 gegen Rückporto 0,80 DM
 (in Briefmarken) anfordern

3,5" Leerdisketten
 Spitzenqualität 10 St. 29,- DM
 100 St. 270,- DM

Zweitlaufwerk
 (anschlußfertig) für AMIGA 360,- DM

Kopierservice Public-Domain Software
 Dipl.-Betriebswirt Chr. Bellingrath
 Hans-Böckler-Str. 55 · 5860 Iserlohn · Tel. 02371/24192

Digi-View V2.0 DM 379,-

Golem-2MB-Rambox DM 949,-
 Amigos-3.5"-Laufw. DM 349,-
 Aztec C V3.4 Prof. DM 399,-
 Public Domain ab DM 5,50
 Programmers Handbook .. DM 45,-
 Modula II V3.0 Dev. DM 294,95
 MCC-Makroassembl. DM 149,-
 Metacomco Pascal DM 175,-
 Lattice C V3.1
 mit Textutilities DM 329,-
 MCC-Shell DM 99,95
 Silent Service DM 69,-
 Amiga 2000 mit IBM-Karte
 Copy II-Option Board V4.3
 - Hard- u. Software zum Kopieren
 von 3.5"-Amiga und 5.25/3.5"-Disk.
 für IBM. Kontinuierliche Up-Dates
 sind durch uns erhältlich . DM 219,-
 Turbo Prolog V1.1 DM 199,-
 Eco-C-Compiler DM 49,-
 Kostenlose Prospekte, auch für Atari ST

CWTE

Computerversand CWTG

Joachim Tiede

Bergstraße 13 · 7109 Roigheim

☎ 0 62 98 / 30 98 von 17-19 Uhr sonst Anrufbeantw.

HÄNDLERANFRAGEN ERWÜNSCHT!

AMIGA - Floppy ab DM 299.--

- Garantiert kompatibel - anschlussfertig - leise und zuverlässig - farblich
 passendes, formschönes professionelles Metallgehäuse - geschirmtes Ka-
 bel - Laufwerk: NEC FD1035LP - ohne Ausgangsbuchse - abschaltbar -
 als Bausatz nur DM 299.--, als Fertiggerät nur DM 349.--

Lieferung per Nachnahme - zuzügl. 6.-DM Porto/Verp. - tel. Bestellung - 6 Monate Garantie

Dipl. Ing. Gerhard Trumpp
 Mitterlängstr. 7, 8039 Puchheim

Tel. 089 / 80 68 23 17 - 22 Uhr

UND JETZT DER PROFI.

**BECKERtext
Amiga!**

„... wirklich hervorragend. Keine derzeit auf dem AMIGA-Markt erhältliche Textverarbeitung kann Vergleichbares bieten. Man darf deshalb gespannt auf die ‚Profiversion‘ (BECKERtext) warten, die bald zu einem Preis von DM 199,- erhältlich sein wird und mit einer Vielzahl zusätzlicher Features aufwarten soll.“

Soweit die Zeitschrift KICKSTART (9/87) über unsere Textverarbeitung TEXTOMAT Amiga.

Nun - der angekündigte Textprofi ist jetzt erhältlich. Und neben all den leistungsstarken Features von TEXTOMAT Amiga wie Direktformatierung mit allen Textattributen am Bildschirm (WYSIWYG), Verknüpfung von Text und Grafik, Shortcuts und einfache Druckeranpassung bietet BECKERtext Amiga dann auch Dinge, die den Zusatz „Profiversion“ rechtfertigen.

Komfortables Rechnen im Text z.B. - unterstützt durch Dezimaltabulatoren können Sie nicht nur spalten-, sondern auch zeilenweise rechnen. Mehrspaltige Druckausgabe mit bis zu 5 Spalten, ein ONLINE-Lexikon, das wahlweise während des Schreibens oder nachträglich Ihre Texte auf Rechtschreibung überprüft und individuell erweiterbar ist, bis zu 999 Zeichen pro Zeile bei horizontalem Scrolling und Formulare als nicht überschreibbare Eingabemaske sowie Datenaustausch mit anderen Programmen über Clipboard sind weitere, hervorstechende Features. Empfohlen wird Amiga mit 1 MByte RAM.

BECKERtext Amiga - Professionalität zum unglaublichen Preis

DM 199,-

DATA BECKER

Merowingerstr. 30 · 4000 Düsseldorf · Tel. (02 11) 310010

BESTELL-COUPON

Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:

per Nachnahme

Verrechnungsscheck (liegt bei

Name

Strasse

Ort

Baukastenprinzip

TDI MODULA - 2

Auf dem Markt der Programmiersprachen gibt es einen völlig neuen Kandidaten: Modula-2. Das Leistungsvermögen und die Konkurrenzfähigkeit dieser Sprache sollen in diesem Text beurteilt werden.

Niklaus Wirth, Professor an der Züricher ETH, der 1971 die Sprache PASCAL definierte, scheint mit seinem ersten Produkt nicht vollauf zufrieden gewesen zu sein, da er Anfang der achtziger Jahre eine neue Programmiersprache „erfand“. In dieser Sprache sollte die Strukturierung des Programmierens noch verstärkt werden. Das machte schließlich das Besondere an dieser Programmiersprache aus und brachte ihr auch den Namen ein: Die Modularisierung der Programme. In Modula (MODULAR LANGUAGE) wird ein Problem in sehr kleine Teilprobleme zerlegt. Diese Bruchstücke können sehr viel einfa-

cher gelöst werden und haben zudem folgenden Vorteil: Hat man nach einer gewissen Programmierzeit einen Grundstock an Lösungen für Teilaufgaben gesammelt, kann man durch ihre Verwendung in neuen Programmen viel einfacher arbeiten. Ansonsten ist die Sprache dem PASCAL sehr ähnlich, was sicher kein Nachteil ist.

Seit Kurzem liegt nun eine Implementation für den Amiga vor. Sie stammt von der Modula-2 Software Ltd. aus England. Derzeit sind drei verschiedene Ausführungen auf dem Markt: Die einfachste und zugleich billigste ist die Standard-Version. Sie stellt nur das Grundsystem zur Verfügung. Im Lieferumfang enthalten sind zwei Disketten und das Handbuch. Auf den Disketten findet man Editor, Compiler, Linker und die externen Module. Außerdem werden

noch viele Demo-Programme und die Definition-Module mitgeliefert. Speziell die Demo-Programme sind für den Anfänger recht nützlich, um einen Einblick in Modula gewinnen zu können. Eine weitere Variante ist die „Developer-Version“. Auf der dritten Diskette, die hier mitgeliefert wird, befinden sich neben weiteren Demos auch die Entwickler-Utilities. Sie dienen zum Analysieren von Modulen. Schließlich gibt es die kommerzielle Version, die zusätzlich noch die Implementations-Module beinhaltet. Unserem Test stand die Developer-Version zur Verfügung. Eine Besonderheit des MODULA von Software Ltd. ist der vollständige Durchgriff auf die Betriebssystemebene. Dadurch ist eine wesentlich bequemere Programmierung als in „C“ möglich.

Das Handbuch

Das mitgelieferte Handbuch erweist sich mit 400 Seiten schon als richtig dicker Brocken. Es ist, wie meist üblich, in Englisch gehalten, der Anwender hat also am Anfang mit zwei Problemen zu kämpfen: Der neuen (Programmier-)Sprache und der fremden Sprache. Als Lehrbuch (MODULA für „blutige Laien“) ist das Handbuch ohnehin ungeeignet. Ein kurzer Abschnitt am Anfang des Buches geht auf die Besonderheiten von MODULA-2 ein. Bei mir erweckte das Handbuch den Eindruck, als ginge man davon aus, daß alle Käufer perfekte PASCAL-Programmierer seien. Wer in PASCAL firm ist, kann es auch mal ohne MODULA-Lehrbuch versuchen. Allen anderen kann nur geraten werden, sich ein Lehrbuch anzuschaffen.

Man wird sich nun fragen, womit die 400 Seiten gefüllt wurden. 85 Prozent des Handbuches sind mit den Listings der Definition-Module belegt. Diese Listings findet man auch auf der zweiten Diskette wieder.

```

MODULE sieve;
FROM Terminal IMPORT WriteString,WriteLn;
FROM Conversions IMPORT ConvertToString;

CONST size = 8190;

VAR  flags : ARRAY[0..size] OF BOOLEAN;
     i,prime,k,count,iter : CARDINAL;
     s : ARRAY [0..39] OF CHAR;
     done : BOOLEAN;

BEGIN
  WriteString ("10 Iterationen");
  WriteLn;
  FOR iter := 1 TO 10 DO
    count := 0;
    FOR i := 0 TO size DO flags[i] := TRUE;
    END;
    FOR i := 0 TO size DO
      IF flags[i] THEN
        prime := i*2+3;
        k := i + prime;
        WHILE k <= size DO
          flags[k] := FALSE;
          INC(k, prime);
        END;
        INC(count);
      END;
    END;
  END;
  ConvertToString(LONGCARD(count),10,FALSE,s,done);
  WriteString (s);
  WriteString (" Primzahlen gefunden");
  WriteLn;
END sieve.

```

```

MODULE hello;
FROM Terminal IMPORT WriteString;
BEGIN
    WriteString(" hello ")
END hello.

```

Der Editor

MODULA-2 wird mit einem eigenen Editor ausgeliefert, mit dem man vollständig mausgesteuert arbeiten kann. Zusätzlich stehen für manchen Befehl auch <AMIGA>-Tastenkombinationen zur Verfügung. Die Auswahl der Editierbefehle ist für den praktischen Einsatz völlig ausreichend. Positiv fällt die Möglichkeit auf, mit der Maus Marken zu setzen und diese dann beliebig anzuspringen. Außerdem werden Fehler, die beim Compilieren auftraten, angezeigt. Auf diese Weise wird das Beheben der Fehler stark vereinfacht. Ein Manko des Editors darf jedoch nicht verschwiegen werden: Wenn man den Programmtext auf die RAM-Disk speichern will, wird man sich schwertun. Der Editor meldet sich dann mit der Meldung „Not enough room on disc to write new file“. Ein Effekt, der wohl auf der automatischen Dimensionierung der RAM-

auf die häufig zugegriffen wird (das sind auch Quelltexte bei der Programmierarbeit), auf die RAM-Disk legen wollen. Dies ist beim MODULA-Editor nicht möglich. Ansonsten ist es zu begrüßen, daß man nicht, wie andere Sprachen, auf den DOS-eigenen ED zurückgreift, der ja auch seine Eigenheiten hat.

Der Compiler

Beim MODULA-Compiler handelt es sich um einen 5-Pass-Compiler. Im ersten Durchlauf wird eine Importliste erstellt. Dies bedeutet, daß der Compiler alle Module, die in das zu compilierende Programm importiert werden sollen, in eine Liste geschrieben werden, um beim späteren Linken die richtigen Dateien anlinken zu können. Nächster Pass ist die Syntax-Analyse, in der überprüft wird, ob das Programm auch die Sprachvereinbarung einhält. Im dritten Durchgang werden Deklarationen überprüft, im darauffolgenden

übergeben wird. Außerdem erzeugt der Compiler bei auftretenden Fehlern im Quelltext eine .erm-Datei, die der Editor zur Fehleranzeige benötigt.

Die Compilierzeiten sind eher mäßig, was wohl darauf zurückzuführen ist, daß der Compiler etliche Diskettenzugriffe durchführt, um auf die externen Module zugreifen zu können. Wer nun alle zum Compilieren benötigten Files auf die RAM-Disk kopieren will, sollte mindestens ein MegaByte RAM besitzen, da schon bei einem 512k-AMIGA das Starten des Compilers (wegen des zu geringen Speicherplatzes) unmöglich ist, wenn man ihn gleichzeitig in der RAMDisk hat. Selbst bei einem Megabyte wird es eng, wenn man alle externen Module ins RAM laden will. Da die meisten AMIGAs wohl mit 512k betrieben werden, sind die Compilierzeiten auf einem 512k-1000er mit zwei Floppys ermittelt worden. Beim Testen erwies sich der Compiler als etwas eigensinnig. Mancher Absturz wurde ohne erkennbaren Grund produziert. Bei anderen Compilerversuchen, die mit Fehlern abgebrochen wurden, konnten anschließend bestimmte Dateien nicht benutzt werden, da sie vom Compiler nicht mehr freigegeben wurden. Dabei geht es hauptsächlich um zwei temporäre Dateien (il1.tmp und il2.tmp), die vom Compiler selbst erzeugt werden. Startet man einen zweiten Compilerversuch, so kann dieser nicht durchgeführt werden, da diese Dateien vor dem Zugriff (was ein Überschreiben der Datei ja auch ist) geschützt ist. Gelöscht werden können die Dateien ebenfalls nicht, da sich der AMIGA mit der Meldung „Object in use“ beharrlich weigert. In diesem Falle hilft nur ein Reset. Des weiteren war es uns unmöglich, ein auf der Diskette gespeichertes Implementations-Modul zu compilieren. Der Compiler läuft zwar am Anfang problemlos, hört aber mitten in der Statement-Analyse auf und der CLI meldet sich mit seinem Prompt. Dabei wird zwar kein Link-File erzeugt, dafür aber einige schwarze Striche auf dem Bildschirm – wohl als Trostpflaster, da bei 60 Sekunden Compilierzeit sonst gar nichts erzeugt würde. Normale Programme und Definition-Module wurden relativ problemlos compiliert.

```

link code version: 4H 35H
checksum: o.k.
scmod header: MODULE hello, key = D92H 4C6H 8AH checksum: o.k.
import Terminal, key = AE3AH 498H 84H, modnum = 1 checksum: o.k.
data size, number of bytes = 0 checksum: o.k.
scmod init code, procnum = 0, entrypoint = 0H, number of bytes = 44
DECODE ----- INSTRUCTION
0H 4EF9 0000 0000 JMP 00000000
6H 7000 MOVEQ #0000,D0
8H 4EB9 0000 0000 JSR 00000000
EH 4E56 0000 LINK A6,#0000
12H 3F3C 0006 MOVE.W #0006,-(A7)
16H 4879 0000 0000 PEA 00000000
1CH 4EB9 0000 0000 JSR 00000000
22H 5C8F ADDQ.L #0006,A7
24H 4E5E UNLK A6
26H 4EF9 0000 0000 JMP 00000000
checksum: o.k.
ref ext init call at 0H checksum: o.k.
load util:, procnum = 0 checksum: o.k.
ref util: at AH, procnum = 3 checksum: o.k.
ref own string: at 18H checksum: o.k.
ref ext proc call at 1EH, procnum = 6, modnum = 1 checksum: o.k.
ref ext init call at 26H checksum: o.k.
string data:, number of bytes = 8
NO DECODE ----- DATA
0H 2068 656C 6C6F 2000 checksum: o.k.
scmod end checksum: o.k.

```

Disk beruht, der aber keineswegs unvermeidbar ist. Da der AMIGA bei Diskettenzugriffen wirklich nicht gerade schnell ist, wird man Dateien,

Pass eine Überprüfung der Aussagen auf Richtigkeit durchgeführt. Der letzte Teil des Compilers erzeugt den Code, der danach an den Linker

Die beschriebenen Fehler sollten jedoch nicht überbewertet werden, da man davon ausgehen kann, daß es sich um Bugs handelt, die wohl schnellstens entfernt werden. Man sollte aber beim Kauf darauf achten, daß man eine Version ohne diese Fehler erwirbt, da sie das Arbeiten erschweren.

Auftrumpfen kann MODULA-2 bei den Ausführungszeiten. Die fertigen Programme laufen alle sehr schnell und brauchen den Vergleich mit anderen Hochsprachen nicht zu scheuen.

Der Linker...

...bewegt sich in seinem Tempo auf derselben Ebene wie der Compiler, da auch er viele Diskettenzugriffe benötigt. Er muß ja alle Module an das Programm anbinden und so das lauffähige Programm erzeugen. Dabei bietet er zwei Optionen. Einmal die DT-Option, durch die der Linker Informationen für den Debugger anhängt. Daneben gibt es die o-Option: Ruft man den Linker etwa in der Form „link progname o“ auf, so veranlaßt ihn dies alle Routinen, die nicht benötigt werden, nicht anzulin-

ken. In den importierten Modulen sind ja meist Prozeduren enthalten, die das Programm zwar nicht benötigt, die aber normalerweise ebenfalls angelinkt werden. Erstaunlicherweise ist die Link-Zeit trotz des zusätzlichen Aufwandes kürzer.

Das Developerpaket

MODULA-2 von Modula Software Ltd. hat auch einen Post-Mortem-Debugger zu bieten. Post-Mortem bedeutet, daß man dem Programmfehler erst nach dem Absturz zu Leibe rückt. Um ein Programm debuggen zu können, muß man ins Programm den Trapper importieren, das Programm mit der DT-Option linken und – das Programm muß natürlich abstürzen. Dann werden mit dem Debugger aber einige sehr nützliche Hilfen zur Seite gestellt.

Außerdem gehören zum Entwicklerpaket noch einige Decoder, die die von MODULA-2 erzeugten Files entschlüsseln können. Die Quelldateien können zwar nicht wieder erzeugt werden, aber alle nur erdenklichen Informationen werden zur Verfügung gestellt. Für Programmentwickler sind diese Programme eine sehr große

Hilfe. Man sollte aber schon Grundkenntnisse in Assembler besitzen, da ein großer Teil der Informationen als Assemblerlisting angeboten wird.

Fazit

MODULA-2 verdient es, auf dem AMIGA Beachtung zu finden. Insgesamt kann man mit der Implementation von Modula-2 Software Ltd. zufrieden sein, die Bugs werden sicher behoben, oder sind es schon. Für Programmierer, die vom PC mit der Haussprache Pascal kommen, ist es die beste Möglichkeit, auf dem AMIGA einzusteigen, da MODULA-2 von Software Ltd. ja vollen Zugriff auf das Betriebssystem erlaubt, was bei den Pascaldialekten für den AMIGA nur schwer möglich ist. Zudem ist die Geschwindigkeit so hoch, daß Programmierer, die kurze Programmlaufzeiten wünschen, zur Zeit keine andere Hochsprache mit dieser Leistung finden werden.

(chk)

+ Guter Editor, viele Beispielprogramme, Durchgriff auf die Betriebssystemebene des Amiga, Linker kann optimierten Code erzeugen, sehr gute Dokumentation der Definition-Module, im Developer-Paket sehr nützliche Decoderprogramme, sehr schnelle compilierte Programme, groe Anzahl guter Beispielprogramme auf Disk.

– Keine Einführung in die Programmierung vorhanden, Compiler und Linker etwas langsam, Compiler in der getesteten Version nicht fehlerfrei, Editor arbeitet nicht mit der RAM-Disk.

	MODULA 2	MODULA 2 *	MCC-Pascal
Länge sieve	12564 Byte	4028 Byte	8354 Byte
Länge hello	5508 Byte	1444 Byte	1444 Byte
Compilierz.	42 sec	42 sec	19 sec
Linkzeit	101 sec	88 sec	81 sec
Ausführungz.	7 sec	7 sec	65 sec

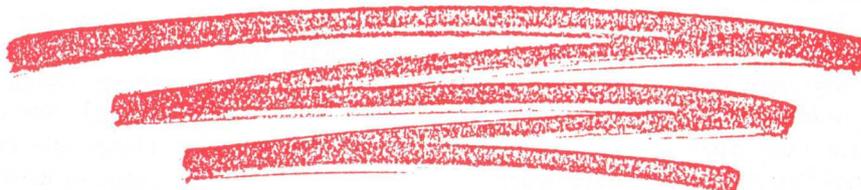
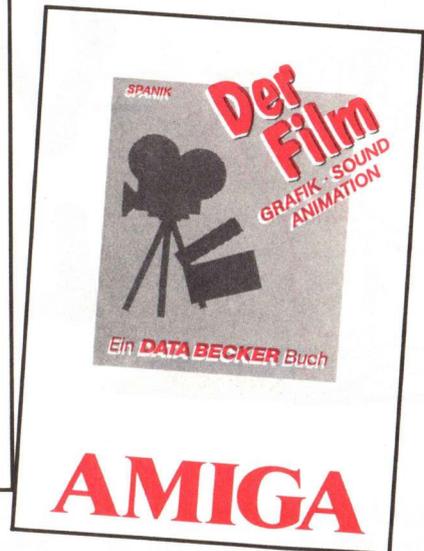
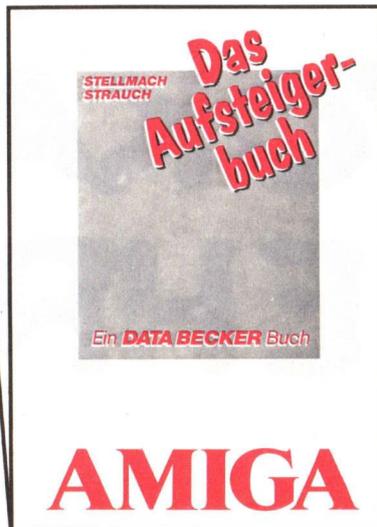
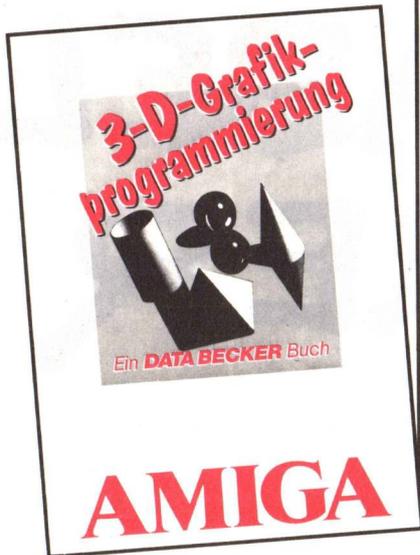
Die mit * gekennzeichneten Spalte zeigt die Werte mit o-Option beim Linken.

Preise:

MODULA-2 (Standard V2.0) 298 DM
 (Developer V2.0) 448 DM
 (Commercial V2.0) 848 DM

Philgerma GmbH
 Ungererstraße 42
 8000 München 40

Top aktuell:



3-D-Grafik auf dem Amiga – hier ist der Schlüssel zu dieser faszinierenden Welt. In diesem Buch werden Grafikalgorithmen beschrieben und erläutert, die es in sich haben. Mit ihnen können Sie absolut realistisch gestaltete Bilder erstellen: Die einzelnen Komponenten eines Bildes lassen sich dabei mit einem Editor problemlos eingeben und solange durch die Wahl verschiedener Materialien, Farben, Lichtquellen und Spiegelungen verfeinern, bis sie eine absolut naturgetreue Darstellung erreicht haben. Die Berechnung des Bildes übernimmt das Programm automatisch, in allen Auflösungen mit bis zu 4096 Farben. 3-D-Grafikprogrammierung – ein Muß für alle, die Spaß an tollen Grafiken haben. **Amiga 3-D-Grafikprogrammierung Hardcover, 283 Seiten inkl. Diskette, DM 59,-**

Jahrelang haben Sie mit dem Commodore 64 gearbeitet. Nun endlich – der Traum ist erfüllt: Der Amiga steht auf Ihrem Schreibtisch. Doch nun? Wieder ganz von vorn anfangen? Diese Arbeit können Sie sich sparen mit dem Aufsteigerbuch. Hier finden Sie Schritt für Schritt einen schnellen Einstieg im Umgang mit der Workbench, dem CLI und dem Multitasking. Ob AmigaBASIC, C oder 68000 Assembler – eine effiziente Einführung dieser Sprachen hilft Ihnen, sich für die richtige zu entscheiden. Alles immer im Vergleich zu Ihrem alten Rechner. Auch die vorhandene Hardware läßt sich sinnvoll einsetzen. Der gute alte C64 mit seinem Drucker als 60 KByte großer Druckerpuffer beispielsweise. Außerdem enthält das Buch für den eigentlichen Daten-Transfer von Texten, Bildern, Dateien und BASIC-Programmen fertige Konvertierungsprogramme. Was will ein Umsteiger mehr?

Amiga Aufsteigerbuch
ca. 350 Seiten, DM 39,-
erscheint ca. 10/87

Der Amiga ist das ideale Werkzeug für Kreative – aber einen Film mit ihm erstellen? Warum nicht? Dieses Buch hilft Ihnen dabei. Vom Drehbuch bis zum fertigen Drei-Minuten-Film. Mit allem, was dazu gehört: Grafik, Sound, Animation und und und. Dabei arbeiten Sie mit den verschiedenen DeLuxe-Programmen – auch mit DeLuxe-Video 1.2 und DeLuxe-Paint II, lernen deren fantastische Möglichkeiten und Anwendungsgebiete kennen. Da Sie als angehender Filmemacher auch mit Videorecorder und Sounddigitizer professionell umgehen müssen, wird das nötige Know-how gleich mitgeliefert. Wann dürfen wir zu Ihrem ersten Oskar gratulieren? **Amiga – Der Film Hardcover, ca. 400 Seiten DM 49,- erscheint ca. Ende 11/87**

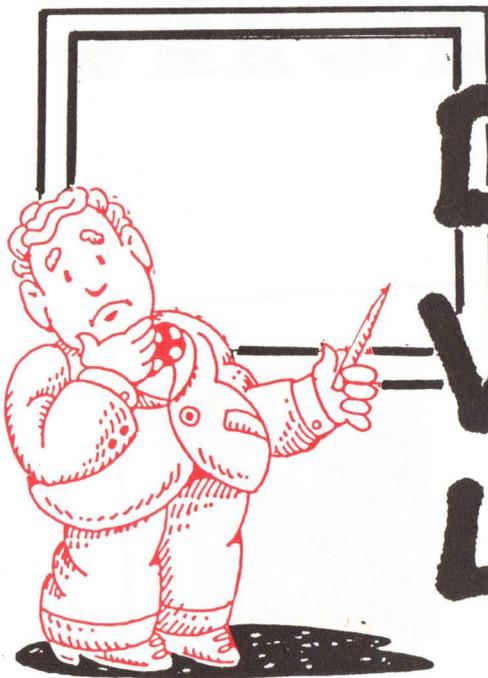
DATA BECKER

Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 31 00 10

BESTELL-COUPON
Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:

Zzgl. DM 5,- Versandkosten
unabhängig von der bestellten Stückzahl
 per Nachnahme Verrechnungsscheck liegt bei

Name _____
Straße _____
Ort _____



DER VOKABEL- LEHRER

Viele Schüler tun sich schwer mit dem Lernen von Vokabeln. Oft ist auch keine Person zur Hand, die einen Schüler die gerade gelernten Wörter abfragt. Das Programm Vocabcraft schafft hier Abhilfe. Egal, ob Englisch, Französisch oder Spanisch gelernt wird – das Programm bewältigt alle Sprachen problemlos. Der Programmierer hat sich sehr viel Mühe mit dem Programm gegeben und es so flexibel wie möglich gehalten.

Das Programm Vocabcraft V1.1 ist für den Commodore Amiga mit 512 KByte in AmigaBasic geschrieben. Es handelt sich um ein Vokabel-Lernprogramm, in das man die Vokabeln lektionsweise eingibt und dann auf Diskette sichert. Die Vokabeln können angesehen, abgefragt und ausgedruckt werden. Das Programm wird über eine grafische Benutzeroberfläche mit Pull-Down-Menüs und Windows gesteuert. Auf der Datendiskette wird ein 'Quickdir' (= Schnell-Directory) angelegt, so daß das Vokabelverzeichnis nach einem extrem kurzen Zeitraum zur Verfügung steht.

Bedienungsanleitung

Disk-Menu: Im Disk-Menu findet man an erster Stelle den Menüpunkt

'Alles löschen'

Das heißt, das Programm wird neu gestartet, vorhandene Vokabeln werden aus dem RAM gelöscht und die Uhr läuft nach den Werten der inneren Uhr weiter. Wählt man den Punkt

'Datei laden',

so werden die auf der Diskette gesicherten Daten aufgelistet und man kann durch einen Mausklick mit der linken Maustaste eine Datei auswählen und einladen. Danach wird zu dem Menüpunkt

'Datei ändern'

aus den Datei Menu verzweigt. Der nächste ist

'Datei speichern'.

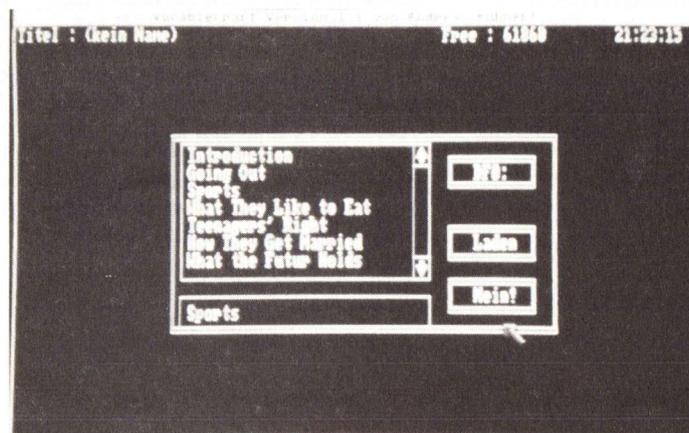
Nach der Anwahl dieses Menüpunktes werden die im RAM enthaltenen Daten unter dem aktuellen Namen auf der Diskette gesichert. Haben die Daten noch keinen Namen, so wird nach

'Datei mit Namen speichern'

verzweigt. Hier werden die im RAM gespeicherten Daten nach Eingabe des Dateinamens auf der Diskette festgehalten.

Mit

'Datei löschen'



Eine komfortable Auswahlbox für die verschiedenen Lektionen

kann eine Datei natürlich auch wieder von der Diskette verbannt werden. Man darf eine Datei allerdings nicht vom CLI aus löschen, da sie dann im Quickdir noch aufgeführt wird. Dem Programm macht man mit

'Programmende'

ein vorläufiges Ende. Man kann auch mit

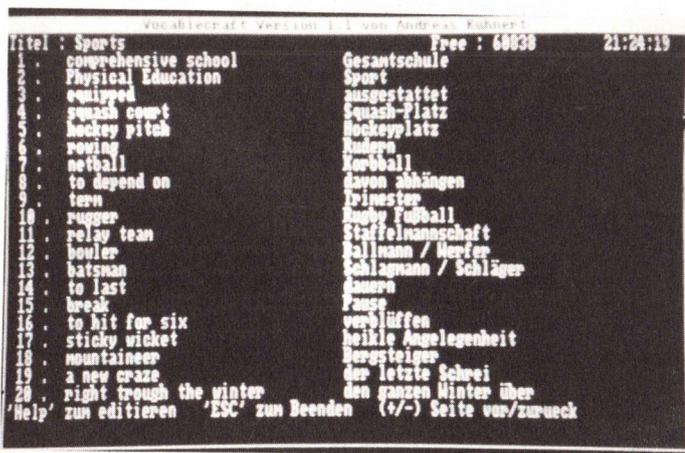
'Datei drucken'

sich die aktuelle Datei schwarz auf weiß ausgeben lassen, um sich den Vokabeln intensiver widmen zu können.

Datei Menu: Nach der Auswahl des Punktes

'Datei erstellen', bzw. 'erweitern'

werden neue Vokabeln eingegeben. Wenn schon Vokabeln im RAM enthalten sind, werden die Eingaben an die im RAM enthaltenen angehängt. Nach dem Verlassen des Menüpunktes, durch ein Drücken der ESC-Taste, wird zum Menüpunkt



'Datei ändern'

verzweigt, um die Eingabe noch einmal kontrollieren zu können. Dann werden die im RAM vorhandenen Daten auf dem Bildschirm angezeigt. Da nur 20 Vokabeln untereinander Platz finden, kann man durch einen Druck auf die Taste (+) bzw. (-) 20 Vokabeln vor- bzw. zurückblättern. Nach einem Druck auf die HELP-Taste und Eingabe der Vokabelnummer kann eine Vokabel und die Be-

deutung neu eingegeben oder geändert werden. Wenn eine Änderung erfolgt, d. h. nur <Return> gedrückt wird, bleibt die Vokabel bzw. die Bedeutung unverändert. Ähnlich verhält es sich mit dem Menüpunkt

'Dateiteile ändern',

genauer gesagt, ist bis auf den Druck auf die Help-Taste alles gleich. Nach diesem Druck wird auch eine Vokabelnummer verlangt, diese Vokabel wird allerdings nicht geändert, sondern ganz gelöscht.

Programm-Menu: An erster Stelle dieses Menüpunktes steht der Punkt

'Test Vokabeln'.

Die Vokabeln werden durch den Zufallsgenerator abgefragt, und zwar wird die Bedeutung angezeigt und die Vokabel wird abgefragt. Wenn dabei eine falsche Eingabe erfolgt, so wird die richtige Vokabel kurz angezeigt und die Abfrage weitergeführt. Das Programm merkt sich den Fehler und mischt die falsche Vokabel unter die anderen Vokabeln. Bei richtiger Ein-

Die Vokabelauflistung eignet sich auch gut zum Lernen

gabe wird diese Vokabel nicht mehr abgefragt. Am Schluß, wenn alle Vokabeln abgefragt und nach dem genannten System aussortiert worden sind, wird eine Statistik der Fehlerquote in Prozent angezeigt. Ein vorzeitiges Abbrechen der Abfrage ist mit der ESC-Taste möglich. Der zweite und letzte Punkt aus diesen Menu ähnelt auf den ersten Blick wieder den vorangegangenen. Er heißt

'Test Bedeutungen'.

Er unterscheidet sich nur darin von dem vorangegangenen, daß jetzt die Bedeutung gefragt und die Vokabel angezeigt wird.

Sprache-Menu: Nach der Anwahl von

'An'

(Sprache an), schaltet man die englische Sprache ein. Es werden beim Testen nur die Vokabeln ausgesprochen, nicht die Bedeutungen. Es empfiehlt sich aber, diesen Menüpunkt nur anzuwählen, wenn englische Vokabeln im Speicher sind, aber das wird sich wohl jeder denken können. Übrigens passen 600 Vokabeln inclusive Bedeutungen in den Speicher. Der soeben beschriebene Zustand kann durch die Anwahl des Punktes

'Aus'

(Sprache aus) wieder rückgängig gemacht werden.

Uhr-Menu: Auch hier ist der Punkt

'An'

an erster Stelle im Pull-Down-Menu. Die Uhr wird, falls sie inaktiv sein sollte, wieder aktiviert. Ausschaltet wird die Uhr, wenn man den nächsten Punkt anwählt. Es erscheint in der rechten oberen Ecke nicht mehr die aktuelle Uhrzeit, sondern die Meldung 'Uhr aus'. Mit

'Uhrzeit setzen'

kann man gegebenenfalls die Uhrzeit korrigieren. Die Uhr wird dann auch wieder aktiviert. Im allgemeinen richtet sich die programmeneigene Uhr nach den Werten der Systemuhr.

Dies war eine Bedienungsanleitung des Programms, die (hoffentlich!) keine Fragen mehr offen läßt. Ein sehr wichtiges Merkmal für die Leistungsfähigkeit des Programms ist meiner Meinung nach, daß man durch Umklappen des Vocablecraftscreens wieder auf den Workbenchscreen ge-

langt und dort anderen Dingen nachgehen kann. Diese Möglichkeit vermissen ich oft bei anderen (auch professionellen) Programmen. Die Multitaskingfähigkeiten werden nicht ausgenutzt. Des weiteren bietet Vocabcraft die Möglichkeit, nicht nur englische, sondern auch französische und spanische Vokabeln einzugeben, da durch die Funktionstasten-Belegung (F1-F10) ein erweiterter Zeichensatz entsteht. Deutsche Umlaute sind selbstverständlich. Aus technischen Gründen muß jedoch auf eine Abtrennung zweier Wörter durch Kommata verzichtet werden.

Ein π befindet sich unmittelbar neben der Taste {BACK SPACE}. Ein umgedrehtes Ausrufezeichen erscheint

Die Neubelegung der Tastatur ergibt sich wie folgt:

F1 = á F2 = é F3 = í F4 = ó F5 = ú
F6 = à F7 = è F8 = ì F9 = ò F10 = ù

nach einem Druck auf die Taste unter der ESC-Taste und ein ebenfalls umgedrehtes Fragezeichen wird dann erscheinen, wenn man vergeblich auf ein Komma hofft.

Die deutschen Zeichen ü/Û, ö/Ö und ä/Ä findet man dort, wo sie auch bei einer echten deutschen Tastatur gewohnt ist. Dadurch mußten natürlich die Zeichen ' und " ihre Position ändern. Sie sind nun auf den Tasten], / und } zu entdecken. Das ebenfalls im deutschen Zeichen-

satz vorhandene ß ist links neben dem Zeichen plaziert. Das Ypsilon befindet sich allerdings noch an der Stelle der amerikanischen Tastatur, da ich persönlich die amerikanische Tastatur nicht mit zu leicht verrutschenden Tastaturaufklebern vershandeln möchte. Doch jeder eingefleischte Computeruser wird genügend Basic-Kenntnisse besitzen, um diese Umgebung in dem Unterprogramm Wortholen selbst vorzunehmen.

Andreas Kuhnert

```

voc.bas.asc
1:      '          Vocabcraft V1.1          '
2:      '          Andreas Kuhnert        '
3:      '          '                        '
4:      '          KICKSTART 11/87        '
5:      '          '                        '
6:      HappyStart:
7:      SCREEN 1,640,200,1,2
8:      WINDOW 1,"          Vocabcraft Version 1.1 vo
          n Andreas Kuhnert",,0,1
9:      CLEAR ,90000&
10:     LOCATE 1,1
11:     start:
12:     CLS
13:     h=VAL(MID$(TIME$,1,2))
14:     d=VAL(MID$(TIME$,4,2))
15:     s=VAL(MID$(TIME$,7,2))
16:     Zeit=INT(TIMER)
17:     ON BREAK GOSUB knack
18:     BREAK ON
19:     ON ERROR GOTO werner
20:     DIM vo$(621):DIM me$(621):DIM st(621)
21:     MENU 1,0,1,"Disk Menu"
22:     MENU 1,1,1,"Alles löschen          "
23:     MENU 1,2,1,"Datei laden           "
24:     MENU 1,3,1,"Datei speichern       "
25:     MENU 1,4,1,"Datei mit Namen speichern "
26:     MENU 1,5,1,"Programmende         "
27:     MENU 1,6,1,"Datei löschen         "
28:     MENU 1,7,1,"Datei drucken         "
29:     MENU 2,0,1,"Datei Menu"
30:     MENU 2,1,1,"Datei erstellen, bzw. erweitern "
31:     MENU 2,2,1,"Datei ändern          "
32:     MENU 2,3,1,"Dateiteile löschen   "
33:     MENU 3,0,1,"Programm"
34:     MENU 3,1,1,"Test Vokabeln        "
35:     MENU 3,2,1,"Test Bedeutungen     "
36:     MENU 4,0,1,"Info"
37:     MENU 4,1,1,"Über Vocabcraft      "
38:     MENU 5,0,1,"Sprache"
39:     MENU 5,1,1," An          "
40:     MENU 5,2,2," Aus          "
41:     MENU 6,0,1,"Uhr"
42:     MENU 6,1,2," An          "
43:     MENU 6,2,1," Aus          "
44:     MENU 6,3,1,"Uhrzeit setzen      "
45:     MENU ON
46:     ON MENU GOSUB project
47:     UhrStatus=1:CLS
48:
49:     warten:
50:     LOCATE 1,1:MENU ON:otto=MOUSE(0):otto=MOUSE(3):otto=M
          OUSE(4):otto=0
51:     er$="Titel : "
52:     IF sie$="" THEN sie$="(kein Name)"
53:     LOCATE 1,1:PRINT er$+sie$
54:     MENU 1,0,1:MENU 2,0,1:MENU 3,0,1:MENU 4,0,1:MENU 5,0,
          1:MENU 6,0,1
55:     GOSUB uhr
56:     GOTO warten
57:
58:     werner:
59:     CLOSE
60:     IF fehlerstat<>0 THEN GOTO werner1
61:     LINE (112,50)-(500,140),0,bf
62:     LOCATE 10,18:PRINT "Quickdir nicht auf der aktuellen
          Diskette!"
63:     LOCATE 12,18:PRINT "          Na Und?          O
          K"
64:     LINE (128,64)-(480,104),1,b
65:     LINE (130,66)-(478,102),1,b
66:     LINE (381,84)-(419,99),1,b
67:     LINE (383,86)-(417,97),1,b
68:     LINE (162,84)-(242,98),1,b
69:     LINE (164,86)-(240,96),1,b
70:
71:     werner1:
72:     GOSUB uhr
73:     IF MOUSE(0)=0 THEN GOTO werner1
74:     IF MOUSE(3)>381 THEN IF MOUSE(4)>84 THEN IF MOUSE(3)
          <419 THEN IF MOUSE(4)<99 THEN GOTO werner2
75:     IF MOUSE(3)>162 THEN IF MOUSE(4)>84 THEN IF MOUSE(3)
          <242 THEN IF MOUSE(4)>98 THEN GOTO werner2.2
76:     IF MOUSE(3)>366 THEN IF MOUSE(4)>84 THEN IF MOUSE(
          3)<446 THEN IF MOUSE(4)>98 THEN GOTO werner2.2
77:
78:     GOTO werner1
79:     werner2:
80:     LINE (128,64)-(480,104),0,bf
81:     OPEN "df0:quickdir" FOR OUTPUT AS 1
82:     v=0:PRINT #1,v
83:     CLOSE:KILL "df0:quickdir.info"
84:     LINE (128,64)-(480,104),1,b
85:     LINE (130,66)-(478,102),1,b
86:     LINE (381,84)-(419,99),1,b
87:     LINE (383,86)-(417,97),1,b
88:     FOR i=1 TO 10:v=MOUSE(0):NEXT i
89:     LOCATE 10,24:PRINT "Quickdir ist nun installiert."
90:     LOCATE 12,50:PRINT "OK"
91:     werner2.2:
92:     LINE (128,64)-(480,104),0,bf
93:     RESUME warten
94:
95:     werner3:
96:     GOSUB uhr
97:     IF MOUSE(0)=0 THEN GOTO werner3
98:     IF MOUSE(3)>381 THEN IF MOUSE(4)>84 THEN IF MOUSE(3)
          <419 THEN IF MOUSE(4)<99 THEN GOTO werner4
99:     GOTO werner1

```

```

100:   werner4:
101:   LINE (128,64)-(480,104),0,bf
102:   RESUME warten
103:
104:   werner1:
105:   LINE (128,64)-(480,104),1,b
106:   LINE (130,66)-(478,102),1,b
107:   LINE (381,84)-(419,99),1,b
108:   LINE (383,86)-(417,97),1,b
109:   IF fehlerstat=1 THEN fehlerstat=0:LOCATE 10,18:PRINT
      "Keine TRANSLATOR-LIBRARY auf der Diskette!":talk=0:
      MENU 5,1,1:MENU 5,2,2
110:   IF fehlerstat=2 THEN fehlerstat=0:LOCATE 10,22:PRINT
      "Was ist denn mit dem Drucker los?"
111:   LOCATE 12,50:PRINT "OK"
112:   werner2:
113:   GOSUB uhr
114:   IF MOUSE(0)=0 THEN GOTO werner2
115:   IF MOUSE(3)>381 THEN IF MOUSE(4)>84 THEN IF MOUSE(4)
      <419 THEN IF MOUSE(4)<99 THEN GOTO werner3
116:   GOTO werner2
117:   werner3:
118:   LINE (128,64)-(480,104),0,bf
119:   CLS:RESUME warten
120:
121:   notfound:
122:   LINE (128,64)-(480,104),1,b
123:   LINE (130,66)-(478,102),1,b
124:   LINE (381,84)-(419,99),1,b
125:   LINE (383,86)-(417,97),1,b
126:   LOCATE 10,24:PRINT "Es ist keine Datei vorhanden!"
127:   LOCATE 12,50:PRINT "OK"
128:   notfound1:
129:   GOSUB uhr
130:   IF MOUSE(0)=0 THEN GOTO notfound1
131:   IF MOUSE(3)>381 THEN IF MOUSE(4)>84 THEN IF MOUSE(4)
      <419 THEN IF MOUSE(4)<99 THEN GOTO notfound2
132:   GOTO notfound1
133:   notfound2:
134:   LINE (128,64)-(480,104),0,bf
135:   RETURN
136:
137:   nany:
138:   LINE (162,84)-(242,98),1,b
139:   LINE (164,86)-(240,96),1,b
140:   LINE (366,84)-(446,98),1,b
141:   LINE (368,86)-(444,96),1,b
142:   LINE (128,64)-(480,104),1,b
143:   LINE (130,66)-(478,102),1,b
144:   LOCATE 10,24:PRINT nany$
145:   LOCATE 12,24:PRINT "NEIN!":LOCATE 12,51:PRINT "OK"
146:   nany1:
147:   GOSUB uhr:IF MOUSE(0)=0 THEN nany1
148:   IF MOUSE(3)>162 THEN IF MOUSE(4)>84 THEN IF MOUSE(4)
      <242 THEN IF MOUSE(4)<98 THEN GOTO nany2
149:   IF MOUSE(3)>366 THEN IF MOUSE(4)>84 THEN IF MOUSE(4)
      <446 THEN IF MOUSE(4)<98 THEN GOTO nany3
150:   GOTO nany1
151:   nany2:
152:   nany=0:CLS:RETURN
153:   nany3:
154:   nany=1:CLS:RETURN
155:
156:   knack:
157:   RETURN
158:   project:
159:   roll=1
160:   m2=MENU(1)
161:   m1=MENU(0)
162:   MENU 1,0,0:MENU 2,0,0:MENU 3,0,0:MENU 4,0,0:MENU 5,0,
      0:MENU 6,0,0
163:   IF m1=1 THEN GOTO Me1
164:   IF m1=2 THEN GOTO Me2
165:   IF m1=3 THEN GOTO Me3
166:   IF m1=4 THEN GOTO Me4
167:   IF m1=5 THEN GOTO Me5
168:   IF m1=6 THEN GOTO Me6
169:   Me1:
170:   IF m2=1 THEN GOTO neu
171:   IF m2=2 THEN GOTO offen
172:   IF m2=3 THEN GOTO Sichern
173:   IF m2=4 THEN GOTO sichern.wie
174:   IF m2=5 THEN nany$="Das Programm wirklich beenden?"
      :GOSUB nany:IF nany=1 THEN SYSTEM ELSE RETURN
175:   IF m2=6 THEN GOTO loeschen
176:   IF m2=7 THEN GOTO drucken
177:   Me2:
178:   IF m2=1 THEN GOTO N.Document
179:   IF m2=2 THEN GOTO C.Document
180:   IF m2=3 THEN GOTO D.Document
181:   Me3:
182:   IF m2=1 THEN test=0:GOTO Start.Test
183:   IF m2=2 THEN test=1:GOTO Start.Test
184:   Me4:
185:   IF m2=1 THEN GOTO About
186:   Me5:
187:   IF m2=1 THEN fehlerstat=1:talk=1:MENU 5,1,2:MENU 5,
      2,1:SAY TRANSLATE$("Speech on"):fehlerstat=0:RETUR
      N
188:   IF m2=2 THEN fehlerstat=1:talk=0:MENU 5,1,1:MENU 5,
      2,2:SAY TRANSLATE$("Speech off"):fehlerstat=0:RETU
      RN
189:   Me6:
190:   IF m2=1 THEN UhrStatus=1:MENU 6,1,2:MENU 6,2,1:roll
      =0:RETURN
191:   IF m2=2 THEN UhrStatus=0:MENU 6,1,1:MENU 6,2,2:roll
      =0:RETURN
192:   IF m2=3 THEN UhrStatus=1:MENU 6,1,2:MENU 6,2,1:GOTO
      UhrSet
193:
194:   neu:
195:   nany$=" Wirklich alle Daten löschen?"
196:   GOSUB nany
197:   IF nany=0 THEN CLS:RETURN
198:   CLS:CLEAR:GOTO start
199:
200:   offen:
201:   quicko=0
202:   LINE (145,50)-(500,140),1,b
203:   LINE (147,52)-(498,138),1,b
204:   LOCATE 9,54:PRINT "DF0:"
205:   LOCATE 13,54:PRINT "Laden"
206:   LOCATE 16,54:PRINT "Nein!"
207:   LINE (154,54)-(367,116),1,b
208:   LINE (367,54)-(382,116),1,b
209:   LINE (367,64)-(382,64),1
210:   LINE (367,106)-(382,106),1
211:   LINE (400,60)-(480,74),1,b
212:   LINE (402,62)-(478,72),1,b
213:   LINE (400,92)-(480,107),1,b
214:   LINE (402,94)-(478,105),1,b
215:   LINE (400,116)-(480,131),1,b
216:   LINE (402,118)-(478,129),1,b
217:   LINE (153,124)-(382,138),1,b
218:   RESTORE Pfeile
219:   FOR i=1 TO 17
220:   READ px1:READ py1:READ px2:READ py2
221:   LINE (px1,py1)-(px2,py2),1
222:   NEXT i
223:   LINE (155,55)-(366,115),0,bf
224:   OPEN "df0:quickdir" FOR INPUT AS #1
225:   INPUT #1,quickz
226:   DIM quick$(quickz+7)
227:   FOR i=1 TO quickz
228:   INPUT #1,quick$(i):NEXT i
229:   CLOSE:quicka=1
230:   klickoffen:
231:   FOR i=quicka TO quicka+6
232:   LOCATE 7+i,21:PRINT quick$(i):NEXT i:quicka=1
233:   klick1:
234:   GOSUB uhr
235:   IF MOUSE(0)=0 THEN GOTO klick1
236:   IF MOUSE(3)>367 THEN IF MOUSE(3)<382 THEN IF MOUSE(4)
      >54 THEN IF MOUSE(4)<64 THEN GOTO klickg1
237:   IF MOUSE(3)>367 THEN IF MOUSE(3)<382 THEN IF MOUSE(4)
      >106 THEN IF MOUSE(4)<116 THEN GOSUB klickg2
238:   IF MOUSE(3)>400 THEN IF MOUSE(3)<480 THEN IF MOUSE(4)
      >60 THEN IF MOUSE(4)<74 THEN ERASE quick$:GOTO off
      en
239:   IF MOUSE(3)>400 THEN IF MOUSE(3)<480 THEN IF MOUSE(4)
      >92 THEN IF MOUSE(4)<107 THEN GOTO klickg4
240:   IF MOUSE(3)>400 THEN IF MOUSE(3)<480 THEN IF MOUSE(4)
      >116 THEN IF MOUSE(4)<131 THEN GOTO klickg5
241:   IF MOUSE(3)>154 THEN IF MOUSE(3)<367 THEN IF MOUSE(4)
      >54 THEN IF MOUSE(4)<116 THEN GOSUB intext
242:   GOTO klick1
243:   klickg1:
244:   IF quickz<7 THEN RETURN
245:   IF quicka=1 THEN RETURN
246:   SCROLL (155,55)-(366,115),0,8
247:   LOCATE 8,21:PRINT quick$(quicka-1);
248:   quicka=quicka-1:RETURN
249:   klickg2:
250:   IF quickz<7 THEN RETURN
251:   IF quicka=quickz-6 THEN RETURN
252:   SCROLL (155,55)-(366,115),0,-8

```

```

253: LOCATE 14,21:PRINT quick$(quicka+7);
254: quicka=quicka+1:RETURN
255: klickg4:
256: IF quick$(quicko)="" THEN GOTO klick1
257: LINE (145,50)-(500,140),0,bf
258: GOTO klickg3
259: klickg5:
260: LINE (145,50)-(500,140),0,bf:ERASE quick$:CLS:RETUR
N
261: intext:
262: quicko=INT((MOUSE(4)-56)/8)+quicka
263: IF quicko>quickz THEN quicko=quickz
264: LOCATE 17,21
265: LINE (154,125)-(381,137),0,bf
266: PRINT quick$(quicko)
267: RETURN
268: klickg3:
269: FOR i=1 TO z
270: vo$(i)="" :me$(i)=""
271: NEXT i
272: offen$="df0:"+quick$(quicko)
273: OPEN offen$ FOR INPUT AS #1
274: INPUT #1,z
275: FOR i=1 TO z:INPUT #1,vo$(i):INPUT #1,me$(i):GOSUB uh
r:NEXT
276: sie$=quick$(quicko):ERASE quick$
277: CLOSE:CLS:PRINT er$+sie$
278: GOTO C.Document
279:
280: loeschen:
281: quicko=0
282: LINE (145,50)-(500,140),1,b
283: LINE (147,52)-(498,138),1,b
284: LOCATE 9,54:PRINT "DF0:"
285: LOCATE 13,53:PRINT "Löschen"
286: LOCATE 16,54:PRINT "Nein!"
287: LINE (154,54)-(367,116),1,b
288: LINE (367,54)-(382,116),1,b
289: LINE (367,64)-(382,64),1
290: LINE (367,106)-(382,106),1
291: LINE (400,60)-(480,74),1,b
292: LINE (402,62)-(478,72),1,b
293: LINE (400,92)-(480,107),1,b
294: LINE (402,94)-(478,105),1,b
295: LINE (400,116)-(480,131),1,b
296: LINE (402,118)-(478,129),1,b
297: LINE (153,124)-(382,138),1,b
298: RESTORE Pfeile
299: FOR i=1 TO 17
300: READ px1:READ py1:READ px2:READ py2
301: LINE (px1,py1)-(px2,py2),1
302: NEXT i
303: LINE (155,55)-(366,115),0,bf
304: OPEN "df0:.quickdir" FOR INPUT AS #1
305: INPUT #1,quickz
306: DIM quick$(quickz+7)
307: FOR i=1 TO quickz
308: INPUT #1,quick$(i):NEXT i
309: CLOSE:quicka=1
310: klickoffenl:
311: FOR i=quicka TO quicka+6
312: LOCATE 7+i,21:PRINT quick$(i):NEXT i:quicka=1
313: klick11:
314: GOSUB uhr
315: IF MOUSE(0)=0 THEN GOTO klick11
316: IF MOUSE(3)>367 THEN IF MOUSE(3)<382 THEN IF MOUSE(4
)>54 THEN IF MOUSE(4)<64 THEN GOSUB klickg11
317: IF MOUSE(3)>367 THEN IF MOUSE(3)<382 THEN IF MOUSE(4
)>106 THEN IF MOUSE(4)<116 THEN GOSUB klickg21
318: IF MOUSE(3)>400 THEN IF MOUSE(3)<480 THEN IF MOUSE(4
)>60 THEN IF MOUSE(4)<74 THEN ERASE quick$:GOTO loe
schen
319: IF MOUSE(3)>400 THEN IF MOUSE(3)<480 THEN IF MOUSE(4
)>92 THEN IF MOUSE(4)<107 THEN GOTO klickg41
320: IF MOUSE(3)>400 THEN IF MOUSE(3)<480 THEN IF MOUSE(4
)>116 THEN IF MOUSE(4)<131 THEN GOTO klickg51
321: IF MOUSE(3)>154 THEN IF MOUSE(3)<367 THEN IF MOUSE(4
)>54 THEN IF MOUSE(4)<116 THEN GOSUB intext1
322: GOTO klick11
323: klickg11:
324: IF quickz<7 THEN RETURN
325: IF quicka=1 THEN RETURN
326: SCROLL (155,55)-(366,115),0,8
327: LOCATE 8,21:PRINT quick$(quicka-1);
328: quicka=quicka-1:RETURN
329: klickg21:
330: IF quickz<7 THEN RETURN
331: IF quicka=quickz-6 THEN RETURN
332: SCROLL (155,55)-(366,115),0,-8
333: LOCATE 14,21:PRINT quick$(quicka+7);
334: quicka=quicka+1:RETURN
335: klickg41:
336: IF quick$(quicko)="" THEN GOTO klick11
337: LINE (145,50)-(500,140),0,bf
338: GOTO klickg31
339: klickg51:
340: LINE (145,50)-(500,140),0,bf:ERASE quick$:RETURN
341: intext1:
342: quicko=INT((MOUSE(4)-56)/8)+quicka
343: IF quicko>quickz THEN quicko=quickz
344: LOCATE 17,21
345: LINE (154,125)-(381,137),0,bf
346: PRINT quick$(quicko)
347: RETURN
348: klickg31:
349: loesch$="df0:"+quick$(quicko)
350: KILL loesch$
351: FOR i=quicko TO quickz:quick$(i)=quick$(i+1):NEXT i:q
uickz=quickz-1
352: OPEN "df0:.quickdir" FOR OUTPUT AS #1
353: PRINT #1,quickz
354: FOR i=1 TO quickz:PRINT #1,quick$(i):NEXT i
355: CLOSE:KILL "df0:.quickdir.info":ERASE quick$:CLS:RETU
RN
356:
357:
358:
359: Sichern:
360: IF sie$="(kein Name)" THEN GOTO sichern.wie
361: OPEN "df0:.quickdir" FOR INPUT AS #1
362: INPUT #1,quickz
363: DIM quick$(quickz+2)
364: FOR i=1 TO quickz
365: INPUT #1,quick$(i)
366: NEXT i
367: CLOSE
368: FOR i=1 TO quickz
369: IF quick$(i)=sie$ THEN statsich=1
370: NEXT i
371: IF statsich=1 THEN GOTO sich1
372: quickz=quickz+1
373: quick$(quickz)=sie$
374: OPEN "df0:.quickdir" FOR OUTPUT AS #1
375: PRINT #1,quickz
376: FOR i=1 TO quickz
377: PRINT #1,quick$(i)
378: NEXT i:CLOSE:KILL "df0:.quickdir.info"
379: sich1:
380: statsich=0
381: OPEN "df0:"+sie$ FOR OUTPUT AS #1
382: PRINT #1,z
383: FOR i=1 TO z:PRINT #1,vo$(i):PRINT #1,me$(i):GOSUB uh
r:NEXT
384: CLOSE:ERASE quick$
385: KILL "df0:"+sie$+".info"
386: CLS:RETURN
387:
388: sichern.wie:
389: IF z=0 THEN GOSUB notfound:RETURN
390: LOCATE 12,18:PRINT "Sichern Nein!"
391: LINE (122,84)-(202,98),1,b
392: LINE (124,86)-(200,96),1,b
393: LINE (256,84)-(336,98),1,b
394: LINE (258,86)-(334,96),1,b
395: LINE (122,69)-(336,81),1,b
396: LINE (112,61)-(346,106),1,b
397: LINE (114,63)-(344,104),1,b
398: OPEN "df0:.quickdir" FOR INPUT AS #1
399: INPUT #1,quickz
400: DIM quick$(quickz+2)
401: FOR i=1 TO quickz
402: INPUT #1,quick$(i)
403: NEXT i
404: CLOSE
405: sichw:
406: LINE (123,70)-(335,80),0,bf
407: lx=17:ly=10:cor=18:be$="" :GOSUB wortholen
408: sichw1:
409: GOSUB uhr:IF MOUSE(0)=0 THEN sichw1
410: IF MOUSE(3)>122 THEN IF MOUSE(4)>84 THEN IF MOUSE(3
)<202 THEN IF MOUSE(4)<98 THEN GOTO sichw2
411: IF MOUSE(3)>256 THEN IF MOUSE(4)>84 THEN IF MOUSE(3
)<336 THEN IF MOUSE(4)<98 THEN GOTO sichw3
412: IF MOUSE(3)>122 THEN IF MOUSE(4)>69 THEN IF MOUSE(3
)<336 THEN IF MOUSE(4)<81 THEN GOTO sichw
413: GOTO sichw1
414: sichw2:

```

AMIGA BUCHHITS



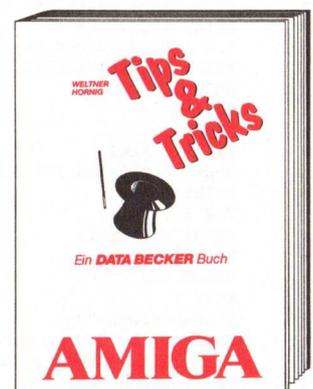
Was leisten die neuen Amigas? Hier finden Sie die Antwort. Unabhängig davon, ob Sie den Amiga schon haben oder den Kauf planen. Dieses Buch bietet Ihnen Entscheidungshilfen, technische Details und jede Menge von dem, was man mit Amiga 500 & 2000 so alles anstellen kann. Eben Informationen, die man braucht, wenn man sich für die neuen Amigas interessiert. Aufbereitet nach einem völlig neuartigen didaktischen Konzept, in einer Sprache, die zum Amiga paßt.
Das können Amiga 500 & 2000
 190 Seiten, DM 29,-



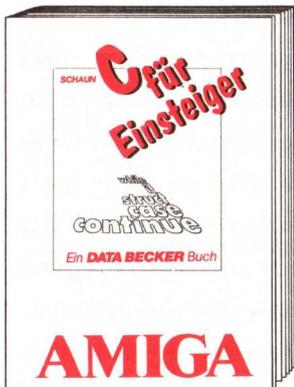
Wählen Sie gleich den richtigen Einstieg zu Ihrem Amiga 500. Denn das Handbuch läßt Sie dabei völlig allein. Versuchen Sie es lieber gleich mit Amiga 500 für Einsteiger. Hier heißt es: Anschließen und loslegen. Verständlich für jedermann zeigt Ihnen dieses Buch: Workbench, AmigaBASIC, CLI und AmigaDOS. Locker aufbereitet bietet es Ihnen alles Wissenswerte. Bis hin zu den beim Amiga 500 mitgelieferten Zusatzprogrammen.
Amiga 500 für Einsteiger
 343 Seiten, DM 39,-



Das erfolgreiche Buch zu Amiga-BASIC – jetzt in Neuauflage! Erweitert um Kickstart 1.2, neuer Workbench und Amiga 500 & 2000. Mit allem, was BASIC-Programmierern Spaß macht: Grafik und Sound, Laden und Speichern von Graficraft-Bildern in BASIC-Programme, sequentielle und relative Dateien, Business-Grafik, Computeranimation, Windows, Umgang mit IFF-Bildern, Sprachausgabe und, und, und. Das Buch für Einsteiger, Aufsteiger und Profis.
AmigaBASIC
 Hardcover, 774 Seiten, DM 59,-



Amiga Tips & Tricks. Ein Buch, das voller Überraschungen steckt: 64 Farben gleichzeitig auf dem Amiga. Von BASIC aus Zugriff auf die Libraries. Benutzung verschiedener Zeichensätze in BASIC. Sinnvoller Einsatz von Windows, Screens und Menüs. Tips zu einzelnen Grafikbefehlen, Programm- und AmigaDOS-Routinen! Greifen Sie in die Trickkiste, und schon sind Dinge möglich, die man gar nicht gedacht hätte.
Amiga Tips & Tricks
 Hardcover, 364 Seiten, DM 49,-



C an einem Wochenende? Durchaus möglich! Mit C für Einsteiger. Ein Einführungskurs, der Ihnen schnell und einfach die wichtigsten Grundlagen dieser Sprache vermittelt. Vom ersten Programm bis hin zu den Routinen in den Bibliotheken. Mit dem gesamten Sprachumfang und den besonderen Features von C. Zahlreiche Tips & Tricks zur Programmierung und eine Beschreibung der beiden Compiler Lattice C und Aztek runden das Ganze ab.
Amiga C für Einsteiger
 254 Seiten, DM 39,-



Schreiben Sie Ihre Programme in Maschinensprache – und Sie werden sehen, wie schnell ein Amiga sein kann. Das nötige Know-how liefert Ihnen dieses Buch: Grundlagen des 68000, das Amiga-Betriebssystem, Druckeransteuerung, Diskettenoperationen, Sprachausgabe, Windows, Screens, Register, Pull-Down-Menüs ... Und damit Sie auch gleich praktisch arbeiten können, werden die wichtigsten Assembler vorgestellt.
Amiga Maschinensprache
 Hardcover, 282 Seiten
 DM 49,-



Hier die unverzichtbare Hilfe für jeden Adventure-Freak. 20 der besten, aktuellen Infocom-Adventures, weitere Top-Adventures sowie die besten Rollenspiele lassen sich nun problemlos lösen. Aber Sie können diesen DATA BECKER Führer auch ganz gezielt befragen: Nach bestimmten Lösungen oder nach entsprechenden Plänen. Der Spielspaß bleibt dabei immer garantiert.
Der DATA BECKER Führer Adventures & Rollenspiele
 196 Seiten, DM 19,80

Super

Regelmäßig in der DATA WELT: Amiga Window – das Forum für den engagierten Amiga-Anwender. Mit kreativen Projekten, Interviews, Software-Tests und wichtigen News. Und: Auch was sich sonst so in der Computerszene tut, erfährt der Amiga-Anwender. Die DATA WELT – das aktuelle Computermagazin. Monat für Monat überall da, wo es Zeitschriften gibt.

DATA BECKER
 Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

BESTELL-COUPON
 Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
 Bitte senden Sie mir:
 zzgl. DM 5,- Versandkosten
 unabhängig von der bestellten Stückzahl
 per Nachnahme Verrechnungsscheck liegt bei
 Name _____
 Straße _____
 Ort _____

```

415: IF be$="" THEN GOTO sichw
416: LINE (112,61)-(346,106),0,bf
417: FOR i=1 TO quickz
418: IF quick$(i)=be$ THEN quicksich=i
419: NEXT i
420: IF quicksich=0 THEN quickz=quickz+1:quicksich=quickz

421: quick$(quicksich)=be$
422: OPEN "df0:.quickdir" FOR OUTPUT AS #1
423: PRINT #1,quickz
424: FOR i=1 TO quickz
425: PRINT #1,quick$(i)
426: GOSUB uhr
427: NEXT i
428: CLOSE
429: ERASE quick$
430: OPEN "df0:"+be$ FOR OUTPUT AS #1
431: PRINT #1,z
432: FOR i=1 TO z
433: PRINT #1,vo$(i)
434: PRINT #1,me$(i)
435: GOSUB uhr
436: NEXT i
437: sie$=be$:be$=""
438: CLOSE:KILL "df0:.quickdir.info"
439: KILL "df0:"+sie$+".info"
440: quicksich=0:CLS
441: RETURN
442: sichw3:
443: LINE (112,61)-(346,106),0,bf:ERASE quick$:CLS:RETURN

444: drucken:
445: IF z=0 THEN GOSUB notfound:RETURN
446: fehlerstat=2
447: OPEN "LPT1:" FOR OUTPUT AS #1
448: PRINT #1," Vocabcraft V1.1 "
449: PRINT #1,er$+sie$;" " ;DATES;" " ;h$;" " ;d$;" " ;
s$
450: PRINT #1,"-----"
451: FOR i = 1 TO z
452: drle=LEN(vo$(i))
453: dru$=dru$+vo$(i):FOR dr=1 TO 40-drle:dru$=dru$+" ":NE
XT dr
454: dru$=dru$+me$(i)
455: PRINT #1,dru$
456: dru$=""
457: NEXT i:fehlerstat=0:CLS:CLOSE:RETURN
458:
459:
460: N.Document:
461: CLS:PRINT er$+sie$:LOCATE 22,1:PRINT "Zum verlassen'E
SC' dr";CHR$(252);"cken"
462: z=z+1:IF z>600 THEN z=z-1:GOTO C.Document
463: LOCATE 5,30:PRINT "Vokabel ":"ly=5:lx=40:cor=40
464: IF ret=1 THEN ret=0:z=z-1:CLS:PRINT er$+sie$:GOTO C.D
ocoment
465: GOSUB wortholen
466: vo$(z)=be$
467: LOCATE 7,30:PRINT "Bedeutung ":"ly=7:lx=42:cor=42
468: IF ret=1 THEN ret=0:z=z-1:CLS:PRINT er$+sie$:GOTO C.D
ocoment
469: GOSUB wortholen
470: me$(z)=be$
471: GOTO N.Document
472:
473: C.Document:
474: IF z=0 THEN GOSUB notfound:RETURN
475: m=1
476: cd1:
477: IF plu=-1 THEN IF m>9 THEN m=m-20:plu=0:GOTO cd1.1
478: IF plu=1 THEN IF m<z THEN m=m+20:plu=0:GOTO cd1.1
479: cd1.1:
480: CLS:PRINT er$+sie$:FOR i=m TO m+19:PRINT i;".TAB(8
) vo$(i) TAB(40) me$(i):NEXT
481: PRINT "'Help' zum editieren 'ESC' zum Beenden (
+/-) Seite vor/zurueck"
482: cd2:
483: a$=INKEY$:GOSUB uhr
484: IF a$="" THEN cd2
485: IF ASC(a$)=27 THEN CLS:PRINT er$+sie$:RETURN
486: IF ASC(a$)=139 THEN cd3
487: IF a$="-" THEN plu=-1:GOTO cd1
488: IF a$="=" THEN plu=1:GOTO cd1
489: GOTO cd2
490: cd3:
491: LOCATE 22,1:FOR i=1 TO 80:PRINT " ";:NEXT
492: LOCATE 22,1:PRINT "Vokabelnummer ":"ly=22:lx=17:cor
=17:GOSUB wortholen:cdc=VAL(be$):IF cdc>z OR cdc=0
THEN BEEP:GOTO cd1
493: LOCATE 22,1:FOR i=1 TO 80:PRINT " ";:NEXT
494: LOCATE 22,1:PRINT "Neue Vokabel ":"ly=22:lx=18:co
r=18:GOSUB wortholen
495: IF be$="" THEN GOTO cd3.1
496: vo$(cdc)=be$
497: cd3.1: LOCATE 22,1:FOR i=1 TO 60:PRINT " ";:NEXT
498: LOCATE 22,1:PRINT "Neue Bedeutung ":"ly=22:lx
=18:cor=18:GOSUB wortholen
499: IF be$="" THEN cd1
500: me$(cdc)=be$:GOTO cd1
501:
502: D.Document:
503: IF z=0 THEN GOSUB notfound:RETURN
504: m=1
505: dd1:
506: IF plusminus=-1 THEN IF m>9 THEN m=m-20:plusminus=0:
GOTO dd1.1
507: IF plusminus=1 THEN IF m<z THEN m=m+20:plusminus=0:G
OTO dd1.1
508: dd1.1:
509: CLS:PRINT er$+sie$:FOR i=m TO m+19:PRINT i;".TAB(8
)vo$(i)TAB(40)me$(i):NEXT
510: PRINT "'Help' zum l";CHR$(246);"schen 'ESC' zum B
eenden (+/-) Seite vor/zurueck"
511: dd2:
512: a$=INKEY$:GOSUB uhr
513: IF a$="" THEN dd2
514: IF ASC(a$)=27 THEN CLS:PRINT er$+sie$:RETURN
515: IF ASC(a$)=139 THEN dd3
516: IF a$="-" THEN plusminus=-1:GOTO dd1
517: IF a$="=" THEN plusminus=1:GOTO dd1
518: GOTO dd2
519: dd3:
520: LOCATE 22,1:FOR i=1 TO 80:PRINT " ";:NEXT
521: LOCATE 22,1:PRINT "Nummer der zu l";CHR$(246);"sche
nden Vokabel ":"ly=22:lx=36:cor=36:GOSUB wortholen
:ddc=VAL(be$):IF ddc>z OR ddc=0 THEN BEEP:GOTO dd1
522: FOR i=ddc TO z:vo$(i)=vo$(i+1):me$(i)=me$(i+1):NEXT
i:z=z-1:IF z=0 THEN sie$="" :CLS:GOTO D.Document
GOTO dd1
523:
524:
525: Start.Test:
526: IF z=0 THEN GOSUB notfound:RETURN
527: FOR i=1 TO z:st(i)=0:NEXT
528: st1:
529: IF zst>z THEN zst=0:GOTO st3
530: pq=INT(RND*z)+1:IF st(pq)=-1 THEN GOTO st1
531: CLS:PRINT er$+sie$
532: IF test=0 THEN LOCATE 7,30:PRINT "Bedeutung : ";me$(
pq)
533: IF test=1 THEN LOCATE 7,30:PRINT "Bedeutung ":"ly=7
:lx=42:cor=42
534: LOCATE 21,1:PRINT "'ESC' dr";CHR$(252);"cken zum Be
enden des Tests"
535: IF test=0 THEN LOCATE 5,30:PRINT "Vokabel ":"ly=5:1
x=40:cor=40
536: IF test=1 THEN LOCATE 5,30:PRINT "Vokabel ":" ;vo$(p
q):IF talk=1 THEN SAY TRANSLATE$(vo$(pq))
537: GOSUB wortholen:zxx=zxx+1:IF ret=1 THEN ret=0:CLS:P
RINT er$+sie$:RETURN
538: IF test=0 THEN IF be$<>vo$(pq) THEN st(pq)=st(pq)+1:
qwe=qwe+1:GOTO st2
539: IF test=1 THEN IF be$<>me$(pq) THEN st(pq)=st(pq)+1:
qwe=qwe+1:GOTO st2
540: GOTO sta
541: st2:
542: IF test=0 THEN LOCATE 9,15:PRINT st(pq);". Fehler
bei dieser Vokabel."
543: IF test=1 THEN LOCATE 9,15:PRINT st(pq);". Fehler
bei dieser Bedeutung."
544: IF test=0 THEN LOCATE 10,15:PRINT "Die korrekte V
okabel ist ";vo$(pq);"" :IF talk=1 THEN SAY TRA
NSLATE$("The correct answer is :"+vo$(pq)):GOTO
Struggle
545: IF test=1 THEN LOCATE 10,15:PRINT "Die korrekte B
edeutung ist ";me$(pq);""
FOR i=1 TO 100:GOSUB uhr:NEXT
546:
547: Struggle:
548: IF st(pq)>2 THEN ne=ne+1:st(pq)=-1:zst=zst+1:CLS:
PRINT "Moment bitte ...":GOTO st1
549: CLS:PRINT "Moment bitte ...":GOTO st1
550: sta:
551: IF st(pq)>0 THEN st(pq)=-1:zst=zst+1:ne=ne+1:GOTO
st1
552: IF st(pq)=0 THEN ri=ri+1:zst=zst+1:st(pq)=-1:GOTO
st1
553: st3:

```

AMIGA INTERN.

Die Intern-Bücher von DATA BECKER haben für den Computer-Profi schon seit jeher einen ganz besonderen Stellenwert. So kann man beispielsweise C64 Intern selbst in den Reparaturwerkstätten von COMMODORE finden. Ein weiterer, überaus erfolgreicher Intern-Band - ATARI Intern - dient verschiedenen, amerikanischen Softwarehäusern bei der Entwicklung ihrer Software als unverzichtbare Informationsquelle. Fakten, die deutlich machen, was Sie von AMIGA Intern erwarten können. Eben ein Buch vollgepackt mit Informationen, die nur in einem Intern-Buch zu finden sind. AMIGA Intern hilft Ihnen, Ihren Superrechner von Grund auf zu verstehen. Hier werfen Sie quasi einen Blick auf die Innereien Ihres AMIGAs. Unter Anleitung von absoluten Experten versteht sich. Wie sehr Sie hier bis ins kleinste Detail vordringen, verrät bereits ein kurzer Blick ins Inhaltsverzeichnis: Die Hardware des AMIGA mit ausführlicher Beschreibung des 68000-Prozessors, der CIAS, der Co-Stumchips und der Schnittstellen, die Programmierung der Hardware in Assembler mit Speicherbelegung, Interrupts, Grafik und Sound. Doch AMIGA Intern würde diesen Namen nicht verdienen, wenn es zusätzlich nicht noch jede Menge Fakten und Details zum Betriebssystem liefern würde. So erfahren Sie alles Wissenswerte zur EXEC: die wichtigsten Strukturen, Funktion und Arbeitsweise des Multitasking, I/O-Handhabung und Verwaltung der Ressourcen, EXEC-Base, resetfeste Programme und Strukturen. Fehlt nur noch das AmigaDOS. Auch hierzu nur einige Stichworte: Funktionen, Parameterübergabe, Fehlermeldungen, Bootvorgang, Aufbau



der Diskette, IFF-Format, Programmstart von CLI und Workbench, interner Aufbau der CLI-Befehle, die Devices. Selbstverständlich finden Sie auch zahlreiche Beispiele zur Programmierung der EXEC- und DOS-Routinen. Am Ende wissen Sie, warum der AMIGA so enorm leistungsfähig ist - und wir sind gespannt, in welcher Entwicklungsabteilung wir dieses Intern-Buch wiederfinden werden.

AMIGA Intern
Hardcover, 639 Seiten, DM 69,—

DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

BESTELL-COUPON
Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:

Per Nachnahme Verrechnungsscheck (liegt bei)

Name _____
Straße _____
Ort _____

```

554: CLS:PRINT er$+sie$
555: IF ri=z THEN GOTO st4
556: ste:
557: LOCATE 14,3:PRINT "Es waren ";100-INT((100/z)*r
i);"% der Antworten falsch"
558: LOCATE 21,1:PRINT " Taste dr";CHR$(252);"cken"
559: a$=INKEY$:GOSUB uhr
560: IF a$=""THEN GOTO ste
561: ri=0:ne=0:zst=0:zzx=0:qwe=0:CLS:PRINT er$+sie$:
RETURN
562: st4:
563: LOCATE 10,3:PRINT "Herzlichen Gl";CHR$(252);"ckwu
nsch !!"
564: LOCATE 11,3:PRINT "Es sind alle Vokabeln richtig
gewesen."
565: LOCATE 12,3:PRINT "Insgesamt waren ";z;" Vokabeln
gefragt."
566: GOTO ste
567:
568: About:
569: LINE (160,60)-(350,160),1,b
570: LINE (162,62)-(348,158),1,b
571: LOCATE 10,23:PRINT " Vocablcrafft Vl.1"
572: LOCATE 11,22:PRINT " von Andreas Kuhnert"
573: LOCATE 13,22:PRINT " Programmirt im"
574: LOCATE 14,22:PRINT " Zeitraum vom"
575: LOCATE 15,22:PRINT " 7.11.1986 - 1.5.1987"
576: LOCATE 17,22:PRINT " für KICKSTART 11/87"
577: LOCATE 18,22:PRINT ""
578: LOCATE 19,22:PRINT " OK"
579: LINE (293,140)-(330,154),1,b
580: LINE (295,142)-(328,152),1,b
581: a1:
582: IF MOUSE(0)=0 THEN a1
583: IF MOUSE(3)>293 THEN IF MOUSE(3)<330 THEN IF MOUSE(4
)>140 THEN IF MOUSE(4)<154 THEN a2
584: GOTO a1
585: a2:
586: LINE (160,60)-(350,160),0,bf
587: RETURN
588:
589: wortholen:
590: be$="":cor=lx
591: ger:
592: LOCATE ly,lx:PRINT "_":LOCATE ly,lx
593: Wh1:
594: v=MOUSE(0):a$=INKEY$:GOSUB uhr:IF a$="" THEN Wh1
595: IF ASC(a$)=8 THEN Del
596: IF ASC(a$)=13 THEN PRINT " ":RETURN
597: IF ASC(a$)=27 THEN ret=1:RETURN
598: IF LEN(be$)=25 THEN BEEP:GOTO Wh1
599: IF ASC(a$)=129 THEN a$=CHR$(225):GOTO qwe
600: IF ASC(a$)=130 THEN a$=CHR$(233):GOTO qwe
601: IF ASC(a$)=131 THEN a$=CHR$(237):GOTO qwe
602: IF ASC(a$)=132 THEN a$=CHR$(243):GOTO qwe
603: IF ASC(a$)=133 THEN a$=CHR$(218):GOTO qwe
604: IF ASC(a$)=34 THEN a$=CHR$(196):GOTO qwe
605: IF a$="\ "THEN a$=CHR$(241):GOTO qwe
606: IF a$="["THEN a$=CHR$(252):GOTO qwe
607: IF a$="{ "THEN a$=CHR$(220):GOTO qwe
608: IF a$=";"THEN a$=CHR$(246):GOTO qwe
609: IF a$=":"THEN a$=CHR$(214):GOTO qwe
610: IF a$="'"THEN a$=CHR$(228):GOTO qwe
611: IF a$="="THEN a$=CHR$(223):GOTO qwe
612: IF a$="`"THEN a$=CHR$(161):GOTO qwe
613: IF a$=","THEN a$=CHR$(191):GOTO qwe
614: IF a$="]"THEN a$="" :GOTO qwe
615: IF a$="}"THEN a$=CHR$(34):GOTO qwe
616: IF ASC(a$)=134 THEN a$=CHR$(224):GOTO qwe
617: IF ASC(a$)=135 THEN a$=CHR$(232):GOTO qwe
618: IF ASC(a$)=136 THEN a$=CHR$(236):GOTO qwe
619: IF ASC(a$)=137 THEN a$=CHR$(242):GOTO qwe
620: IF ASC(a$)=138 THEN a$=CHR$(217):GOTO qwe
621: qwe:
622: lx=lx+1:PRINT a$:be$=be$+a$:a$="" :GOTO ger
623: Del:
624: IF lx<cor THEN ger
625: PRINT " ":lx=lx-1:LOCATE ly,lx:PRINT " ":be$=LEFT
$(be$,LEN(be$)-1):GOTO ger
626:
627: uhr:
628: v=MOUSE(0)
629: uhrx=POS(0):uhr=CSRLIN
630: Uhr1=INT(TIMER)
631: IF Uhr1=Zeit THEN RETURN
632: Zeit=Zeit+(Uhr1-Zeit):s=s+(Uhr1-Zeit+1)
633: IF s>59 THEN s=0:d=d+1
634: IF d>59 THEN d=0:h=h+1
635: IF h>23 THEN h=0
636: IF s<10 THEN s$="0"+RIGHT$(STR$(s),1)
637: IF d<10 THEN d$="0"+RIGHT$(STR$(d),1)
638: IF h<10 THEN h$="0"+RIGHT$(STR$(h),1)
639: IF s>9 THEN s$=RIGHT$(STR$(s),2)
640: IF d>9 THEN d$=RIGHT$(STR$(d),2)
641: IF h>9 THEN h$=RIGHT$(STR$(h),2)
642: IF UhrStatus=1 THEN LOCATE 1,70:PRINT h$;"":d$;"":s
$;
643: IF UhrStatus=0 THEN LOCATE 1,70:PRINT "Uhr aus!";
644: LOCATE 1,50:PRINT "Free ";FRE(1);:LOCATE uhr,uhrx
645: RETURN
646:
647: UhrSet:
648: CLS:PRINT er$+sie$
649: LOCATE 10,1:PRINT "Stunden ":ly=10:lx=9:cor=9:GOSUB w
ortholen:h=VAL(be$)
650: LOCATE 11,1:PRINT "Minuten ":ly=11:lx=9:cor=9:GOSUB w
ortholen:d=VAL(be$)
651: s=0:Zeit=INT(TIMER):Uhr1=INT(TIMER):CLS:RETURN
652:
653: Pfeile:
654: DATA 373,108,373,112
655: DATA 376,108,376,112
656: DATA 373,108,376,108
657: DATA 370,110,374,114
658: DATA 375,114,379,110
659: DATA 370,110,379,110
660: DATA 374,109,374,113
661: DATA 375,109,375,113
662: DATA 372,111,377,111
663:
664: DATA 373,062,373,057
665: DATA 376,057,376,062
666: DATA 379,060,370,060
667: DATA 371,059,374,056
668: DATA 375,056,378,059
669: DATA 374,062,374,057
670: DATA 375,057,375,062
671: DATA 372,059,377,059

```

Public Domain Service je Disk 8,00 DM

Markendisk, ab 5 Stck. portofrei, Inhalt siehe diese Ausgabe.
ab 10 Stck. 7,00 DM - ab 20 Stck. 6,00 DM

Weitere Public Domain. Derzeit 119 Disks aus 25 Bereichen.
Katalog auf Diskette DM 5,-

HARDWARE + USERSOFTWARE
finden Sie in unserer Preisliste

DAST Computer + Software

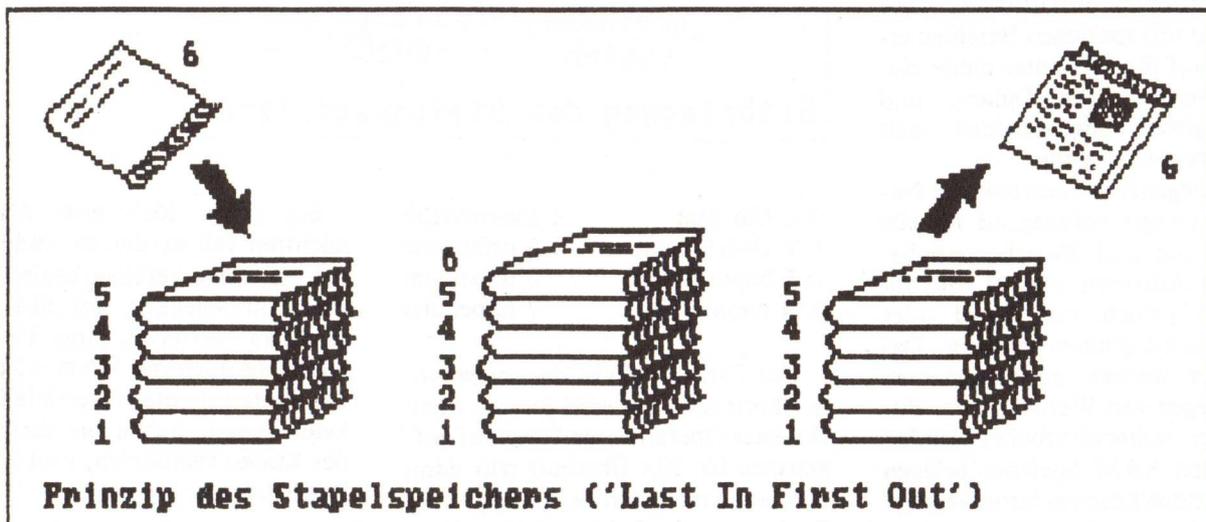
Ronald Danne + Ingo Streitner GbR
Ulmenweg 4a · 4057 Brüggen 1

Auslandslieferung nur gegen Vorkasse. Telefonische Bestellung unter **0 21 57/30 53**

Mit dem Rechner auf

DU und *DU*

Ein Assemblerkurs für Einsteiger



Assembler, ein Wort, das viele in Panik versetzt und an sich selbst zweifeln läßt. Jene suchen dann Zuflucht in Hochsprachen wie C, Pascal oder Basic. Dieser Kurs soll Hemmungen überwinden helfen und zeigen, daß sich jeder Assemblerkenntnisse aneignen kann, wenn er nicht gleich bei der ersten Guru Meditation aufgibt. Er soll dem Anfänger die ersten Schritte in Maschinensprache erleichtern, und Hilfestellungen für die Praxis geben. Bevor wir uns frischen Mutes in die Programmierung unseres Prozessors stürzen, sei hier noch kurz erwähnt, daß wir für die Beispielprogramme den Seka-Assembler von KUMA verwenden werden; einen Assembler, der sich durch seinen niedrigen Preis und seine Vielfältigkeit ohne Nachladen auszeichnet.

Die einzigen Voraussetzungen, die Sie für diesen Kurs benötigen, sind Kenntnisse in der computerinternen Zahlenverarbeitung, das heißt, Sie

sollten wissen, was Bits und Bytes sind und wie diese gehandhabt werden. Wir arbeiten mit dem Hexadezimalsystem, das – anders als das Dezimalsystem, mit dem wir normalerweise rechnen – die Zahl 16 als Basis hat. Das System kennt die Zeichen $\phi\phi-\phi F$, was bedeutet, daß zum Beispiel die dezimale Zahl 10 hexadezimal als ϕA dargestellt wird. Das Stringzeichen (\$) weist den Assembler auf die Hexadezimalschreibweise hin. Rechnet man zum Beispiel im Binärsystem mit Zweierpotenzen, so rechnet man im Hexadezimalsystem mit den Potenzen von 16. So ergibt die hexadezimale Zahl $\phi E3AC$ umgerechnet in das Dezimalsystem den Wert $58284 (14 \cdot 16^3 + 3 \cdot 16^2 + 10 \cdot 16^1 + 12 \cdot 16^0)$. Andersherum ist es schwieriger:

$$58284 : 16^3 = 14 \text{ (Ganzzahlwert)}$$

$$58284 - (16^3 \cdot 14) = 940$$

$$940 : 16^2 = 3 \text{ (Ganzzahlwert)}$$

$$940 - (16^2 \cdot 3) = 172$$

$$172 : 16^1 = 10 \text{ (Ganzzahlwert)}$$

$$172 - (16^1 \cdot 10) = 12$$

$14 = \phi E$; $10 = \phi A$; $12 = \phi C$, daraus folgt ... $\phi E3AC$

Man geht, will man 58284 in $\phi E3AC$ umrechnen, folgendermaßen vor. Man nimmt die Ausgangszahl, teilt sie durch die höchste Potenz von 16 (in diesem Falle 3), die ein Ergebnis über Null ergibt und erhält einen Bruch, dessen Integerwert die erste Hexadezimalstelle ergibt (ϕOE). Nun multipliziert man diesen mit der 16'er Potenz und berechnet daraus eine neue Basiszahl, die man dann durch die nächsttiefere Potenz teilt, und so weiter ...). Der Motorola 68000 Microprozessor, der ein 16/32 Bit Prozessor ist, unterscheidet zwischen Bytes, Worten (Words) und Langworten (Longwords). Ein Byte umfaßt 8 Bits, ein Word 16 und ein Longword 32 Bits. Dies bedeutet, daß man innerhalb eines Bytes die

Zahlen bis 255 (\$FF), in einem Word die Zahlen bis \$FFFF und in einem Longword die Zahlen bis \$FFFFFF darstellen kann. In der Assembler-syntax werden diese Datenlängen durch das Anhängen der Kürzel .b (Byte), .w (Word), .l (Longword) an den jeweiligen Befehl unterschieden. Wir wenden uns jetzt den Registern zu. Der 68000'er unterscheidet zwischen acht Daten- und sieben Adressregistern (d0–d7 bzw. a0–a6). Man kann sich diese Register wie BASIC-Variablen vorstellen. So kann man sie verändern, ihre Inhalte untereinander vertauschen und mit ihnen Rechenoperationen durchführen. Dies kann man mit speziellen Befehlen erreichen, auf die wir später näher eingehen werden. Die Daten-, und Adressregister unterscheiden sich durch ihre Anwendung.

Adressregister werden (wie der Name schon sagt) vorwiegend für die Verarbeitung und Zwischenspeicherung von Adressen benutzt, mit ihnen sind jedoch nur Word oder Longwordoperationen zulässig. Datenregister werden im allgemeinen zum Ablegen von Werten gebraucht, die später weiterverarbeitet werden und keinen RAM Speicher belegen sollen. Jedoch können Adressregister auch als Datenregister fungieren, so daß 15 Datenregister zur Verfügung stehen. Außerdem besitzt der AMIGA 68000 ein Statusregister (SR), zwei Stapelzeiger (USP/SSP), und den Adresszähler (PC). Das Statusregister ist ein Word, das sich aus 2 Bytes, dem Userbyte (untere 8 Bit) und dem Systembyte (obere 8 Bit) zusammensetzt. Wir werden uns aber im Anfangsteil unseres Kurses nur mit dem Userbyte befassen. Im Statusregister sind die augenblicklichen Zustände verschiedener Flags festgehalten. Flags sind Bits, die durch ihren momentanen Wert

(gelöscht = 0/gesetzt = 1) das Ergebnis einer Rechnung genauer spezifizieren, oder durch Verändern ihres Wertes eine bestimmte (interne) Operation auslösen. Tritt zum Beispiel im Laufe einer Rechnung ein negatives Ergebnis auf, so wird das 3. Bit des Statusregisters gesetzt. Flags werden vorwiegend zu Vergleichsoperationen benutzt (auf negativ prüfen, ...). Doch nun zurück zur Bitbelegung des Statusregisters:

dez.	hex.	dez.	hex.
00	\$00	08	\$08
01	\$01	09	\$09
02	\$02	10	\$0A
03	\$03	11	\$0B
04	\$04	12	\$0C
05	\$05	13	\$0D
06	\$06	14	\$0E
07	\$07	15	\$0F

Hexadezimalsystem

76543210 system	76543210 ---enzuc
--------------------	----------------------

Bitbelegung des Statusregisters

0 C Carrybit	4 Extendedbit
1 V Overflowbit	5 unbenutzt
2 Z Null/Zerobit	6 unbenutzt
3 N Negativbit	7 unbenutzt

Das Carrybit (C) ist das sogenannte Übertragsbit, es wird gesetzt, wenn bei einer Operation ein Überlauf aufgetreten ist. Ein Überlauf tritt dann auf, wenn man zum Beispiel bei einer Rechnung die Zahlengrenzen einer Datenlänge überschritten hat, das heißt, wenn man beispielsweise versucht, mit einem Byte in größeren Werten als 255 zu rechnen. Das Carrybit stellt dann das fehlende Bit dar. Das Overflowbit (V) zeigt an, wenn sich bei einer Operation das Vorzeichen geändert hat. Das Zerobit (Z) erhält den Wert 1, wenn das Ergebnis einer Rechnung Null beträgt, wogegen das Negativbit (N) gesetzt wird, wenn das Ergebnis unter Null liegt. Es hat bei der Vorzeichenänderung +/– die gleiche Funktion wie das Overflowbit. Das Extendedbit verhält sich in Bezug auf die Darstellung eines fehlenden Bits ähnlich, wie das Carrybit.

Verlassen wir nun das Statusregister und wenden wir uns dem Stapelspeicher zu. Man kann ihn sich wie einen Stapel Bücher vorstellen. Das System des Zugriffs ist das gleiche. Denn, wenn man ein Buch haben will, muß man zuerst die Oberen herunternehmen. Nach dem selben Prinzip arbeitet der Stapel ("Last in First

Out"). Auf ihm kann man Wert zwischenspeichern, was vor allem das Betriebssystem ausnützt, um die Werte der Register nicht zu verändern. Hat man einen Wert auf den Stapel 'geschoben', so kann man auf diesen auch als erstes wieder zurückgreifen. Der Stapelzeiger hat nun die Funktion, das obere Ende des Stapels zu markieren. Der Programmzeiger (Programcounter) zeigt auf die aktuelle Adresse, an der das Programm steht, das heißt, er bezeichnet die Anweisung, die gerade ausgeführt wird.

So, genug für's erste Mal. Im nächsten Teil werden wir endlich mit den Assemblerbefehlen beginnen und eine Kurzanleitung des SEKA Assemblers vorstellen. Aber bis dahin haben Sie noch vier Wochen Zeit, um das heute gelernte zu vertiefen. Aber keine Angst, haben Sie diesen Teil des Kurses verstanden, sind Sie über das Schwerste hinweg...

Oliver Siebenhaar
Sven Stillich
Ralf Görlach

interplan GmbH
Nymphenburger Str. 134
8000 München 19
Tel.: 089-1234066

Ihr autorisierter Commodore-Händler
Beratung, Verkauf, Schulung
für

Amiga 500, 1000, 2000
Soft- und Hardware

FREIE FAHRT

AMIGA-BASIC-KURS Teil 3

Auch in diesem Teil wollen wir versuchen ein Gebiet der Basicprogrammierung verständlich darzulegen. Wir hoffen, daß bei all denjenigen, die noch nicht selbst programmieren, ein wenig Interesse geweckt wird, um sich mit dem Rechner näher auseinanderzusetzen.

In diesem Teil wollen wir uns ein wenig mit Zeichenketten beschäftigen. Zeichenketten, im Programmiermund auch als Strings bekannt, sind Variablen, die Zeichen enthalten können. Da ein Beispiel mehr sagt, als viele Worte, werfen Sie bitte einen Blick auf folgende Zeilen.

```
A$ = "Guten Tag"  
Bee$ = "Wachholderbusch"  
C2$ = "9*klug"  
Name$ = "Kalle Wirsch"
```

Das ist nichts Neues. Nur, um noch einmal kurz zu wiederholen: es ist äußerst einfach. Damit sind wir noch lange nicht am Ende. Mit String kann man nämlich ziemlich viel anfangen.

Aus eins macht zwei

Beginnen wir mit dem Verknüpfen von Strings. Dazu nehmen wir einen Neuen und schreiben den Inhalt von zwei Anderen hinein.

```
Neu$ = A$ + Name$
```

Um uns vom Ergebnis überzeugen zu können, lassen wir uns den Inhalt von Neu\$ auf dem Bildschirm ausgeben.

```
PRINT Neu$
```

Auf dem Bildschirm erscheint folgende Zeile:

```
Guten TagKalle Wirsch
```

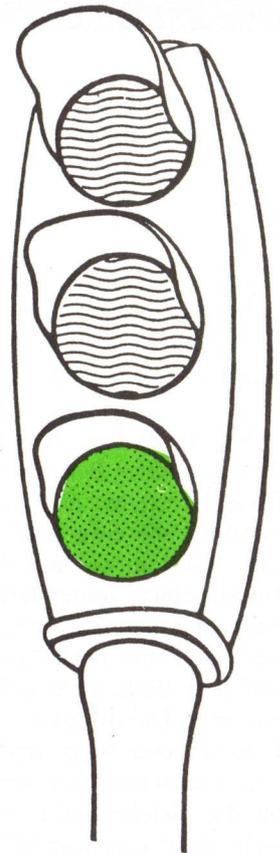
Wie wir sehen, tut der Rechner das, was wir von ihm verlangen. Nur vergaßen wir zwischen dem ersten und dem zweiten String ein Leerzeichen einzufügen. Dies müssen wir noch nachholen:

```
Neu$ = A$ + " " + Name$  
PRINT Neu$
```

Im Prinzip funktioniert alles wie im Scrabble. Zuerst nimmt der Rechner den Inhalt von A\$, legt dahinter ein Leerzeichen und danach schließlich den Inhalt von Name\$. Dies ergibt:

```
Guten Tag Kalle Wirsch
```

Vom Ergebnis überzeugen Sie sich bitte direkt am Rechner.



Die Schere

Nach der einfachen Addition mehrerer Strings wollen wir uns nun einige Befehle zum Ausscheiden bestimmter Stringteile vornehmen. Zuerst ein Beispiel:

```
A$ = "Guten Tag"  
PRINT LEFT$(A$,5)
```

ergibt

```
Guten
```

Der Befehl LEFT\$ schneidet eine bestimmte Anzahl von Zeichen aus einem String heraus und fängt dabei von links an. Ebenso der Befehl RIGHT\$. Er schneidet, wie wir jetzt schon kombinieren können, eine bestimmte Anzahl von Zeichen ab dem rechten Ende des Strings heraus. Dazu wieder einmal ein Beispiel:

```
A$ = "Guten Tag"  
PRINT RIGHT$(A$,3)
```

ergibt

```
Tag
```

Das alles ist nicht schwer. Wie wichtig diese Befehle wirklich sind, zeigt sich erst im Einsatz.

Aus dem Reich der Mitte

```
Test$ = "Rotbarschfilet"  
PRINT MID$(Test$,5,5)
```

Wir verzichten hier auf die Präsentation des Ergebnisses.

Rotation ist angesagt

Nun ist es Zeit, diese Befehle in einem größeren Beispiel anzuwenden. Dazu haben wir eine kleine Routine gewählt, die eine Zeichenkette über den Bildschirm laufen läßt. Das funktioniert folgendermaßen. Aus dem zu rollenden String wird zuerst ein Ausschnitt herausgenommen und auf den Bildschirm gebracht. Beim nächsten Durchlauf der Hauptschleife wird dieser Ausschnitt um eine Position nach rechts versetzt und wieder an die gleiche Stelle gedruckt. Wie sich schon vermuten läßt, eignet sich der MID\$-Befehl prächtig dazu. Um einen sich nach rechts verschiebenden Ausschnitt zu bekommen, muß man einfach den mittleren Parameter in einer Schleife hochzählen lassen. Um die Routine flexibel und universell einsetzbar zu machen, wurde sie in eine Prozedur gelegt, die nur noch folgendermaßen aufgerufen wird.

LAUFSCHRIFT X,Y,Test\$,Laenge

Alle Parameter können dadurch frei gewählt und die Routine mehrfach aufgerufen werden.

```
' Scrollroutine  
'  
' Erzeugt eine Laufschrift variabler Länge  
'  
a$="Die aktuellen NEWS ..... AMIGA auf Vormarsch"  
a$=a$+"... die bekannte Softwarefirma Kallesoft steigt nun"  
a$=a$+" voll ins Softwaregeschehen ein. Das erste Produkt "  
a$=a$+" K A L L O M A T war schon auf dem IBM ein Erfolg ....."  
'  
CLS  
'  
'  
x=10:y=10:laenge=40  
WHILE INKEY$=""  
  Laufschrift x,y,a$,laenge  
WEND  
END  
  
SUB Laufschrift (xpos,ypos,t$,ausschnitt) STATIC  
  IF ausschnitt >79-xpos THEN ausschnitt=79-xpos  
  t$=SPACE$(ausschnitt)+t$  
  FOR i=1 TO LEN(t$)  
    LOCATE xpos,ypos  
    PRINT MID$(t$,i,ausschnitt)  
  NEXT i  
END SUB
```

Beim Durchsehen des Listings stellt man fest, daß sich ein weiterer String-Befehl eingeschlichen hat; 'LEN'. Er gibt nichts anderes als die Länge eines Strings zurück, was im praktischen Einsatz von Wichtigkeit ist. Dazu ein kleines Beispiel. Wir wollen einen String zentriert auf dem Bildschirm ausgeben. Dazu brauchen wir zwei Informationen. Erstens wie breit ist der Bildschirm? und zweitens wie breit ist mein auszugebender Text? Nehmen wir an, wir befinden uns in der mittleren Auflösung. Der Bildschirm ist folglich 80 Zeichen breit. Wenn wir den Text "Oh wie wunderschön ist eine Bootsfahrt zu zweit" in die Bildschirmmitte printen wollen, so müssen wir die Anzahl der Leerzeichen vom linken Rand errechnen. Da dieser Text 47 Zeichen breit ist, müssen wir 16 oder 17 Zeichen einrücken ($[80-47]/2 = 16.5$). Als Listing sähe das folgendermaßen aus:

```
Text$ = "Oh wie wunderschön ist eine  
Bootsfahrt zu zweit"  
Breite = 80  
PRINT TAB ( (Breite-LEN(Text$) )  
/2);Text$
```

Ähnlich dem obigen Beispiel sollten Sie nun einmal versuchen, das ganze in eine Prozedur zu verwandeln, die sich mit folgendem Aufruf aufrufen läßt.

Zentriere (Text\$)

Her mit den Zeichen

Um die Palette der String-Befehle fortzuführen, wollen wir hier noch einen weiteren vorstellen 'INSTR'. Er sucht eine Zeichenkette in einer anderen. Praktische Anwendung findet dieser Befehl vor allem in der Dateiverwaltung und zwar dann, wenn man nach einem bestimmten Namen sucht. INSTR schaut nach, ob der Suchstring enthalten ist und gibt sodann die Position desselben aus. Nehmen wir folgendes Beispiel: Wir haben in unserer Datei fünf Personen gespeichert. In Wirklichkeit können dies natürlich auch 500 sein.

In diesem Beispiel werden die obigen Daten durchsucht, nach einer Person, von der man nur ein Bruchstück des Namens kennt. Ist dieses Bruchstück enthalten (in unserem Fall 'mann', so liefert die Funktion eine Position größer als Null, ist er nicht enthalten, ist der Rückgabewert Null. Deshalb erfolgt die Abfrage auf 'Größer Null'. War die Suche erfolgreich, so erscheinen alle passenden Namen auf dem Bildschirm. In der Datenverwaltung wäre diese eine Routine zum Suchen. Trotz ihrer Kürze ist diese Routine sehr zweckmäßig. (siehe Listing 2)

Lies die Daten

Auch hier haben sich wieder zwei neue Befehle eingeschlichen, 'DATA' und 'READ'. Der Befehl READ macht nichts anderes, als die Daten hinter den Befehlen DATA in eine Variable einzulesen. Das Programm wird nicht durch die Platzierung der Datas gestört. Hauptsache ist das Vorhandensein der Datas.

Die erste READ-Anweisung holt sich in unserem Beispiel die Zeichenkette "Alfons Schortz" und liest sie in die Variable D\$ ein. Beim nächsten READ-Befehl wird die nächste Zeichenkette eingelesen, und zwar ebenfalls in D\$, ganz nach belieben. Der Inhalt von D\$ lautet dann "Uwe Berddels". Dies geht solange, bis keine DATAS mehr da sind. In unserem Fall wird dieser Fall abgefangen ("XXX"), damit der Basic-Interpreter nicht mit einer Fehlermeldung stehen bleibt.

Natürlich kann man mit DATAS nicht nur Zeichen, sondern auch

```
' Suchroutine in bestehenden Daten
'
DATA "Alfons Schortz"
DATA "Uwe Berddels"
DATA "Petra Mann"
DATA "Extremus Fallus"
DATA "Peter Petersen"
DATA "Kalle Wirsch"
DATA "Gabi Whlmann"
DATA "Heike Fachs"
DATA "Andrea Flachmann"
DATA "XXX"
'
CLS
' Suchen nach "mann"
'
Such$="mann"
'
READ D$ ' Ersten Wert
'
WHILE D$<>"XXX"
  IF INSTR (D$,Such$)>0 THEN PRINT D$
  READ D$ ' Nächsten Namen lesen
WEND
'
END
```

Listing 2: Sag mir wo die Männer sind

Zahlen einlesen. Werfen Sie dazu doch einmal einen Blick in das Handbuch.

Groß oder klein?

Einen kleinen Haken hat die Abfrage noch, sie findet nur zwei Namen, obwohl dreimal das Bruchstück 'mann' enthalten ist. Doch halt, da ist ein Unterschied. Der eine Name beginnt mit einem großen 'M'. Da liegt des Rätsels Lösung. Da Rechner bekanntlich dumm sind, unterscheiden Sie die Klein- und Großbuchstaben. Also muß etwas geschehen. Gesagt getippt, man befiehlt seinem kleinen grauen „Schreibtischfüller“ keine Rücksicht auf Klein- und Großschrift zu nehmen. Dazu nehmen wir den Befehl UCASE\$ zur Hilfe. UCASE\$ wandelt einen String in Großbuchstaben um.

```
A$="MERLIN-Computer GmbH"
B$=UCASE$(A$)
PRINT B$
```

Als Ergebnis erhält man A\$ in Großbuchstaben, nämlich

MERLIN-COMPUTER GMBH

Wir setzen diesen Befehl in unsere Suchroutine ein:

Nun findet der Rechner auch Frau Petra Mann.

```
' Durch Abändern findet man das was man will
'
' IF INSTR (UCASE$(D$),UCASE$(Such$))>0 THEN PRINT D$
```

Listing 3: Die abgeänderte Abfrage

Soviel zu den String-Befehlen. Wie in jedem Teil darf auch in diesem die Mahnung an die Basic-Einsteiger nicht fehlen, stark zu üben, da dies der einzige Weg ist, den Rechner zu beherrschen. Wenn Sie Spaß daran gefunden haben, ist diese Mahnung natürlich überflüssig und Sie werden diese Serie sicher bald überholen. Also dann ans Werk.

(HS)



COMPUTERSOFTJONIGK

AMIGA SPIELE		S.D.I.	
A Mind Forever Voyaging	129,-	Shanghai	86,-
Adventure Construction Set	86,-	Sindbad	86,-
Alien Fires	89,-	Sorcerer	89,-
Archon II	89,-	Space Battle	29,95
Artifax	89,-	Starcross	89,-
Balance of Power	98,-	Starglider	86,-
Barbarien	79,-	Stationfall	98,-
City Defence	29,95	Strip Poker	29,95
Championshio Football	86,-	Swooper	49,-
Chessmaster 2000	129,-	Turbo Car	69,-
Crimson Crown	89,-	Terrorpods	86,-
Cruncher Factory	29,95	The Guids of Thieves	86,-
Deep Space	96,-	The Hally Project	79,-
Defender of the Crown	86,-	The Pawn	79,-
Deja Vu	89,-	Ultima III	86,-
Demolotion	29,95	Uninvited	89,-
Dr. Fruit	29,95	Winnie the Poo	79,-
Emerald Mine	29,95	Willy the Kid	29,95

PREISHITS DES MONATS

PHANTASIE III	69,-	TYPHOON	49,-
OGRE	79,-	KAMPFGRUPPE	96,-
GUNSHIP	86,-	VIZAWRITE AMIGA	198,-
Fortress Underground	29,95	AMIGA ANWENDER	
Farytale	119,-	Aegis Images	98,-
Gokart Racing	29,95	Aegis Artpack	86,-
Garrison	69,-	Aegis Impact	439,-
Gold Runner	86,-	Aegis Sonix	198,-
Knight Orc	69,-	Gismoz(Kalkulation + Datenbank)	198,-
Karate Kid II	86,-	K-Seka (68000 Assembler)	249,-
Karate King	49,95	Marauder II	128,-
Leather Goddess	96,-	Metacomco Cambridge Lisp	498,-
Marble Madness	86,-	Modula II Compiler Dev	449,-
Mindshadow	86,-	Page Setter (Desktop)	399,-
Portal	119,-	Prism (Malprg. 4096 Farben)	179,-
Rodwar 2000	86,-	Superbase (Deutsch)	249,-
Racter (Sprachversion)	84,-	Videoscape 3D	498,-
Rocket Attack	29,95	Digi-View (Videodigitizer)	498,-

★★ Public Domain Liste anfordern ★★ ★★ Preisänderungen vorbehalten ★★

CSJ COMPUTERSOFT JONIGK ★ Händleranfragen erwünscht
An der Tiefenriede 27 · 3000 Hannover 1 · Tel. Bestellservice (05 11) 88 63 83
Riesenauswahl an Software · sofort CSJ NEWS anfordern (Computertyp angeben)
Versand Inland: Vorkasse + 2,50 DM (Euroscheck in DM), per Nachnahme + 7,- DM

GO AMIGA!

Sehr geehrter Kunde,

Seit gut einem Jahr vertreiben wir Software und Hardware fuer den Amiga Computer, und sind seitdem einer immer groesser werdenden Anzahl von Anwendern ein Begriff geworden. Bei dem Verkauf von hochwertiger Software ist es nicht nur wichtig, die entsprechende Software moeglichst schnell auf den deutschen Markt zu bringen, sondern auch immer einen moeglichst optimalen Kundendienst zu gewahren. In diesem Sinne haben wir unseren Dienstleistungsbereich ganz erheblich ausgebaut, zu Ihrem Vorteil.

Herzlichst,

SOFTWARELAND AG ZUERICH

UPDATE-SERVICE: Auf alle Amiga-Anwenderprogramme, die mehr als DM 100.00 kosten, gewahren wir ein Jahr ab Rechnungsdatum einen internen Update-Service. Senden Sie daher Ihre Originalregistrierkarte einfach an uns. Alles weitere erledigen wir. Wir informieren Sie jeweils schriftlich ueber ein Update, was es eventuel kostet usw.

INFORMATIONSDIENST: Wir erstellen ein Prospektverzeichnis ueber erhaeltliche Testberichte und Produktkurzinformatoren. Anhand dieser Liste koennen Sie sich, gegen einen frankierten und rueckadressierten DIN A5 Briefumschlag, diese Informationen zusenden lassen.

UEBERSETZUNGSDIENST: Deutschsprachige Anleitungen werden fuer ausgewaehlte Anwenderprogramme durch uns erstellt. Lieferbare Anleitungen sind in unseren Preislisten einzeln aufgefuehrt.

DEMODIENST: Demonstrationsdisketten erhalten Sie gegen eine Schutzgebuehr inkl. Testberichte oder Produktkurzinformatoren. Lieferbare Demos sind in unseren Preislisten einzeln aufgefuehrt.

SORTIMENT: Softwareland ist einer der ersten Adressen fuer Amiga-Software und Hardware. Die DataWelt schreibt in Ausgabe 9/87: "Was die Aktualitaet betrifft ist nach wie vor Softwareland in der Schweiz ... am zuverlaessigsten."

LAGER: Ein staendiges DM 100'000.00 Lager garantiert optimale Auslieferungszeiten.

PRODUKTKONTROLLE: Laufende Ueberwachungen ermoeeglichen ein Minimum an defekter Ware. Zudem wird unsere Software staendig auf den neusten Versionsstand gebracht, und damit alte Versionen gar nicht ausgeliefert.

PUBLIC DOMAIN: Wie fuehren ein grosses Public Domain Angebot und informieren Sie mittels ausfuehrlichen, deutschsprachigen Gesamtlisten.

LIEFERUNG: Auslieferung versteuert und verzollt ab Lager BRD.

PREISE: 90-Tage Preisgarantie, damit Sie niemals fuer das gleiche Produkt am naechsten Tag weniger bezahlen muessten.

Bestellservice:

BRD: 0041-1-3115959
CH: 01-3115959

Geschäftszeiten:
10.00-12.30, 13.30-18.30 Uhr, außer montags,
Sa.: 10.00-16.00 Uhr.

Versand ins Ausland nur Vorkasse (Scheck, bar,
Visa Card, Master Card) zzgl. DM 7,- Porto.

softwareland
Franklinstraße 27
CH-8050 Zürich (Schweiz)

GO AMIGA!

TEXTVERARBEITUNG und DESKTOP PUBLISHING

GoAmiga Text a.A
 Prowrite dtsh ... 272
 Talker 135
 Dynamic Word a.A
 Word Perfect dtsh a.A
 Shakespeare 435
 PageSetter profess. 765
 Publisher 386
 City Desk 290
 Zuma Fonts 68
 Fast Fonts 78
 Font Set I 68

DATENBANKEN

GoAmiga Datei 199
 Microfiche Filer 192
 Organize 193
 Acquisition 579
 dBMan 290

TABELLENKALKULATION

Analyze 2.0 290
 Haicac 116
 Maxiplan plus 385
 Maxiplan 500 288

INTEGRIERTE SOFTWARE

Logistix 290
 VIP Professional . 290

DFUEBERTRAGUNG

Diga 155
 Online 135
 Bulletin Board Sys 193

GRAFIK SOFTWARE

Images 78
 Impact 174
 Brushworks 1,2,3 je 58
 Butcher 2.0 79
 Deluxe Paint 2 ... 225
 Deluxe Data 1-3 je 58
 Prism 135
 Express Paint 155
 Analytic Art 116
 Digi Paint dtsh . 146
 DCS Apokalypse ... 98
 DCS Starbird 98

GRAFIK HARDWARE

Digi View 386
 Adapter AM 500/2000 49
 Genlock PAL 386
 Easel 1000 963
 Easel 500/2000 ... a.A

VIDEO

Animator/Images .. 271
 Deluxe Video 1.2 . 225
 EF/X 570
 Videoscope3D dtsh 386
 GoAmiga Titel 99
 Videotitler a.A
 Animator Apprent. a.A
 TV Text 193
 Forms in Flight .. 155

Sculpt 3D 193
 Video Wizard 116

CAD

Draw plus 502
 Logicworks 193
 Dynamic CAD 965

MUSIK SOFTWARE

Sonix 155
 Deluxe Musik 2.0 ... 193
 Musik Data 1 58
 Pro Studio 288
 Musik X a.A
 Audiomaster a.A
 ETC Sampleware a.A

MUSIK HARDWARE

Future Sound 338
 ECE Midi 1000 ... 116
 ECE Midi 500/2000 116
 GOL Midi Interface 153
 MIM Midi Interface 95
 Sampler 1000 ... 192
 Sampler 500/2000 192
 SUN Sound Sampler 174

SPIELE

Accolade
 Mean 18 Golf 39
 Activision
 Hacker 1 39
 Aegis Development
 Arazoks Tomb 97
 Baudville
 Video Vegas 68
 Classic Image
 Diablo 49
 Dark Horse
 Chessmate 58
 Kings Quest 78
 Ogre 58
 Finally Software
 Dr. Xes 97
 Infinity Software
 Galileo 97
 ISM
 Surgeon 97
 Jagware
 Alien Fires 78
 Micro Illusions
 Fairy Tale Advent 97
 Land of Legends 97
 Fire Power 49
 Turbo 49
 Galactic Invasion... 49
 Miles Computing
 Quintettes 87
 Mindscape
 Shadowgate 97
 Sinbad & Falcon . 78
 Brataccas 39
 Psygnosis
 Arena 39
 Sublogic
 Flight 2 deutsch... 127
 Interkabel zu FS2 10
 Scenery Disk 7 49

Scenery Disk 11 49
 Scenery Europa..... a.A

BASIC

AC Basic 377
 True Basic 193
 True Runtime 290
 True Developers 97
 True 3D Graphics 97
 Basic Professional... 386

MODULA 2

Modula 2 - Dev ... 290
 Modula 2 - Kermit 58
 Modula 2 - Examples 49
 Modula 2 - Editor 97
 Modula 2 - Grid 97
 Modula 2 dtsh 270

PASCAL

Pascal Prof. 386
 MCC Pascal 193

C

Lattice C + TxUt . 450
 Manx C - Dev 578
 Manx C - Com 963

FORTRAN

AC Fortran 570

ASSEMBLER

MCC Assembler 193
 MCC Toolkit 97

KOPIERPROGRAMME

Mirror 97
 Marauder 2 78

WORKBENCH + CLI

DOS Express 58
 Zing englisch 155
 CLI Mate 1.2 engl... 68
 MCC Shell 135

DIENSTPROGRAMME

Floppy Accelerator 68
 Transfer Atari-AM 107
 Transfer IBM-AM 135
 Transfer C64-AM 97
 Hacker Pack 97
 Explorer 97
 Drive Alignment 578
 Zing Keys englisch 97
 TXED Texteditor .. 78
 Disk Pro plus 58
 Custom Screens ... 135
 Power Windows ... 174
 Diskettenverwalter 99
 Harddisk Backup .. 139
 Encore a.A
 Wow a.A
 Key Genie 97

PRODUKTIVITAET

Gizmos 2.0 135 IFF

GRABBER

Grabbit 58
 DBuddy a.A

ASTROLOGIE

Programm d. Lebens 149

SPEICHERWEITERUNGEN

Alegra 0,5 M RAM 732
 Alegra Expansion 58
 Insider 1 M RAM 714
 Alegra 2 M RAM 1446
 Cage 2MB RAM 1349
 Cage 4MB RAM 2505

HARDDISK

C Ltd. 22 M 1930
 Supra 20 M 1921

C64 EMULATOR

Go 64 a.A

HARDWARE

ALLGEMEIN

Kickstart Eliminat 251
 Kwikstart 327
 Diskmappe fuer 3,5" 19
 MousePad klein ... 10
 MousePad gross ... 20
 FlickerMaster 35
 Kabel parallel 35
 Kabel RS 232 m/w 29
 Kabel RS 232 m/m ...29
 Kabel Midi 25

BUECHER ENGLISCH

Amiga DOS Manual . 49
 User Guide Graphics 37
 The Amiga System . 31
 Microsoft Amiga .. 39

BUECHER DEUTSCH

Handbuch Flight 2 . 30
 Handbuch DPaint 2 a.A
 ROM Kommentiert ... 69

ZEITSCRIFTEN ENGLISCH

Amazing Computing . 7

AMIGA MAGAZINE

Jumpdisk 10

PUBLIC DOMAIN

FAUG Hot Mix je . 12
 FRED FISH je 12
 AMICUS je 12

DEMODISKETTEN

Demo TV Text 15
 Demo Zing 15
 Demo GoAmiga Datei 15
 Demo Perfect Sound 15
 Demo Zuma Fonts ... 15
 Demo Animator 15
 Demo Draw 15
 Demo True Basic ... 15
 Demo PageSetter ... 15

MINDESTBESTELLWE-RT DM 50.00 pro Auftr.
 Gueltig ab 1.10. 87

Diese Preisliste ist nur ein kleiner Auszug aus unserem Gesamtortiment. Sie erhalten unsere aktuelle Liste mit vielen attraktiven Angeboten kostenlos. Preise DM.

Bestellservice:

BRD: 0041-1-3115959
 CH: 01-3115959

Geschäftszeiten:
 10.00-12.30, 13.30-18.30 Uhr, außer montags,
 Sa.: 10.00-16.00 Uhr.

Versand ins Ausland nur Vorkasse (Scheck, bar,
 Visa Card, Master Card) zzgl. DM 7,- Porto.

softwareland
 Franklinstraße 27
 CH-8050 Zürich (Schweiz)

* = Lieferbar nach Verfügbarkeit.

Lieferungen ins Ausland sind zu deklarieren.

Preisänderungen vorbehalten.

Arbeitserleichterung:

Luxuriöse KOMMANDO OBERFLÄCHE

Wer mit dem CLI gelegentlich zu tun hat, weiß, wie nervenaufreibend die Arbeit sein kann. Kleinste Tippfehler zwingen oftmals dazu, lange Kommandos wieder zu löschen, sei es nur wegen eines kleinen Fehlers im ersten Wort. Vor allem die Handhabung von Compilersystemen mit vielen Optionen erwächst mitunter zum Chaos. Dieses Programm soll Abhilfe schaffen für alle jene, die nicht schon Besitzer einer Shell geworden sind.

Zum Ersten ermöglicht es die Editierung der Kommandozeile mit nebenstehenden Befehlen.

Die Funktionstasten sind beliebig belegbar in Zusammenhang mit <ALT> <CTRL>SHIFT> u.a., genau genommen mit einem Mehrfachen von zehn Möglichkeiten.

Einige Tasten sind für die Bedienung des Aztec-C belegt. Hier können Optionen und ein aktueller Arbeitsfile angegeben werden, der auf <F2> mit den eingestellten und dann auch angezeigten Optionen compiliert und gelinkt wird.

Die schon belegten Tasten und ihre Funktionen lauten wie folgt:

	Löscht Zeichen rechts vom Cursor.
<SHIFT> + 	Löscht Wort rechts vom Cursor
<BACKSPACE>	Löscht Zeichen links vom Cursor
<SHIFT> + <BACKSPACE>	Löscht Wort links vom Cursor
<SHIFT> + <CURSOR LINKS>	Sprung an Zeilenanfang
<SHIFT> + <CURSOR RECHTS>	Löscht Zeichen rechts vom Cursor
<ALT> + <CURSOR LINKS>	Sprung an Wortanfang links
<ALT> + <CURSOR RECHTS>	Sprung an Wortanfang rechts
<CURSOR LINKS>	Ein Zeichen nach links
<CURSOR RECHTS>	Ein Zeichen nach rechts
<CURSOR HOCH/RUNTER>	Scrollt durch die 10 letzten Command

<F1>	Ediere eingestellten Sourcefile
<F2>	Compiliere und Linke den Sourcefile
<F3>	Führe das entstandene Programm aus
<F4>	Drucke den Sourcefile
<F5>	Ediere Compiler-Optionen (nur Optionen ohne Namen, da dieser automatisch angehängt wird)
<F6>	Ediere Linker-Optionen (auch nur die Optionen)
<F7>	Compiler Error Ausgabe kann 'to PAR:' umgelenkt werden und wieder auf den Bildschirm (Aktuelles Ziel der Compilerausgabe wird angezeigt)
<F8>	Save alle Optionen und Sourcefilenamen nach SYS2: examples (Wenn die Optionen gespeichert sind, werden sie beim Shell Neustart gleich gesetzt)
<F9>	Neuer aktueller Sourcefile
<F10>	CLI Commando 'DIR'.

- <SHIFT> + <F1> Ediere aktiven Pre-Include.
- <SHIFT> + <F2> Precompiler aktiven Pre-Include.
- <SHIFT> + <F4> Drucke Pre-Include.
- <SHIFT> + <F5> Deaktiviert Pre-Includes
- <SHIFT> + <F6> Aktiviert Pre-Includes.
(wird angezeigt)
- <SHIFT> + <F10> Zeigt ein 'DIR' von SYS2: examples/pre.

Um diese Belegung zu benutzen, müßte es einen 'PATH' über die Quelltext- und Compiler-/Linkerkommando-Directories geben. Die Quelltext-Directory sollte die aktuelle Directory sein, da die Ergebnisse in diesem abgelegt werden.

Es ist auch leicht möglich, die Tastenbelegungen selbst zu realisieren. Im Quelltext muß an der Stelle, die eine Funktionstaste abfragt, ein String aufgebaut werden, der ein gültiges AmigaDOS Kommando darstellt. Ein Pointer auf das erste Zeichen des Strings muß dann an die

das Ausgabefenster ist, in dessen 'Titlebar' immer der letzte ausgeführte Befehl angezeigt wird. Dieses Fenster war vorher das 'CLI', welches lahmgelegt wurde.

Das kleine Fenster beinhaltet die Eingabezeile und eine Anzeige der Namen und Optionen für den Compiler. Die Art des Fensters kann, mit etwas Wissen über Window Strukturen im INTUITION, leicht geändert werden. (Siehe INTUITION-Kurs KickStart Erstaussgabe)

Die Hauptschleife des Programms wartet auf den Druck einer Taste,

Compilieren und Linken eines Programmes. Die Arbeitsweise des Programmes, und dessen Steuervariablen ist in 'Remarks' dem Listing an gemerkt.

Natürlich müssen und sollten die Remarks, ebenso wie die Zeilennummern, nicht mit abgetippt werden.

Beim Benutzen dieser Shell muß folgendes beachtet werden:

Befehle, die die Dateihierarchie beeinflussen, dürfen nicht mit RUN gestartet werden. (CD, ASSIGN, PATH). Ebenso dürfen diese Befehle nicht umgelenkt werden, was allerdings auch keinen Sinn ergäbe.

Einen zweiten Task dieser Shell zu fahren, ist nur begrenzt sinnvoll. Allerdings ist ein neues 'CLI' wie gewohnt zu starten, um sich z.B. in dessen Window ein Directory zeigen zu lassen, an dem man mit der Shell arbeiten will. Einen zweiten Task' der Shell, könnte man nur von einem neu geöffneten 'CLI' aus, sinnvoll fahren, da ansonsten zwei Eingabefenster ein Ausgabefenster besäßen.

Bei der Eingabe der Optionen für Compiler und Linker, wird der File name nicht mitangegeben, da dieser automatisch angehängt wird.

Die Kommandoingabe erfolgt ausschließlich in Kleinbuchstaben.

<SHIFT>, <CTRL> etc. müssen jedesmal, wenn sie wirksam sein sollen, neu gedruckt werden. Sie gelten nur für das nächste Zeichen.

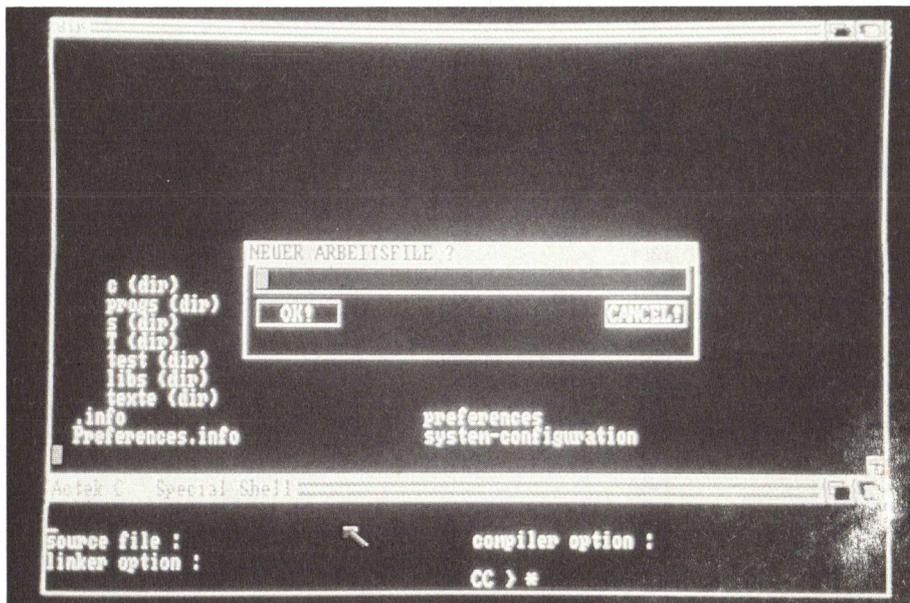
Das Programm wurde mit dem Aztec-C Compiler Version 3.40a 'cc c-shell.c' und dem Linker 'ln c-shell.o -lc' erstellt.

Zur Benutzung der Version 3.20 unter KickStart 1.1 muß der Befehl

ActivateWindow (windowname): durch

IntuitionBase->ActiveWindow = windowname; RemakeDisplay (); ersetzt werden. Bitte dafür die entsprechenden Hinweise im Listing beachten.

(A.D.)



Funktion commando (); übergeben werden. Dies kann für eine Funktionstaste beliebig oft geschehen, wodurch ganze BATCH-Files in das Programm integriert werden können. Dies hat gegenüber dem Amiga-Dos Kommando 'EXECUTE' (nicht zu verwechseln mit der aus 'C' aufgerufenen Funktion Execute (); zum Vorteil, daß kein temporärer File aufgebaut werden muß.

Das Programm präsentiert sich mit zwei Fenstern, deren Großes Oberes

stellt fest, ob diese ein Schriftzeichen wie "abc+ - d*/" etc. ist, oder eine Steuertaste wie <SHIFT>, <RETURN>, <F6> etc.

Ist es ein Schriftzeichen, so wird mit diesem ein String aufgebaut, der über Steuertasten manipuliert, d.h. ediert werden kann.

Von gewissen Eigenständigen Befehlskonsequenzen, wie z.B. dem

ns2.c

```

1: #include <intuition/intuitionbase.h>
2: #include <functions.h>
3: #include <libraries/dos.h>
4: #include <libraries/dosexterns.h>
5: #include <devices/inputevent.h>
6:
7: #define CO      &comp_opt[0] /* Adresse */
8: #define LO      &link_opt[0] /* der */
9: #define WF      &work_file[0] /* Compiler-
10: #define PF      &pre_file[0] /* Variablen */
11: #define PRI     &preinc_str[0]
12: #define FK     &function_key_command[0]
13: #define SWT     SetWindowTitles
14: #define PFC     SWT(ow,FKC,-1L)
15: #define IMMER   for(;;)
16: #define TASTE   botschaft->Code
17: #define QUALI   botschaft->Qualifier
18: #define ART     botschaft->Class
19: #define RASTP   iw->RPort
20: #define STL     str_len(&cmd_str[0])
21: #define AMIGAL  0x0066 /* Hex-Werte aller */
22: #define AMIGAR  0x0067
23: #define CAPSLOCK 0x0062 /* RAW - Tasten */
24: #define ALTL    0x0064
25: #define ALTR    0x0065
26: #define SHIFTL  0x0060
27: #define SHIFTR  0x0061
28: #define CONTROL 0x0063
29: #define DEL     0x0046
30: #define BACKSP  0x0041
31: #define HELP    0x005f
32: #define RETURN  0x0044
33: #define ENTER   0x0043
34: #define CURS_UP 0x004c
35: #define CURS_DOWN 0x004d
36: #define CURS_L  0x004f
37: #define CURS_R  0x004e
38: #define F1      0x0050
39: #define F2      0x0051
40: #define F3      0x0052
41: #define F4      0x0053
42: #define F5      0x0054
43: #define F6      0x0055
44: #define F7      0x0056
45: #define F8      0x0057
46: #define F9      0x0058
47: #define F10     0x0059
48: #define INTUITION_REV 0L
49: #define GRAPHICS_REV 0L
50:
51: struct NewWindow newwind;
52: struct IntuiMessage *botschaft; /* IDCMP Daten */
53: struct FileHandle *output, *save_file;
54: struct Window *iw,*ow; /*In-,Output */
55: struct IntuitionBase *IntuitionBase; /* BASIS */
56: struct GfxBase *GfxBase; /* ADRESSEN */
57: struct FileLock *lock,*lastlock; /* DER */
58: struct FileInfoBlock *finfoptr, finfo; /*ROM-ROUTINEN*/
59:
60:
61: char empf char,cmd_str[80]
62: ,cmd_buf[10][80] = { {'\0'},{'\0'},{'\0'},{'\0'}
63: ,{'\0'},{'\0'},{'\0'},{'\0'}, {'\0'},{'\0'} };
64:
65:
66: /* die Zeichensätze d + usa */
67: char *onshift,*onunshift,*trueunshift,*onshift_d,
68: *onunshift_d,*onshift_usa,*onunshift_usa,
69: function_key_command[80];
70:
71: LONG keytyp,succ,control,alt,shift,
72: redir_errors;
73:
74: SHORT coun,len,zahl,cmd_ptr,cmd_count=0,old_cmd_count=0;
75:
76: USHORT taste;
77:
78: /* Options-Strings für das Compiler System */
79: char comp_opt[60],link_opt[60],work_file[20],
80: pre_file[20],preinc_str[40];
81:
82: main()
83: {
84: /* Ein FileInfoBlock muß an einer durch 4 Teilbaren*/
85: /* Adresse liegen (= longword boundary) */
86: /* AllocMem() erldigt dies
87:
88: finfoptr = (struct FileInfoBlock *)
89: AllocMem((long)sizeof(struct FileInfoBlock),NULL);
90: if(!finfoptr){
91: printf("\nkein Speicherplatz fuer FileInfoBlock");
92: }
93:
94: onunshift_d =
95: "[1234567890ß'\\" 0qwertzuiopü+ 123asdfghjklöä# 456<yxcv
96: bnm,-.789 ";
97: onshift_d =
98: "]:!\"$$%&/()=?':! 0QWERTZUIOPU* 123ASDFGHJKLÖÄ\` 456>YXC
99: VBNM;:_ .789 ";
100: onunshift_usa =
101: "1234567890=-\\ 0qwertyuiop[] 123asdfghjkl;' 456 zxcv
102: bnm,./ .789 ";
103: onshift_usa =
104: "!@#$$%^&*()_+; 0QWERTYUIOP{} 123ASDFGHJKL:\` 456 ZXC
105: VBNM<>? .789 ";
106: onshift = onshift_d;
107: onunshift = onunshift_d;
108:
109: /* Öffnen der Libraries */
110: GfxBase = (struct GfxBase *)
111: OpenLibrary("graphics.library",GRAPHICS_REV);
112: if (GfxBase == NULL) ende();
113:
114: IntuitionBase = (struct IntuitionBase *)
115: OpenLibrary("intuition.library",GRAPHICS_REV);
116: if (IntuitionBase == NULL) ende();
117:
118: /* File Pointer auf den Standard-Output */
119: /* hier CLI-Window */
120: output = Output();
121:
122: /* Da das CLI-Window hier aktiv ist erhält man den*/
123: /* Pointer darauf */
124: ow = IntuitionBase->ActiveWindow;
125:
126: /* initialisierung des Eingabewindows */
127: newwind.LeftEdge = 0;
128: newwind.TopEdge = 200;
129: newwind.Width = 640;
130: newwind.Height = 50;
131: newwind.DetailPen = 0;
132: newwind.BlockPen = 1;
133: newwind.Title = (UBYTE *)"Actek C Special Shell";
134: newwind.Flags = ACTIVATE|WINDOWDRAG
135: |WINDOWDEPTH|SIMPLE_REFRESH;
136: newwind.IDCMPFlags = RAWKEY;
137: newwind.Type = WBENCHSCREEN;
138: newwind.FirstGadget= NULL;
139: newwind.CheckMark = NULL;
140: newwind.Screen = NULL;
141: newwind.BitMap = NULL;
142: newwind.MinWidth = 640;
143: newwind.MinHeight = 50;
144: newwind.MaxWidth = 640;
145: newwind.MaxHeight = 50;
146:
147: if (( iw = (struct Window *) /* Window öffnen */
148: OpenWindow(&newwind)) == NULL) ende();
149:
150: curs_Pos(NULL); /* Eingabe-Cursor auf Null */
151:
152: IMMER
153: {
154: /* Eingabe-Window Aktiv machen */
155: if(IntuitionBase->ActiveWindow != iw)
156: ActivateWindow(iw);
157:
158: /* Warten auf Message vom Window (Tastendruck)*/
159: Wait(1L<<iw->UserPort->mp_SigBit);
160:
161: /* Solange Informationen kommen werden diese */
162: /* nacheinander abgearbeitet */
163: while(botschaft = (struct IntuiMessage *)
164: GetMsg(iw->UserPort))
165: {
166: /* Prüfen ob RAWKEY empfangen wurde */
167: if(ART != RAWKEY)break;
168:
169: /* RAWWERT festhalten und Message freigeben */
170: taste = TASTE;
171: ReplyMsg(botschaft);

```

```

172:
173: /* Prüfen ob RAWWERT einem Buchstaben entspricht*/
174: /* und entsprechend verzweigen */
175:
176:         if(taste < 0x0041){
177:             verarbeiteASCII();
178:         }
179:         else{
180:             verarbeiteRAW();
181:         }
182:
183:     } /* END OF while */
184:
185: } /* END OF IMMER */
186:
187: } /* END OF main() */
188:
189:
190: /*****
191: *
192: * TASTENVERARBEITUNG
193: *
194: * hier werden alle Tasten außer Schriftzeichen
195: * abgearbeitet entsprechend einer vorangehend
196: * betätigten <CTRL> <SHIFT> oder <ALT> Taste
197: *****/
198: verarbeiteRAW()
199:
200: {
201:     switch(TASTE)
202:     {
203:         case HELP:
204:             Write(output, "\n\r\n\rKommandospeicher : \n\r
205:             \n\r", 26L);
206:             for(coun=0; coun<10; coun++)
207:             {
208:                 Write(output, &cmd_buf[coun][0],
209:                 (long)str_len(&cmd_buf[coun][0]
210:                 ));
211:                 Write(output, "\n\r", 2L);
212:             }
213:             break;
214:         case SHIFTL:
215:         case SHIFTR:
216:             shift = TRUE;
217:             break;
218:         case ALTL:
219:         case ALTR:
220:             alt = TRUE;
221:             break;
222:         case CONTROL:
223:             control = TRUE;
224:             break;
225:
226:         case BACKSP:
227:             if(shift){
228:                 shift = FALSE;
229:                 while(cmd_str[cmd_ptr-1] != 32){
230:                     --cmd_ptr;
231:                     if(cmd_ptr < 0){
232:                         cmd_ptr = 0;
233:                         break;
234:                     }
235:                 }
236:                 del_String_Char();
237:             }
238:             else
239:             {
240:                 --cmd_ptr;
241:                 if(cmd_ptr < 0)cmd_ptr = 0;
242:                 del_String_Char();
243:             }
244:             zeige_String();
245:             curs_Pos((long)cmd_ptr);
246:             break;
247:
248:         case DEL:
249:             if(shift)
250:             {
251:                 shift = FALSE;
252:                 while(cmd_str[cmd_ptr] != 32
253:                 && cmd_str[cmd_ptr] != 0)
254:                     del_String_Char();
255:             }
256:             else del_String_Char();
257:             zeige_String();
258:             break;
259:
260:         case ENTER:
261:         case RETURN:
262:             cmd_count = old_cmd_count;
263:             str_copy
264:                 (&cmd_buf[cmd_count][0], &cmd_str[0]);
265:             if(++cmd_count > 9)cmd_count = 0;
266:             old_cmd_count = cmd_count;
267:             SetWindowTitles(ow, &cmd_str[0], -1L);
268:
269:             commando(&cmd_str[0]);
270:
271:             ActivateWindow(iw);
272:             curs_Pos(0L);
273:             cmd_ptr = 0;
274:             clear_inp();
275:             cmd_str[0] = '\0';
276:             break;
277:
278:         case CURS_UP:
279:             if(--cmd_count < 0)cmd_count = 9;
280:             str_copy
281:                 (&cmd_str[0], &cmd_buf[cmd_count][0]);
282:             zeige_String();
283:             curs_Pos((long)(cmd_ptr = 0));
284:             break;
285:
286:         case CURS_DOWN:
287:             if(++cmd_count > 9)cmd_count = 0;
288:             str_copy
289:                 (&cmd_str[0], &cmd_buf[cmd_count][0]);
290:             zeige_String();
291:             curs_Pos((long)(cmd_ptr = 0));
292:             break;
293:
294:         case CURS_L:
295:             if(shift)
296:             {
297:                 shift = FALSE;
298:                 cmd_ptr = 0;
299:             }
300:             else
301:             {
302:                 if(alt)
303:                 {
304:                     alt = FALSE;
305:                     while(cmd_str[--cmd_ptr] != 32
306:                     && cmd_ptr >= 0);
307:                 }
308:                 else
309:                 {
310:                     --cmd_ptr;
311:                     if(cmd_ptr < 0)cmd_ptr = 0;
312:                 }
313:             }
314:             curs_Pos((long)cmd_ptr);
315:             break;
316:
317:         case CURS_R:
318:             if(shift)
319:             {
320:                 shift = FALSE;
321:                 cmd_ptr = STL;
322:             }
323:             else
324:             {
325:                 if(alt)
326:                 {
327:                     alt = FALSE;
328:                     while(cmd_str[++cmd_ptr] != 32
329:                     && cmd_ptr <= STL);
330:                 }
331:                 else
332:                 {
333:                     ++cmd_ptr;
334:                     if(cmd_ptr > STL)cmd_ptr = STL;
335:                 }
336:             }
337:             curs_Pos((long)cmd_ptr);
338:             break;
339:
340:         case F1:
341:             if(alt)
342:             {
343:                 alt = FALSE;
344:                 Write(output, "ALT F1", 10L);
345:                 ende();
346:             }
347:             break;
348:
349:             if(shift)
350:             {
351:                 shift = FALSE;
352:                 str_copy(FKC, "RUN EM \0");
353:                 str_cat(FKC, PF);
354:                 str_cat(FKC, ".h\0");
355:                 PFC;

```

```

348:         commando(FKC);
349:         break;
350:     }
351:
352:     if(control)
353:     { control = FALSE;
354:       if(keytyp )
355:       { keytyp = FALSE;
356:         onshift = onshift_usa;
357:         onunshift = onunshift_usa;
358:       }
359:       else
360:       { keytyp = TRUE;
361:         onshift = onshift_d;
362:         onunshift = onunshift_d;
363:       }
364:     }
365:     else
366:     { str_copy(FKC,"RUN EM \0");
367:       str_cat(FKC,WF);
368:       str_cat(FKC,".c\0");
369:       PFC;
370:       commando(FKC);
371:     }
372:     break;
373:
374: case F2:
375:     if(alt)
376:     {
377:     }
378:
379:     if(shift)
380:     { str_copy(FKC,"CC \0");
381:       if(redir_errors)
382:         str_cat(FKC,"> PAR: -B \0");
383:       str_cat(FKC,"+HSYS2/examples/pre/\0");
384:
385:       str_cat(FKC,PF);
386:       str_cat(FKC,".pre\0");
387:       str_cat(FKC," \0");
388:       str_cat(FKC,PF);
389:       str_cat(FKC,".h\0");
390:       PFC;
391:       commando(FKC);
392:     }
393:
394:     if(control)
395:     {
396:     }
397:     else
398:     {
399:       str_copy(FKC,"CC \0");
400:       if(redir_errors)
401:         str_cat(FKC,"> PAR: -B \0");
402:       str_cat(FKC,PRI);
403:       str_cat(FKC," \0");
404:       str_cat(FKC,CO);
405:       str_cat(FKC," \0");
406:       str_cat(FKC,WF);
407:       str_cat(FKC,".c\0");
408:       PFC;
409:       commando(FKC);
410:       str_copy(FKC,"LN \0");
411:       str_cat(FKC,WF);
412:       str_cat(FKC,".o \0");
413:       str_cat(FKC,LO);
414:       PFC;
415:       commando (FKC);
416:     }
417:     break;
418: case F3:
419:     if(alt)
420:     {
421:     }
422:
423:     if(shift)
424:     {
425:     }
426:
427:     if(control)
428:     {
429:     }
430:     else
431:     {
432:       str_copy(FKC,WF);
433:       PFC;
434:       command(FKC);
435:     }
436:     break;
437:
438: case F4:
439:     if(alt)
440:     {
441:     }
442:
443:     if(shift)
444:     { str_copy(FKC,"type > PAR: \0");
445:       str_cat(FKC,PF);
446:       str_cat(FKC,".h\0");
447:       PFC;
448:       command(FKC);
449:     }
450:
451:     if(control)
452:     {
453:     }
454:     else
455:     {
456:       str_copy(FKC,"type > PAR: \0");
457:       str_cat(FKC,WF);
458:       str_cat(FKC,".c\0");
459:       PFC;
460:       command(FKC);
461:     }
462:     break;
463:
464: case F5:
465:     if(alt)
466:     {
467:     }
468:
469:     if(shift)
470:     { str_copy(PRI," \0");
471:       zeigeInfo();
472:     }
473:
474:     if(control)
475:     {
476:     }
477:     else
478:     { neue_Option(CO,"EDIERE COMPILER OPTIO
479:       N\0");
480:       zeigeInfo();
481:     }
482:     break;
483:
484: case F6:
485:     if(alt)
486:     {
487:     }
488:
489:     if(shift)
490:     { str_copy(PRI,"+ISYS2:examples/pre/\0"
491:       );
492:       str_copy(PRI,PF);
493:       str_copy(PRI,".pre \0");
494:       zeigeInfo();
495:     }
496:
497:     if(control)
498:     {
499:     }
500:     else
501:     {
502:       neue_Option(LO,"EDIERE LINKER OPTION\
503:       0");
504:       zeigeInfo();
505:     }
506:     break;
507:
508: case F7:
509:     if(alt)
510:     {
511:     }
512:
513:     if(shift)
514:     {
515:     }
516:
517:     if(control)
518:     {
519:     }
520:     else
521:     {
522:       if(redir_errors)
523:         redir_errors = FALSE;
524:     }

```

```

521:         else redir_errors = TRUE;
522:         zeigeInfo();
523:     }
524:     break;
525:
526: case F8:
527:     if(alt)
528:     {
529:     }
530:
531:     if(shift)
532:     {
533:     }
534:
535:     if(control)
536:     {
537:     }
538:     else
539:     {
540:         str_copy(FKC,"Optionen Speichern\0");
541:         PFC;
542:         save_file =
543:             Open("s:optionen",MODE_NEWFILE);
544:         Write(save_file,WF,str_len(WF));
545:         Write(save_file,CO,str_len(WF));
546:         Write(save_file,LO,str_len(WF));
547:         Close(save_file);
548:     }
549:     break;
550:
551: case F9:
552:     if(alt)
553:     {
554:     }
555:
556:     if(shift)
557:     {
558:         neue_Option(PF,"NEUER ARBEITSFILE ?\0
559:         ");
560:         zeigeInfo();
561:     }
562:
563:     if(control)
564:     {
565:     }
566:     else
567:     {
568:         neue_Option(WF,"NEUER ARBEITSFILE ?\0
569:         ");
570:         zeigeInfo();
571:     }
572:     break;
573:
574: case F10:
575:     if(alt)
576:     {
577:     }
578:
579:     if(shift)
580:     {
581:         str_copy(FKC,"dir SYS2:examples/pre\0
582:         ");
583:         PFK;
584:         commando(FKC);
585:     }
586:
587:     if(control)
588:     {
589:     }
590:     else
591:     {
592:         str_copy(FKC,"dir\0");
593:         PFK;
594:         commando(FKC);
595:     }
596:     break;
597:
598:     } /* End of switch(taste) */
599: } /* End of verarbeiteRAW() */
600:
601: /* Umformung der Schriftzeichen in ASCII-Werte und
602: diese an den String anhängen.
603: Edierfunktionen erfolge über die RAW-Tasten.
604: Dazu wird die Position im String cmd_ptr
605: festgehalten */
606:
607: verarbeiteASCII()
608: {
609:     if(!shift)
610:     {
611:         empf_char = *(onunshift+TASTE);
612:     }
613:
614:     if(shift)
615:     {
616:         empf_char = *(onshift+TASTE);
617:         shift = FALSE;
618:     }
619:
620:     zahl = STL;
621:
622:     while(zahl >= cmd_ptr)
623:     {
624:         cmd_str[(zahl+1)] = cmd_str[zahl];
625:         zahl--;
626:     }
627:     cmd_str[cmd_ptr] = empf_char;
628:     ++cmd_ptr;
629:
630:     zeige_String();
631:     curs_Pos((long)cmd_ptr);
632: }
633:
634: /* Zeichen auf dem cmd_ptr wird gelöscht
635: und dahinterstehende vorgezogen */
636: del_String_Char()
637: {
638:     SHORT loop;
639:
640:     for(loop = cmd_ptr;cmd_str[loop];++loop)
641:     {
642:         cmd_str[loop] = cmd_str[loop +1];
643:     }
644: }
645:
646: /* diese Funktion setzt den Cursor */
647: curs_Pos(neupos)
648:
649: LONG neupos;
650:
651: {
652:     SetAPen(RASTP,0L);
653:     Move(RASTP,2L,22L);
654:     Draw(RASTP,638L,22L);
655:
656:     SetAPen(RASTP,1L);
657:     Move(RASTP,neupos*8L+2L,22L);
658:     Draw(RASTP,neupos*8L+10L,22L);
659: }
660:
661: /* Ausstieg aus dem Programm. Freigeben aller
662: geöffneten Bestandteile an das System */
663: ende()
664: {
665:     if(finoptr)FreeMem(finoptr,(long)sizeof(struct File
666:     InfoBlock));
667:     if(iw)CloseWindow(iw);
668:     if(GfxBase)CloseLibrary(GfxBase);
669:     if(IntuitionBase)CloseLibrary(IntuitionBase);
670:     Exit(TRUE);
671: }
672:
673: /* Darstellung des Befehlsstrings im Eingabefenster */
674: zeige_String()
675: {
676:     clear_inp();
677:     Move(RASTP,2L,20L);
678:     Text(RASTP,&cmd_str[0],(long)(STL));
679: }
680:
681: /* Löschen der Eingabezeile */
682: clear_inp()
683: {
684:     Move(RASTP,2L,20L);
685:     Text(RASTP,"
686:     ",57L);
687: }
688:
689: /* Bestimmung der Stringlänge */
690: str_len(s)
691:
692: char *s;
693:
694: {
695:     char *cp = s;
696:
697:     while(*cp++);
698:     return(cp-s-1);

```

```

693: }
694:
695: /* String kopieren von Quelle nach Ziel */
696: str_copy(ziel,quelle)
697:
698: char *ziel, *quelle;
699:
700: {
701:     while ((*ziel++ = *quelle++) != '\0');
702: }
703:
704: /* String zweiter an erster anhängen */
705: str_cat(erster,zweiter)
706:
707: char *erster, *zweiter;
708:
709: {
710:     SHORT loop;
711:     while(*erster != '\0')
712:         erster++;
713:     while((*erster++) = *(zweiter++)) != '\0';
714: }
715:
716: /* Ausführung eines Kommandos nach einem <RETURN> */
717: commando(unchecked)
718:
719: char *unchecked;
720:
721: {
722:     char *name,*argv[3],*pos1,*pos2;
723:     char commandoname[20],argument[40];
724:
725:     pos1 = name = &commandoname[0];
726:     pos2 = argv[1] = &argument[0];
727:     argv[2] = NULL;
728:
729:     /* Aktivieren des Ausgabefensters */
730:     ActivateWindow(ow);
731:
732:     Write(output, "\n\r\n\r",4L);
733:     /* Überflüssige Leerstellen beseitigen */
734:     while( *unchecked == 32 )unchecked++;
735:     while( *unchecked != 32 && *unchecked != 0 )
736:         *pos1++ = *unchecked++;
737:     *pos1 = 0;
738:
739:     /* RUN und Output-Redirection wird durch
740:      Execute() realisiert */
741:     if(str_cmp("run\0",name) || *(unchecked+1)=='<'
742:        || *(unchecked+1) == '?' || *(unchecked+1)=='>')
743:         return(Execute(&cmd_str[0],NULL,NULL));
744:
745:     /* nochmals Leerstellen beseitigen und
746:      Argument bestimmen */
747:     if(*unchecked)
748:         while( *unchecked == 32 )unchecked++;
749:         while( *pos2++ = *unchecked++ );
750:     *pos2 = 0;
751:
752:     fexecv(name,argv);
753: }
754:
755: /* Vergleiche Strings*/
756: str_cmp(str1,str2)
757:
758: char *str1,*str2;
759:
760: {
761:     while( *str1++ == *str2++ )
762:     {
763:         if(*str1 == 0 && *str2 == 0)return(1);
764:     }
765:     return(0);
766: }
767:
768: /* gibt Variablen des Compiler Systems aus */
769: zeigeInfo()
770:
771: {
772:     LONG loop;
773:     for(loop=2;loop<=624;loop+=8){
774:         Move(RASTP,loop,29L);
775:         Text(RASTP," ",1L);
776:         Move(RASTP,loop,38L);
777:         Text(RASTP," ",1L);
778:     }
779: }
780:
781: }
782:
783: Move(RASTP,2L,29L);
784: Text(RASTP,"source file :",13L);
785:
786: Move(RASTP,255L,29L);
787: Text(RASTP,"comp. opt. :",12L);
788:
789: Move(RASTP,2L,38L);
790: Text(RASTP,"preinclude :",13L);
791:
792: Move(RASTP,255L,38L);
793: Text(RASTP,"link. opt. :",12L);
794:
795: Move(RASTP,120L,29L);
796: Text(RASTP,Wf,(long)str_len(Wf));
797:
798: Move(RASTP,363L,29L);
799: Text(RASTP,CO,(long)str_len(CO));
800:
801: Move(RASTP,363L,38L);
802: Text(RASTP,LO,(long)str_len(LO));
803:
804: Move(RASTP,120L,38L);
805: if(str_len(PRI)
806:    Text(RASTP,PRI,(long)str_len(PRI));
807: else
808:    Text(RASTP,"keiner",6L);
809:
810: Move(RASTP,2L,47L);
811: if(redir_errors)Text(RASTP,"CC > PAR:",8L);
812: else
813:    Text(RASTP,"CC > * ",8L);
814: }
815:
816: UBYTE puffer1[40],puffer2[40];
817:
818: SHORT xy_pt2[] = { -2,-2,-2,9,322,9,322,-2,-2,-2 };
819: SHORT xy_pt1[] = { 0,0,0,10,60,10,60,0,0,0 };
820:
821: struct Border rahmen =
822:     { 0,0,1,0,JAM1,5,(SHORT *)&xy_pt1[0],NULL };
823:
824: struct Border str_rahmen =
825:     { 0,0,1,0,JAM1,5,(SHORT *)&xy_pt2[0],NULL };
826:
827: struct StringInfo option_str =
828:     {(UBYTE *)&puffer1[0],(UBYTE *)&puffer2[0],0,40,0,
829:      NULL,NULL,NULL,NULL,NULL,NULL };
830:
831: struct IntuiText ok_txt = {(UBYTE)1,(UBYTE)0,JAM1,2,2,
832:                             NULL,(UBYTE *) " OK! \0",NULL
833:                             };
834:
835: struct IntuiText cancel_txt = {(UBYTE)1,(UBYTE)0,JAM1,2,2,
836:                                 NULL,(UBYTE *) "CANCEL!\0",NULL
837:                                 };
838:
839: struct Gadget option_gad =
840:     { NULL,10,12,320,9,SELECTED,TOGGLESELECT,
841:       STRGADGET,(APTR)&str_rahmen,NULL,0,(APTR)
842:       &option_str,NULL,NULL };
843:
844: struct Gadget ok_gad =
845:     { &option_gad,10,25,60,10,GADGNONE,GADGIMMEDIATE
846:       !RELVERIFY,BOOLGADGET,(APTR)&rahmen,NULL,&ok_txt,
847:       NULL,NULL,2,NULL };
848:
849: struct Gadget cancel_gad =
850:     { &ok_gad,270,25,60,10,GADGNONE,GADGIMMEDIATE
851:       !RELVERIFY,BOOLGADGET,(APTR)&rahmen,NULL,
852:       &cancel_txt,NULL,NULL,1,NULL };
853:
854: neue_Option(option,text)
855:
856: char *option, *text;
857:
858: {
859:     struct Window *gadwindow;
860:     struct Gadget *empf_gad;
861:
862:     newwind.LeftEdge = 150;
863:     newwind.TopEdge = 100;
864:     newwind.Width = 340;
865:     newwind.Height = 50;
866:     newwind.DetailPen = 0;
867:     newwind.BlockPen = 1;

```

```

866: newwind.Title = (UBYTE *)text;
867: newwind.Flags = ACTIVATE|SIMPLE_REFRESH;
868: newwind.IDCMPFlags = MOUSEBUTTONS|GADGETDOWN;
869: newwind.Type = WBENCHSCREEN;
870: newwind.FirstGadget = &cancel_gad;
871: newwind.CheckMark = NULL;
872: newwind.Screen = NULL;
873: newwind.BitMap = NULL;
874: newwind.MinWidth = 200;
875: newwind.MinHeight = 25;
876: newwind.MaxWidth = 640;
877: newwind.MaxHeight = 150;
878:
879: str_copy(&puffer1[0],option);
880: str_copy(&puffer2[0],option);
881:
882: if (( gadwindow = (struct Window *)
883:      OpenWindow(&newwind)) == NULL) ende();
884:
885: OnGadget(&option_gad,gadwindow,NULL);
886:
887: IMMER
888: {
889: Wait(1L<<gadwindow->UserPort->mp_SigBit);
890:
891: while(botschaft = (struct IntuiMessage *)GetMsg(gadwin
dow->UserPort))
892: {
893:     if(ART == GADGETDOWN)
894:     {
895:         empf_gad = (struct Gadget *)botschaft->IA
address;
896:         switch(empf_gad->GadgetID)
897:         {
898:             case 2:
899:                 str_copy(option,puffer1);
900:             case 1:
901:                 CloseWindow(gadwindow);
902:                 puffer1[0]=puffer2[0]=0;
903:                 return;
904:             default:
905:                 break;
906:         }
907:     }
908:     ReplyMsg(botschaft);
909:
910: } /* END OF while */
911:
912: } /* END OF IMMER */
913:
914: }
915:
916:

```



MEDIEN-CENTER

Wermingser Straße 45 (Marktpassage) · 5860 Iserlohn
 Telefon 0 23 71/2 45 99



Alle Neuheiten immer superschnell und preiswert durch USA-Direktimport!

- Aegis Souix DM 198,-
- Aegis Draw Plus DM 489,-
- Aegis Animator DM 265,-
- + Images DM 185,-
- Aegis Impact DM 245,-
- Deluxe Paint II DM 199,-
- Deluxe Paint I DM 65,-
- Art Disk DM 215,-
- Deluxe Musix DM 235,-
- Constr. Set DM 65,-
- Deluxe Print DM 245,-
- Deluxe Print DM 198,-
- Art Disk 2 DM 398,-
- Deluxe Video 1.2 DM 179,-
- Sculpt 3D DM 249,-
- Videoscape 3D
- Go Amiga Datei
- Super Base

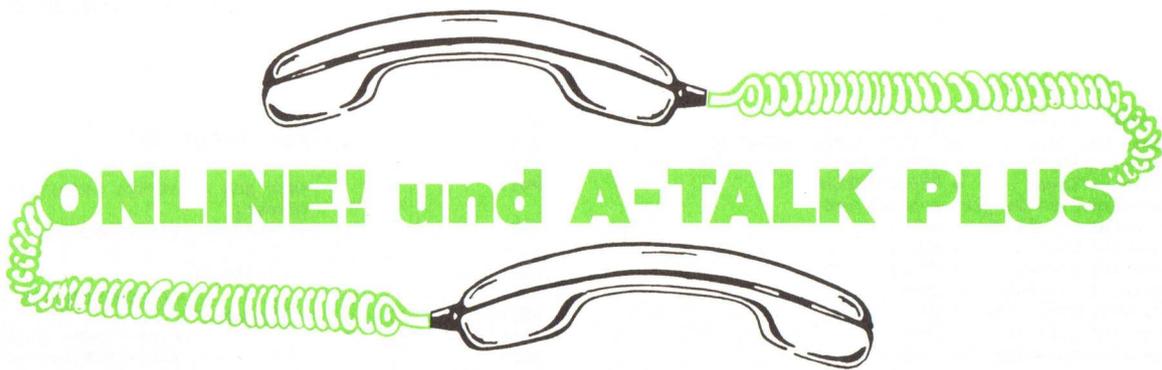
- Sindbad DM 69,-
- Faery Tale DM 129,-
- Bards Tale DM 99,-
- Barbarian DM 79,-
- Allien Fires DM 99,-
- Star Fleet I DM 109,-
- Super Huey DM 69,95
- Defender of the Crown DM 88,-
- Amiga Karate DM 69,95
- Ultima III DM 98,-
- Karate Kid II DM 69,95
- Karate King DM 49,95
- Mean 18 Golf DM 88,-
- Bad Cat DM 59,95
- Knight Orc DM 59,95
- Goldrunner DM 79,-
- Stationfall DM 99,95
- Typhoon DM 69,95
- Roadwar 2000 DM 59,95

- Amiga 2000 DM 2990,-
- mit Monitor 1081 DM 698,-
- Monitor 1081 DM 699,-
- Mon. 8833 Philips
- Speicher-erweiterung 501 DM 298,-
- Echtzeit-Digitizer DM 1748,-
- Genlock Adapter DM 1798,-
- RGB-Splitter DM 698,-
- Mouse Pad DM 17,90
- NEC-Drucker P6* DM 1148,-
- NEC-Drucker P6 Color* DM 1448,-
- NEC-Multisync DM 1398,-
- Disketten 3,5" ab 10 St. DM 29,90
- 2S2D

*mit deutschen Handbüchern u. 12 Monaten Garantie

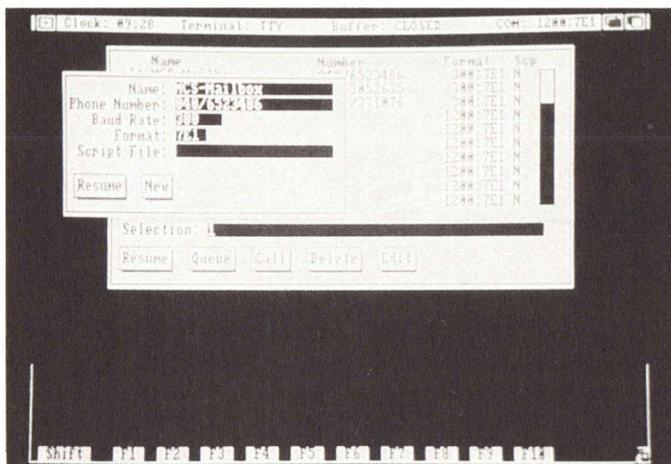
Reparatur-Schnellservice bei allen Commodore-Produkten.
 Alle Produkte lieferbar nach Verfügbarkeit.
 Lieferung per Nachnahme oder V-Scheck. Porto und Verpackung nach Aufwand.
 Bei Softwarebestellungen ab DM 300,- kostenfreier Versand.





ONLINE! und A-TALK PLUS

Zwei amerikanische DFÜ-Programme in deutschen Telefondrähten



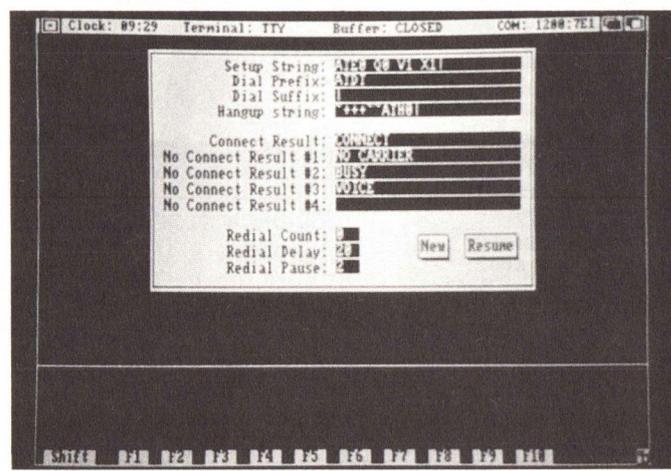
Komfortable Telefonnummereingabe mit ONLINE! 2.0

Die bundesrepublikanischen Postleitungen sind am glühen. Der Grund: In Zeiten, wo selbst ein Minister Kommissionen einsetzt, um an dem festen Gefüge der vielen Kupferdrähtchen zu rütteln und von der Gegenseite kräftig ins gelbe Posthörnchen gestoßen wird, entdecken auch hierzulande immer mehr Computerbegeisterte die Datenfernübertragung. Das diese Art der Kommunikation auch mit einem Amiga geht, hat nicht zuletzt Aegis Terminalprogramm DIGA! bewiesen. In Punkto Bedienerfreundlichkeit, Zuverlässigkeit der Datenübertragung und Schnelligkeit wurden hier Maßstäbe gesetzt. Doch Konkurrenz ist bereits in Sicht. Felsina Software aus Kalifornien und Micro-Systems aus Florida versuchen sich ebenfalls mit ihren DFÜ-Programmen A-TALK PLUS und ONLINE! 2.0 auf diesem zukunftsträchtigen Markt zu etablieren. Mit den an die deutsche Bildnorm angepaßten Terminalprogrammen stehen ihre Chancen gut. So-

wohl ONLINE! als auch A-TALK PLUS bieten hier jeden Komfort. So stehen dem ONLINE!-Benutzer in der höchsten Auflösung 71 Zeilen mal 132 Zeichen zur Verfügung. Der kleinste verwendbare Zeichensatz hat eine 5*8 Pixelmatrix. A-TALK PLUS arbeitet hingegen mit 5*6 Pixelgroßen Zeichensätzen und erreicht dadurch 80 Zeilen mit jeweils 132 Zeichen. Die Größe des sichtbaren

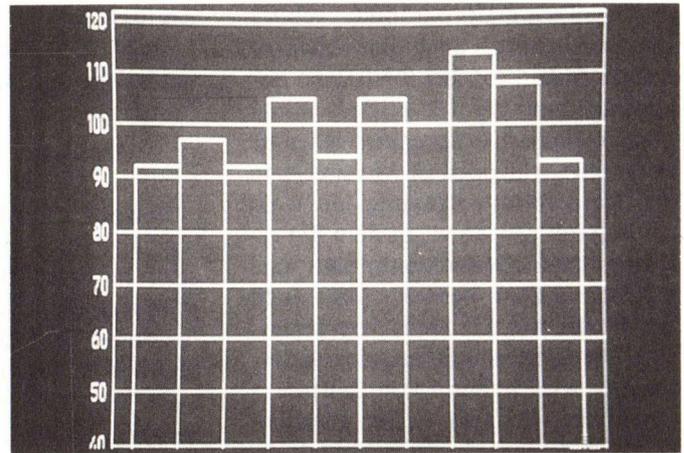
Bildschirms kann variabel bis zu einer Breite von 704 Bildpunkten eingestellt werden. Vertikal ist die Grenze erst bei 582 Pixel erreicht. Hier dürfte so mancher billige Amiga-Monitor auf der Strecke bleiben. Auch in Bezug auf Geschwindigkeit muß man kaum Einschränkungen hinnehmen. Bei ONLINE! lassen sich Baudraten von 300, 600, 1200 bis zu 19200 anwählen. A-TALK PLUS erreicht hier die gleichen Übertragungsraten. Die wenig verbreitete 600 Baudrate fehlt hingegen. Beide Programme arbeiten über Pulldown-Menüs. Ein komfortables Fast-Menü a la Aegis DIGA! über das alle Voreinstellungen festgelegt werden, gibt es nicht. Dadurch gestaltet sich das Festlegen der Default-Werte recht langwierig. Eine nötige Parametereinstellung hat man schnell übersehen. Allein bei A-TALK PLUS stehen dem Benutzer neben den Übertragungsprotokollen XModem und Kermit, 11 verschiedene Modemanpassungen zur Verfügung. Zudem kann man über Steuer-codes eigene Modemansteuerungen erstellen. Der ONLINE!-Besitzer hat

Die Anpassung verschiedener Modemtypen geschieht mittels Steuerungs-codes



ebenfalls vielfältigste Möglichkeiten. An Protokollen gibt es hier XModem, B-Protokoll, YModem, Kermit und ZModem. Für die Kommunikation mit den verbreitetsten Klein- und Großrechner ist ebenfalls gesorgt. So gibt es Emulationen der DEC-Terminals VT52, VT100 und VT102. Neben dem normalen Dump-Terminal TTY läßt sich sogar der voll grafikfähige Tektronix 4010/14 emulieren. A-TALK PLUS verständigt sich auf Wunsch auch mit einem zwar nicht grafik-, dafür aber farbfähigen ANSI-Terminal. Am meisten bietet hier jedoch das Tektronix-Terminal. Selbst komplexe Grafiken lassen sich hier über eine Superbitmap darstellen. Dabei wird in einer Auflösung von 1024*768 Pixel gearbeitet. Einen Haken gibt es jedoch bei dieser Darstellungsart. Nur Besitzer eines Amigas mit mindestens einem Megabyte ist der Anblick dieser Superbitmap vergönnt. Bei 512 KByte Amigas bekommt man lediglich die lapidare Meldung „Low on Memory“. Wie bei Aegis DIGA! lassen sich die übertragenen High-Res Grafiken beliebig vergrößern und in verschiedenen Formaten abspeichern. Neben dem A-Talk eigenen Plot-10-Format gibt es noch Weiterverarbeitungsmöglichkeiten mit DPaint II und Aegis Draw. Bei ONLINE! muß man auf diese nützlichen Zusatzfunktionen verzichten. Lediglich das Aegis Draw-Format wird hier unterstützt. Dafür gibt es die Möglichkeit, sich eingehende oder zu sendende Texte mit Hilfe der Amiga-Sprachphoneme vorlesen zu lassen. Deutsche Texte klingen hier jedoch reichlich merkwürdig. Beide Programme lassen den Betrieb in Halbduplex-Modus zu. Bei der Datenübertragung von einem Rechner zum anderen, ist diese Echofunktion unerlässlich. Einen Chat-Modus besitzen beide Programme. Hierbei wird der Bildschirm in zwei verschiedene Fenster aufgeteilt. In einem Window tippt man seine Nachricht ein und im anderen erscheinen alle empfangenen Mitteilungen. Für Telefonkonferenzen ist dies eine unerlässliche Voraussetzung. Das gleichzeitige Senden und Empfangen von Daten wie im Doubletalk-Modus von DIGA! ist damit leider nicht möglich. Für das Protokollieren einer

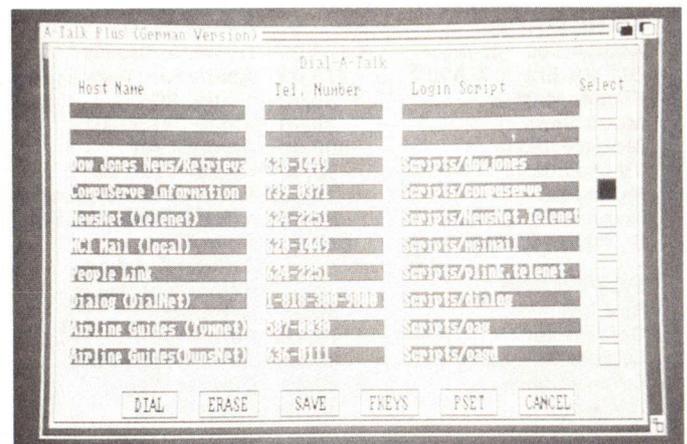
**Tektronix-Emulation
in höchster Auflösung
– kein Problem für
A-TALK PLUS**



Verbindung ist sowohl bei ONLINE! als auch bei A-TALK PLUS bestens gesorgt. Man arbeitet dabei aus Geschwindigkeitsgründen mit einem Pufferbereich. Die Größe dieses Speichers kann vom Benutzer definiert werden. Ist der Puffer voll, werden entweder die alten Daten im Schleifenmodus überschrieben, oder die Verbindung beendet. Textzeilen können zwar markiert, gelöscht, verschoben und kopiert werden, ein eingebauter Texteditor fehlt jedoch beiden Programmen. So muß man für das Erstellen eigener Datenfiles entweder den Ed, oder eine entsprechende Textverarbeitung benutzen. Zum Anloggen an eine Mailbox ist meist eine ganze Prozedur von Befehlen und Paßwörtern nötig. ONLINE! nimmt mit einer eigens für diesen Zweck entwickelten Scriptsprache, dem geplagten User so manche Arbeit ab. Mit komfortablen Befehlen lassen sich so – ähnlich einem Basicprogramm – ganze Scriptsequenzen erstellen. Mit Hilfe der integrierten Uhr kann der Amiga dann über

Nacht mehrere Telefonnummern anwählen und damit kostengünstig Meldungen von Mailboxen abrufen. Bei A-TALK PLUS ist diese Scriptsprache wesentlich bescheidener ausgefallen. Hier funktioniert das Ganze über eine Art Telefonbuch, das jedoch keine längeren Scriptsequenzen zuläßt. Bei beiden Programmen lassen sich zudem 20 Funktionstasten mit Textmakros bis zu einer Länge von 80 Zeichen belegen. Diese nützliche Automatisierungsfunktion der Scriptsprache kommt leider nur den Modembesitzern zugute. Wegen der noch hohen Kosten beim Betrieb eines postzugelassenen Modems, kommen wohl nur wenige in den Genuß dieser Zusatzfunktion. Nach der erst kürzlich in Bremen durchgeführten Beschlagnahmungsaktion von illegal betriebenen Modems, ist vom Betrieb eines solchen nur dringendst abzuraten. Hierbei wird meist recht rigoros durchgegriffen. Selbst vor einer – zumindest zeitweiligen – Sicherstellung der gesamten Computeranlage schreckt man nicht zurück. Zu einem

**Bei A-TALK PLUS
erspart das Telefon-
buch umständliche
Anwählprozeduren.**



gutem Terminalprogramm gehört selbstverständlich auch eine entsprechende Anleitung. Bei ONLINE! bekommt man ein 128-seitiges Ringbuch in englischer Sprache. Neben einem Einführungsteil ist Anfängern in der Datenfernübertragung ein eigenes Kapitel gewidmet, in dem die erforderlichen Grundkenntnisse ausführlich erklärt werden. Aber auch die Profis werden nicht vernachlässigt. Die Programmierung der Scriptsprache wird anhand von Beispielen ausführlich behandelt. Der anschließende Referenzteil gibt nochmals einen Überblick aller verfügbaren Möglichkeiten. Bei A-TALK PLUS ist die Programmbeschreibung mit 32 Seiten wohl doch etwas kurz ausgefallen.

Es gibt zwar detaillierte Beschreibungen aller emulierten Terminaltypen mit den jeweiligen Tastenbelegungen und bereits fertige Anlogprozeduren für UNIX-Rechner. Ohne das nötige Grundwissen werden Einsteiger jedoch recht alleingelassen. Ein Stichwortverzeichnis wie bei ONLINE! fehlt hier gänzlich. Auch auf die verfügbare Scriptsprache wird nur kurz eingegangen. Was man bei den Programmen hoch anrechnen muß, ist der Verzicht auf jeglichen

Kopierschutz. Weder Hardware-Dongle noch Disketten-Kopierschutz behindern die Installation auf der Festplatte.

Fazit:

Sowohl ONLINE! als auch A-TALK PLUS unterstützen die deutsche PAL-Auflösung. Beiden Programmen fehlt jedoch der dringend benötigte Texteditor. Auch eine Art Doubletalk-Modus wäre wünschenswert. Was man jedoch hauptsächlich vermißt, ist die Geschwindigkeit und Übersichtlichkeit der DIGA!-Menüs. Ansonsten gibt es außer der, bei allen amerikanischen Programmen nicht integrierten BTX-Funktion, nichts zu beanstanden. Jedes der zwei Terminalprogramme zeigte im Test sowohl Schwächen, als auch große Stärken. Wobei ONLINE!, wegen der hervorragenden Anleitung für Einsteiger, vorzuziehen ist. A TALK PLUS war zum Zeitpunkt des Tests noch nicht auf dem deutschen Markt erhältlich. Bei Erscheinen der KICKSTART wird es aber auch bei deutschen Händlern zu haben sein. Ansonsten muß ich auf die amerikanische Bezugsadresse verweisen.

(ROS)

ONLINE! 2.0

- + eingebaute Uhr
- + Scriptsache
- + alle Standardübertragungsprotokolle
- + ausführliches Handbuch

- kein eingebauter Editor
- keine BTX-Funktion
- komfortables Menühandling fehlt

Online!

Philgerma, Tel.: 0 89/39 55 51

Preis DM 198,-

A-TALK PLUS

- + hervorragende Tektronix-Emulation
- + Abspeichern in verschiedenen Grafikformaten
- + fertige Logon-Prozeduren für UNIX-Rechner

- kein Texteditor
- keine BTX-Emulation
- zu kurzes Handbuch

A TALK PLUS

PDC, 0 62 72 / 2 47 48

Preis ca. DM 180,-

IHR COMMODORE AMIGA-VERSTAND SAGT: NEHMEN SIE DEN P D C - VERSAND !

HARDWARE

AMIGA 500 mit Tutor	1.198,00
512 KB Erw. f. A 500	278,00
Flicker Master	35,00
Mous Hous	19,00
Monitor 1081	800,00
TV Modulator	50,00
Mous Pad	14,50
Trackball f. AMIGA	90,00
5 1/4" LW (40/80)	550,00

SOFTWARE

Bad Cat	59,95
Goldrunner	79,95
Leisure Suit Larry	95,00
Emerald Mine	29,00
Gerision	69,95

Amiga Karate	69,95
Dynamic Drums	170,00
Animation: Apprentice	535,00
Scuip 3D	190,00
Video Scape 3D	385,00
Analytic Art	110,00
Logistix (deutsch)	335,00
Fortress Underground	29,00
Adventure Const. Set	70,00
Alien Fires	90,00
Archon II: Adept	70,00
Balance of Power	90,00
Babarian	70,00
Bards Tale	85,00
Chessnaster 2000	70,00
Diabolo Game	72,00
Deep Space	70,00

Deja Vú	90,00
Express Paint	145,00
Faery Tale Adventure	80,00
Financial Time Machine	80,00
Facc Flopy Beschleuniger	90,00
Guild of Thieves	80,00
Halley Project	80,00
Karate King	49,00
Kings Quest I-III	90,00
Leather Goddesses of P.	70,00
Mindshadow	80,00
Portal	80,00
Roadwar 2000	85,00
S.D.I.	80,00
Sinbad	80,00
Starglider	65,00
Starfleet I	100,00
Strip Poker	55,00
Take 2 (Arena + Brataccas)	70,00
Tass Times in Tonetown	80,00



6380 Bad Homburg
Ladenpassage Alter Bahnhof
Tel. 0 61 72 - 2 47 48 u. 2 07 99

ADVERSA

Nachnahme 6.- DM
Vorkasse 4.- DM
Auslandsversand
nur gegen Vorkasse
und 10.- DM Porto

SPEICHER RIESEN

Festplatten im Test für den Amiga 1000 und eine Filecard für den Amiga 2000

Festplatten haben ihren Reiz, bieten sie zum Einen einen enormen Speicherplatz und zum anderen eine schnellere Zugriffszeit gegenüber normalen Floppystationen.

Durch diese, nicht zu unterschätzenden Eigenschaften, haben Festplatten für Anwender einen hohen Wert. Besonders Programme, die oft auf ein Speichermedium zugreifen, ist die Nutzung einer Festplatte ratsam. User werden es der Festplatte danken.

Leider ist es nicht möglich, alle angebotene Software auf einer Festplatte zu installieren, ein Kopierschutz vereitelt meistens einen Versuch der Installation. Glücklicherweise bringen immer mehr Softwarehäuser ihre Produkte ohne Kopierschutz auf den Markt, was natürlich für den Festplattengebrauch von großem Nutzen ist.

Schau her ! 20 Mbyte

Die Festplatte aus dem Hause XEBEC hat ein eigenwilliges Design (siehe Bild 1). Der hochkantstehende Aufbau benötigt wenig Platz auf dem schon sehr strapazierten Arbeitstisch. Außerhalb des Festplattengehäuses befindet sich der Konverter, der es ermöglicht, die Festplatte zu betreiben und direkt am Expansionsport des Amigas seinen

Platz findet. Über ein Kabel ist er mit der Platte verbunden. Leider besitzt der Konverter keinen durchgeschleiften Bus, sodaß keine weitere Peripherie angeschlossen werden kann. Die Speicherkapazität der XEBEC Festplatte liegt bei 20 MByte. Ebenfalls ist die Festplatte Autoparking, was soviel bedeutet, daß beim Ausschalten der Platte die Köpfe (4 bei der XEBEX) festgesetzt werden. Dadurch wird ein Aufschlagen auf der Plattenoberfläche beim Transport oder ähnlichem vermieden, was zur Zerstörung einzelner Sektoren führen kann. Als Speichermedium dient eine 5 1/4" Winchester Platte. Die Arbeitsgeräusche sind gering; lediglich beim Einschalten (hochfahren der Platte) vernimmt man etwas lautere Schallwellen vom Gerät. Die Ar-

beitsgeschwindigkeit der XEBEC Platte können Sie in der übersichtlichen Tabelle nachlesen.

Im Lieferumfang befindet sich neben dem Controller und der Festplatte eine Workbench Diskette auf der sich ein Programm für das Parken der Platte befindet, welches nicht benötigt wird, ein Backup Programm, das den Inhalt der Festplatte auf Diskette sichert, sowie zwei Programme für die Hardformatierung und Softformatierung, sowie ein Manual. Betreffend der Formatierung sollte man wissen, daß Festplatten vor Benutzung zweimal formatiert werden müssen. Zunächst die Hardformatierung (manchmal vom Hersteller schon vorgenommen), die für den Controller bestimmt ist und danach die Softformatierung, welche die

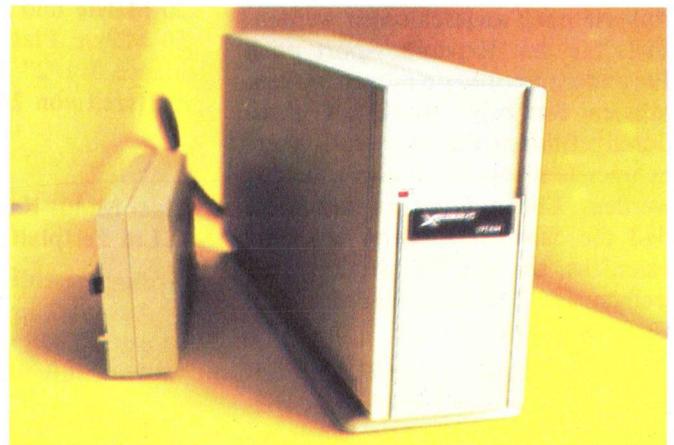


Bild 1: Die 20 MByte XEBEC Festplatte hat ein eigenwilliges Design.

Platte erst im eigentlichen AmigaDOS Format beschreibt. Danach können Programme auf die Festplatte kopiert und von dort aus aufgerufen werden. Ein solcher Formatierungsvorgang braucht Zeit (etwa 1 Std.), bedenkt man, daß etwa 600 Zylinder (je nach Festplattengröße) formatiert werden müssen. Eine Diskette hingegen besitzt nur 80 Zylinder. Der Preis der XEBEC Festplatte liegt bei 1385,- DM und ist damit für Amiga Platten recht günstig.

Beziehen kann man die XEBEX Festplatte bei folgendem Händler:

CAS, Tel. 069/842013

Plus-, Minuskasten der XEBEX Festplatte

- + einfache Installation
- + geringer Platzbedarf
- + Autoparking
- + solides Festplattengehäuse

- Konverter kein durchgeführten Bus
- externer Konverter

33 MByte !

Eine 50 MByte Festplatte aus dem Hause C.Ltd stand uns weiterhin zum Test zur Verfügung (siehe Bild 2). Hingegen älteren Modellen hat sich das Aussehen der Festplatte verändert. Etwas kompakter ist das Gerät verpackt, wobei das alte Metallgehäuse durch ein stabiles Plastikgehäuse ersetzt worden ist. Die Arbeit verrichtet, wie in der XEBEC Festplatte, ein 5 1/4" Laufwerk. Der Konverter ist ebenfalls ausgelagert und wird am Expansionsport angeschlossen. Beim C.Ltd Konverter ist der Bus durchgeschleift, dadurch können weitere Festplatten, Speichererweiterungen oder anderweitige Peripherie noch angeschlossen werden. Falls Sie den Expansionsport noch für andere Peripherie nutzen wollen, müssen Sie beim Kauf auf einen durchgeführten Bus achten, unangenehmes Umstöpseln kann vermieden werden. Ebenfalls ist zu beachten, daß die Stromversorgung am Port ausreicht, um alle Peripherie zu betreiben. Ein Test beim Händler kann Gewissheit bringen.

Der Geräuschpegel der C.Ltd Festplatte ist gering und stört nicht sonderlich bei der Arbeit. Die Arbeitsge-

Bild 2: Die 50 MByte CLTD Festplatte ist gegenüber der XEBEC Platte flach gehalten.



windigkeit kann ebenfalls der abgedruckten Tabelle entnommen werden. Ebenfalls ist die Platte mit Autoparking ausgerüstet, was vor Zerstörung einzelner Sektoren schützen kann. Weiterhin besteht die Möglichkeit, 14 weitere Festplatten anzusteuern.

Im Lieferumfang befindet sich neben dem Gerät und dem Kontroller eine Boot-Diskette, auf der sich die Programme zum Formatieren der Festplatte befinden. Außerdem sind noch etliche Textfiles über den Gebrauch und Handhabung des Speicherriesen auf der Diskette enthalten. Der Preis für die 33 MByte Festplatte liegt bei 1895,- DM und ist bei nachstehenden Händlern zu beziehen.

IM, Tel.: 069/7071102 und DTM, Tel.:06121/560084

Die Firma C.Ltd vertreibt den Festplatten Kontroller auch einzeln, unter selbigem Händler wie die Platte, ist der 4000A Kontroller für 295,- DM zu beziehen. Im Angebot von C.Ltd stehen noch weitere Festplatten mit 20 MByte und 50 MByte, wobei die 50 MByte Platte eine High Performance 3 1/2" Platte mit einer Zugriffszeit von 28 ms ist.

Plus- Minus-Kasten der C.Ltd Festplatte

- + durchgeschleifter Bus
- + Autoparking
- + neues DOS (Interleave Faktor 3)
- + Anschluß von 14 weiteren Festplatten
- externer Konverter

MS-DOS und AmigaDOS

Die dritte Festplatte in unseren Test ist für den Einbau in den Amiga 2000, bzw. der Sidecar des Amiga 1000 bestimmt, allerdings muß beim 2000er eine PC-Karte zum Betrieb vorhanden sein. Die Filecard mit 50 MByte Speicherkapazität stammt aus dem Hause Mountain (siehe Bild 3) und wird einfach in einen leeren Slot des Amigas bzw. der Sidecar gesteckt und fertig ist der Einbau. Bei unserem Testgerät mußte man etwas herumhebeln beim Einbau in den Amiga 2000, bis der Speicherriese fest verankert im Slot steckte. Doch das stellte kein größeres Problem dar. Ebenfalls verdeckt die Platte noch 1 1/2 freie Slots, sodaß diese nicht mehr genutzt werden können, was sich allerdings nicht vermeiden läßt.

Die Besonderheit der Filecard besteht nun darin, daß sie sowohl von AmigaDOS, als auch von MS-DOS aus genutzt werden kann. Besitzt man also ein Bridgeboard (PC-Platine) oder ein Sidecar, ist es möglich, die Festplatte in zwei Partitionen aufzuteilen, eine für AmigaDOS und die andere für MS-DOS. Unsere Filecard wurde so formatiert, daß dem Benutzer 30MByte von AmigaDOS und 20 MByte von MS-DOS zur Verfügung standen.

Die Mountain Filecard gibt kaum Arbeitsgeräusche von sich. Autoparking ist die Platte ebenfalls, sodaß keine Beschädigungen beim Transport unterlaufen können.

Die Zugriffszeiten sind zwischen den beiden Betriebssystemen unterschiedlich, beispielsweise benötigt die Filecard für das 100 KByte lange AmigaBASIC 5 Sekunden, hingegen



Bild 3: 50 MByte besitzt die Mountain Filecard.

- + Partitionen unter AmigaDOS und MS-DOS
- + sehr schneller Zugriff unter MS-DOS
- + günstiger Preis
- + Filecard Controller kann zwei Laufwerke ansteuern
- verdeckt freie Slots
- relativ lange Wartezeit beim Hochfahren der Filecard

Geschwindigkeit !

Die Geschwindigkeiten der Platten in Bezug auf die RAM-Disk und einer normalen Diskette können Sie der abgedruckten Tabelle entnehmen. Hierbei wurden verschiedenen Verfahren verwendet. Die erste Zeitmessung wurde beim Kopieren von AmigaBASIC (101 KByte) vorgenommen und zwar von der RAM-Disk zur Festplatte, die zweite Messung kopiert AmigaBASIC von der Platte zur RAM-Disk. Weiterhin wurde die Ladezeit des BASICs gestoppt und die Ausgabe des Inhaltes einer Amiga 2000 Workbench von der grafikorientierten Oberfläche, der Workbench.

Zusätzlich wurde noch eine Zeit der Mountain Filecard im Betrieb unter MS-DOS gemessen.

Fazit:

Festplatten haben ohne Frage ihren Reiz, durch den schnelleren Zugriff ist das Arbeiten beispielsweise mit Compilern ein Genuß. Auch die große Speicherkapazität ist das zweite große Plus, wer viele Daten verwalten muß, kommt um den Gebrauch einer Festplatte nicht herum.

(AK)

im Betrieb unter MS-DOS vergehen 2 Sekunden für ein gleichgroßes Programm. Diese Zeitunterschiede liegen darin begründet, daß die Amiga-daten erst einen Umweg über eine Emulator Karte machen müssen, was einige Zeit in Anspruch nimmt. Um die schnellste Zugriffszeit unter AmigaDOS zu erreichen, mußte der Befehl 'Addbuffers' herangezogen werden. Dieser vergrößert den Puffer über den die Daten wandern. Mit 'Addbuffers jh0:30' wurde die schnellste Zugriffszeit gemessen. 'jh0' stellt die Geräteadresse dar, '30' gibt hierbei die Größe des zusätzlichen Puffers an, wobei es sich jeweils um Sektoren (512 Byte) handelt, demnach um eine Erweiterung um 30 Sektoren. Wenn Sie diesen Befehl jetzt in Ihrem AmigaDOS Handbuch suchen, werden Sie keinen Erfolg haben, denn dieses Kommando ist ein neu hinzugekommener Befehl unter der Version 1.2. Leider sind diese Befehle im Handbuch noch nicht aufgeführt. Mittlerweile ist ein englischsprachiges Werk mit den neuen AmigaDOS Befehlen erschienen. Selbstverständlich ist es möglich, auch die andere Festplatte mit dem 'Addbuffers' Befehl etwas zu beschleunigen. Da die Mountain Platte eigentlich nur für den PC Bereich entwickelt wurde, befindet sich neben der Filecard nur ein Manual, das auf den Betrieb im PC Bereich ausgelegt ist. Erklärungen über den Amiga Betrieb wurden gänzlich vergessen. Der Händler sollte hier zu Rate gezogen werden, da ein Laie Schwierigkeiten bekommen wird, die Filecard ohne Handbuch zu initialisieren.

Die Filecard spricht nicht, wie die Festplatten von XEBEC und CLTD, auf den Gerätenamen 'dh0' an, son-

dern nur auf 'jh0'. Der Unterschied zwischen den beiden Namen besteht darin, daß 'jh0' über die PC-Karte arbeitet, was eine längere Hochlaufzeit nach sich zieht. 'jmount' bindet die Filecard, 'mount dh0' die XEBEC und C.Ltd an den Amiga. Diese letztere Form der Einbindung geht schneller vonstatten.

Ein Augenmerk sollte man auch darauf werfen, daß der Controller der Filecard es erlaubt, zwei Platten anzusprechen. Eine weitere braucht deshalb keinen Controller zu besitzen, was sich kann diese Platte wärmstens empfohlen werden. Unter AmigaDOS liegen die Zugriffszeiten höher, was aber generell am AmigaDOS liegt. Trotzdem kann sie auch für den Gebrauch unter AmigaDOS empfohlen werden. Ein gleichzeitiges Nutzen unter beiden Systemen ist möglich, was die Sache sehr interessant gestaltet.

Der Preis liegt mit 2100,- DM für die 50 MByte Filecard recht günstig.

Folgender Händler hat diese Filecard in seinem Verkaufssortiment:

PDC, Tel.: 06172/24748

	XEBEC	C.Ltd	Mountain	Diskette	RAM
AmigaBASIC RAM -> Festplatte	8,85	4,87	10,51	25,21	--
AmigaBASIC Festplatte -> RAM	4,48	4,56	6,28	10,97	--
AmigaBASIC laden	3,68	3,21	6,28	10,15	1,66
Amiga 2000 WB Direktory laden	1,91	1,85	1,13	8,69	1,02
Moutain unter MS-DOS GW-BASIC (78 KB) laden	1,59				
Alle Zeitmessungen in Sekunden					



APPRENTICE!!! Langsam, aber sicher drängt eine Vielzahl von Animationsprogrammen für den Amiga auf den Markt, die seinen Anspruch als semiprofessionellen Grafikrechner untermauern. Zu diesen Programmen zählt der Apprentice Animator, der sich von seinem Konzept erheblich von anderen Animationsprogrammen unterscheidet. Grund genug, dieses Programm-Paket einmal unter die Lupe zu nehmen.

Das Apprentice-Konzept

Normalerweise erwartet man bei einem Animationsprogramm, zumindest auf Mikrocomputern, Darstellungen streng geometrischer Objekte. Hier tanzt der Apprentice Animator aus der Reihe. Seine Zielsetzung besteht im Animieren von Trickfiguren, die als dreidimensionale Modelle erstellt werden, was der Autor Martin Hash als „organische Animation“ bezeichnet, im Gegensatz zu herkömmlichen „mechanischen Animationen“. So ist es mit diesem Programm möglich, eine Figur (beispielsweise einen Mensch oder eine Fantasiefigur) zu erstellen und anschließend die unterschiedlichsten Bewegungen vollführen zu lassen, die um so natürlicher wirken, je differenzierter man seine Figur gestaltet hat. In dieser Hinsicht

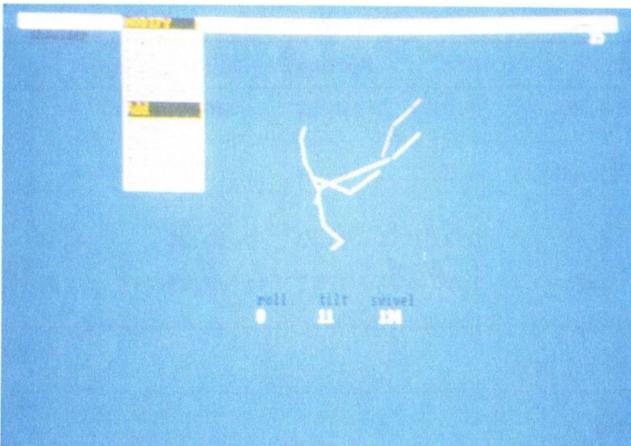
bietet es von anderen Animationsprogrammen ungekannte Möglichkeiten, dafür aber Einschränkungen bei der Animation einfacher geometrischer Objekte.

Zum Lieferumfang von Apprentice gehören in der vorliegenden Version drei Disketten und ein englischsprachiges Handbuch, welches – um dies vorwegzunehmen – leider etwas knapp gehalten wurde und mit Beispielen geizt. Bei den drei Disketten handelt es sich um die MasterDisk, die DataDisk und die MatteDisk. Die MasterDisk enthält sieben einzelne Module, mittels derer man seine Charaktere und Choreographien schrittweise erstellt. Die DataDisk dient zum Ablegen der verschiedenen Dateien, die man mit den Modulen als Zwischenschritte erstellt, und auf der

MatteDisk liegen die sogenannten Mattes, bei denen es sich um IFF-Bilder handelt und die das Modul Sculpt zur Oberflächengestaltung der Figuren heranzieht. Zusätzlich muß man sich, um einen Film zu erzeugen, eine FramesDisk formatieren. Diskettenwechsel bleiben also auch den Besitzern von zwei Floppylaufwerken nicht erspart. Auch gibt sich Apprentice nicht mit 512 Kb zufrieden; ein Megabyte Speicher ist zu seinem Betrieb Mindestvoraussetzung. Wer das Programm intensiv nutzen möchte, sollte jedoch über 1,5 Mb oder mehr und zwei Laufwerke verfügen.

Schritt für Schritt...

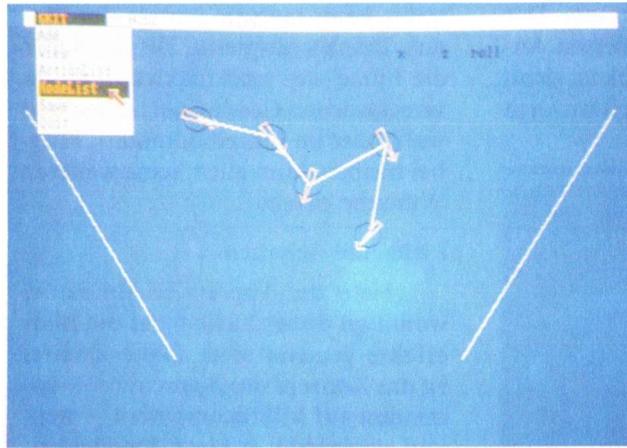
... geht es bei der Erstellung einer Animation mit Apprentice durch die Module Charakter, Action, Director und Rehearsal. Charakter dient zum Erstellen einer Grundstruktur einer Figur. Hierbei werden einzelne Körperteile als „Bones“ definiert und in einer speziellen Ordnung zusammengesetzt, die die Abhängigkeit eines Körperteils von einem oder mehreren anderen festlegt, wie z.B. die Abhängigkeit der Oberarme eines Menschen vom Torso, der Unterarme von den Oberarmen usw. Charakter bietet hierbei viele Edierfunktionen, die man auch intensiv nutzen muß, bis das gewünschte Ergebnis vorliegt,



Laß die Puppen tanzen: Ein Strichmännchen wird animiert

welches auf der DataDisk abgelegt und im nächsten Schritt in das Modul Action eingeladen wird. Dieses dient zur Erstellung einer Bewegung. Der Anwender hat die Möglichkeit, seiner Figur, die als Strichmännchen zu sehen ist, die mannigfaltigsten Bewegungen „beizubringen“. Action arbeitet sehr komfortabel; es interpoliert Bewegungen, d. h., man muß nur den Start- und den Endpunkt einer Bewegung und die gewünschte Anzahl der dazwischenliegenden Bilder eingeben, es versorgt den Anwender für jedes Körperteil seiner Figur mit einer Action List, in welcher die bis dahin festgelegten Bewegungen festgehalten werden, und verwöhnt mit Delete-, Insert- und Modifyfunktionen. Auch verfügt es über Ease-Funktionen für sanftes, natürlich wirkendes Ein- und Ausschwingen von Bewegungen. In Action wird auch die in Charakter festgelegte Ordnung einer Figur sehr anschaulich; läßt man einen Mensch beispielsweise die Oberarme schwingen, so bewegen sich Unterarme und Hände mit diesen, können aber auch für sich noch Bewegungen ausführen (wie z. B. ein Anwinkeln der Unterarme beim Schwingen der Oberarme). Man erstellt die Bewegung einer ganzen Figur, indem man nacheinander die Bewegungsschritte der einzelnen Körperteile definiert. Hat man einen gewünschten Bewegungsablauf gestaltet, wird dieser auf Disk gespeichert, und ab geht es in das Modul Director.

Hier darf man Regisseur sein, seine „Darsteller“ plazieren, ihnen Bewegungsabläufe diktieren, Szenenabläufe festlegen, Kamerafahrten definieren, die Beleuchtung einer Szene arrangieren und vieles mehr. Den Auswahl- und Gestaltungsmöglichkeiten einer Szene sind hierbei kaum Grenzen gesetzt, und wie schon im Modul Action erweist sich die Bedienung – nach kurzem Studium des Handbuchs und ein paar Versuchen – als ausgesprochen komfortabel. Ein paar kleine Wermutstropfen in der Bedienung sind lediglich die etwas einfachen Requester, bei denen man oftmals gezwungen ist, sowohl die Tastatur als auch die Maus zu benutzen, was ein bißchen umständlich ist; hier hätte sich der Entwickler besser auf reine Maus- oder Tastatureingabe festlegen sollen, was sowohl für



**Im Modul Director:
Eine Bühnenanweisung für den „Darsteller“**

Action als auch für Director gilt.

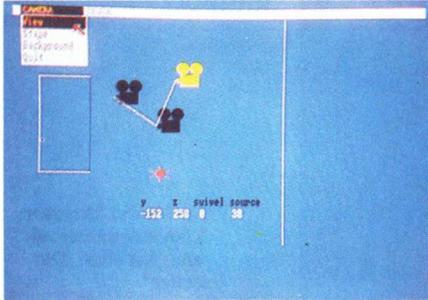
Das Ergebnis seiner Arbeit mit Director speichert man als Choreographie. Diese kann man im letzten Schritt in das Modul Rehearsal einladen und abspielen lassen, wobei die komplette Szene aus der Kamerasicht abläuft, allerdings noch nicht mit einer gestalteten Figur, sondern nur mit deren „Skelett“, also einer Strichfigur. Dennoch ist Rehearsal eine sehr gute Möglichkeit, zu überprüfen, ob die Szene auch wirklich den geplanten Verlauf nimmt. Mit den drei Modulen Charakter, Action und Director kann man übrigens auch bereits früher gestaltete Figuren, Bewegungsabläufe oder Choreographien einladen und modifizieren.

Zeichenkünstler gefragt...

...heißt es beim Objekteditor Sculpt (dieser hat nichts zu tun mit dem gleichnamigen Ray-Tracing-Programm). In diesem Modul wird die eigentliche Gestaltung einer Figur vorgenommen. Diese Arbeit erweist sich allerdings als nicht so einfach wie das Arbeiten mit den vorher erwähnten Modulen. Sculpt bezieht die Information für die Oberflächengestaltung eines Körperteils nämlich aus Bildern im IFF-Format, sogenannten Mattes, die man vor der Konstruktion selbst erstellen muß, was voraussetzt, daß man über ein Malprogramm verfügt, das seine Bilder IFF-Format ablegt, wie zum Beispiel DeluxePaint oder auch DigiView. Will man beispielsweise den Kopf eines Menschen gestalten, so steht man vor der Aufgabe, diesen in allen Ansichten zu zeichnen, also von vorne, hinten, links, rechts, oben und unten, wobei man genau darauf

achten muß, daß der Kopf sowohl in Vorder-, Hinten- und Seitenansicht genau die gleiche Höhe in Pixeln hat, genauso muß sich aus der Seitenansicht und der Draufsicht die gleiche Tiefe des Körpers ergeben. Diese Arbeit ist recht langwierig und kompliziert. Zusätzlich benötigt man noch die deckungsgleichen Profile- und Visage-Mattes, aus denen Sculpt erst einmal die dreidimensionale Form eines Körpers erzeugt, über die danach die Mattes der verschiedenen Ansichten gespannt werden. Die dreidimensionale Form des Objekts setzt sich aus übereinanderliegenden Querschnitten, sogenannten „Slices“, zusammen, welche aus vielen, kleinen Flächen bestehen. Diese Slices kann man von Hand modifizieren. Auch bietet Sculpt eine Funktion namens Autosculpt, die sehr hilfreich beim Erstellen bestimmter Formen ist, aber nur begrenzte Möglichkeiten bietet. Daneben existiert eine Anzahl Hilfsfunktionen zur Perfektionierung und zum Ausgleich eventuell auftretender Ungenauigkeiten. Man hat mit dem in Sculpt angewandten Verfahren und seinen Funktionen zwar sehr weitreichende Gestaltungsmöglichkeiten, muß sich aber auch eingehend damit beschäftigen. Aufgrund der Komplexität des Gestaltungsprinzips ist es praktisch unmöglich, einen Körper „mal eben auf die Schnelle“ zu entwerfen; so ist es mir während der Testzeit nicht gelungen, eine relativ einfache Figur annehmbar zu gestalten. Wer eine komplexe Figur erzeugen möchte, beispielsweise einen Mensch mit bis ins Detail ausgeführten Gliedmaßen (einzelne Finger, beweglicher Mund, Augenlider etc.), der sollte sich doch zumindest

einen Tag dafür frei nehmen. Das sollte den wirklich interessierten Anwender aber nicht abschrecken, denn die Resultate sind dann auch wirklich überzeugend.



Dem Regisseur ist nichts zu schwör:
Auch Kamerafahrten werden in Director festgelegt

Achtung, Aufnahme!

Ein derart beschriftetes Schild sollte man sich zulegen, will man vermeiden, daß wohlmeinende Mitbewohner im Glauben, man habe vergessen, den Amiga abzuschalten, das Modul Record durch Abschalten des Rechners in der Arbeit unterbrechen, denn es ist ratsam, die Aufnahme eines Films in Abwesenheit durchzuführen. Berechnungen einzelner Bilder erfordern zwischen 5 und 20 Minuten Rechenzeit, und da man sehr schnell einmal auf 80 oder 100 Bilder für eine Sequenz kommt, kann das für den Rechner in eine lange Nacht ausarten. Grund dafür ist, daß die Objekte, wie schon vorher erwähnt, aus sehr vielen winzigen Rechtecken bestehen, die bisweilen bis auf Pixelgröße reduziert sind. Dadurch wird allerdings auch eine sehr echt wirkende Schattierung der Objekte ermöglicht, die man optional aktivieren kann. Etwas lästig ist, daß Record während der mehrstündigen Arbeit fast ununterbrochen auf die DataDisk zugreift; wer ausreichend Speicher zur Verfügung hat, sollte sich also im RAM eine Directory „Data“ anlegen, den Inhalt der DataDisk hineinkopieren und den Zugriff des Programms mittels Assign umlenken, was allerdings etwas CLI-Arbeit für den Anwender bedeutet. Wer nur ein Laufwerk besitzt, muß, da Record nach jedem Bild auf die FramesDisk zugreift, so vorgehen, oder während der Aufnahme anwesend sein und nach jedem Bild die Disks wechseln.

Ist die Animation schließlich er-

stellt, kann man sie mittels des Moduls Display abspielen. Display spielt die Filme vor- und rückwärts ab, in verschiedenen Geschwindigkeiten und wahlweise im Einzelbildmodus. Hierbei bleiben dann auch keine weiteren Wünsche offen.

Licht und Schatten...

...bietet der Apprentice Animator, womit an dieser Stelle nicht die Bild-effekte gemeint sind. Ohne Zweifel ist das Konzept des Apprentice – zumindest auf Mikrocomputern – neu und einzigartig; es bietet Möglichkeiten, die mit anderen Animationsprogrammen nicht zu verwirklichen sind, die erreichbaren Ergebnisse sind faszinierend, sowohl von der Art der Animation her gesehen als auch von der Ablaufgeschwindigkeit. Dies unterstreicht den semiprofessionellen



Das Ergebnis: Im Laufschrift hechtet unser Mann über den Bildschirm

Anspruch dieses Programmpakets. Das Prinzip der Bilderzeugung ist sehr gelungen; durch die angewandte Technik und die Unterteilung der Figuren in äußerst kleine Flächen entstehen sehr genaue Licht-, Schatten- und Perspektiveffekte, die teilweise in ihrer Qualität an Ray-Tracing-Bilder erinnern, allerdings bedeutend weniger Rechenzeit in Anspruch nehmen. Man muß allerdings auch bereit sein, viel Zeit in intensive Arbeit zu investieren, um selbst gestaltend und kreativ tätig zu werden. Ein gravierender Nachteil, der die Möglichkeiten für wirklich professionelle Animation stark einschränkt, ist die Festlegung des Programms auf die niedrige Auflösung ohne Interlace, was zwar freie Farbauswahl bis hin zum HAM-Modus erlaubt, des öfteren aber doch die Feinheiten einer Animation in der Bildschirmauflö-

sung untergehen läßt.

Laut Auskunft des deutschen Vertriebs ist inzwischen eine größere Anzahl DataDisks mit bereits erstellten Figuren erhältlich; auch eine umfangreiche deutsche Anleitung ist in Vorbereitung. Da der Preis des Apprentice Animators mit 595,- DM recht hoch angesiedelt ist, und auch die DataDisks mit 79,- bis 89,- DM zu Buche schlagen, ist dieses Programm wohl eher für ambitionierte Anwender, die sich mit semiprofessioneller Computeranimation und/oder Videoproduktion beschäftigen, geeignet, weniger aber für den Anwender, der sich ein Programm „just for fun“ zulegen möchte. Ob die niedrige Grafikauflösung für interessierte Anwender einen bedeutenden Mangel darstellt, sollten diese vor dem Kauf prüfen, was, in Anbe-

tracht der Auskunft des Vertriebs, daß bereits zahlreiche Demodisketten mit erstellten Animationen zur Verfügung stehen, kein Problem darstellen sollte.

(WD)

- + neuartiges Konzept mit außergewöhnlichen Möglichkeiten
- + gut unterteilte Erstellung der Animationen in Einzelschritten
- + viele Korrekturfunktionen
- + viele Kombinationsmöglichkeiten

- Erstellen eigener Figuren sehr arbeitsintensiv
- Animationen nur im LowResolution-NonInterlaced-Modus möglich

Preis: 595,-

Anbieter: DTM

AMIGA SOFTWARE ZUM ABHEBEN

SPIELE

A Mind Forever Voyaging	DM89,00
Adventure Constr. Set	DM79,00
Amiga Karate	DM65,00
Alien Fires	DM 89,00
Arazok's Tomb	DM 89,00
Archon I	DM 79,00
Archon II	DM 79,00
Arcticfox	DM 69,00
Arena	DM 79,00
Balance of Power	DM 89,00
Bard's Tale	DM 97,00
Barbarian	DM 79,00
Beat it	DM 29,95
Bio-Timer	DM 69,00
Borrowed Time	DM 79,00
Brataccas	DM 79,00
Bureaucrazy	DM 96,00
Ch.Basketb. Two on Two	DM 79,00
Championship Baseball	DM 79,00
Championship Golf	DM 79,00
Chessmaster 2000	DM 88,50
Deep Space	DM 88,50
Defenders of the Crown	DM 87,50
Déjà Vu	DM 88,50
Dr. Fruit	DM 29,95
Faery Tale	DM 98,00
Flightsimulator II	DM118,00
Scenerys Disk Flightsim.II	DM 59,00
Flip Flop	DM 29,95
Fußball Manager deut.	DM 72,00
Galileo	DM 215,00
Garrison	DM 68,00
Goldrunner	DM 78,00
Grand Slam Tennis	DM 79,00
Guild of Thieves	DM 89,00
Hacker I	DM 69,00
Hacker II	DM 69,00
Halley Project	DM 69,00
Hollywood Hijinx	DM 89,00
Hollywood Poker	DM 59,00
Jewels of Darkness	DM 48,00
Karate King	DM 48,00
Karate Kid II	DM 64,00
Leaderboard Golf	DM 74,90
Leaderboard Tournament	DM 47,00
Little Computer People	DM 89,00
Marble Madness	DM 68,00
Mind Breaker	DM 29,95
Mindshadow	DM 69,00
Mission Elevator	DM 58,00
Moonmist	DM 86,00
One on One	DM 98,00
Pac Boy	DM 29,95
Planefall	DM 99,00
Portal	DM 89,00
Programm des Leben	DM 155,00
Quintette	DM 99,00
Quiwi	DM 48,00
Racter	DM 99,00
Rocket Attack	DM 29,95
Seven Cities of Gold	DM 89,00
Shanghai	DM 79,00
Silicon Dreams	DM 69,00
Sinbad and t. t. of t. falcon	DM 87,00
Sky Fighter	DM 59,00
Sorcerer	DM 99,00
Space Fighter	DM 29,95
Starglider	DM 79,00
Strip-Poker	DM 79,00
Swooper	DM 48,00
Surgeon	DM 149,00
S. Huey, Heli. Flight. Sim.	DM 67,00
Tass Times In Tonetown	DM 75,00
Temple of Apschai Trilogy	DM 78,00
The Pawn	DM 75,00
The Witness	DM 99,00
The final trip	DM 29,95
Ultima III	DM 99,00
Uninvited	DM 89,00
Vader	DM 29,95
Winter Games	DM 64,00
World Games	DM 64,00



Software Verlag GmbH • 8000 München 19
Horemansstraße 2

Das Jumbo-Angebot des Monats!

Magic Forms 3D-CAD DM 298,-
Konstruktion komplexer dreidimensionaler Objekte;
betrachtung aus verschiedenen Blickwinkeln;

Barbarian DM 79,-
Actionspiel mit Supergrafik und -sound!

GRAFIK

Animator + Images	DM 275,00
Art & Utility Disk I	DM 65,00
Art Disk Print I	DM 65,00
Art Pak I Aegis	DM 65,00
Art Parts Volume 2	DM 68,00
Butcher V2.0	DM 89,00
Calligrapher	DM 218,00
Deluxe Paint II	DM 239,00
Deluxe Print	DM 139,00
Deluxe Video V1.2	DM 228,00
Digi View	DM 465,00
Digi Paint	DM 152,00
Draw	DM 239,00
Draw plus	DM 479,00
Dynamic CAD	DM 998,00
Impact	DM 199,00
Magic 3D CAD	DM 298,00
Sculpt 3D	DM 178,00
Page Setter (Uml.)	DM 349,00
Laserscript f. Page Setter	DM 99,00
Prism	DM 159,00
Title Construction Set	DM 95,00
TV-Text 3D	DM 245,00
Videoscape 3D	DM 349,00
Zuma Fonts 1	DM 69,00
Zuma Fonts 2	DM 69,00
Zuma Fonts 3	DM 69,00

MUSIK

Deluxe Music Constr.	DM 229,00
Future Sound	DM 395,00
Instant Music	DM 79,00
It's just Rock'n'Roll	DM 69,00
Midi-Interface	DM 99,00
Music Studio	DM 99,00
Pro Midi Studio	DM 349,00
Sonix	DM 178,00
Sound Sampler	DM 219,00

KOMMERZIELL

Analyze V 2.0	DM 349,00
BBS Bulletin Board	DM 210,00
dBMan	DM 345,00
Flow	DM 225,00
Logistix (deutsch) neue V	DM 329,00
Logic Works	DM 398,00
OnLine!	DM 155,00
Organize!	DM 228,00
Pro Write	DM 289,00
Publisher 1000	DM 459,00
Pagesetter (Uml.)	DM 349,00
Scribble 2.0	DM 219,00
Softwood File II GS	DM 229,00
Superbase (deutsch)	DM 245,00
UBM-Text V2.2 (deutsch)	DM 229,00
Vip Professional (engl.)	DM 329,00
Word Perfect	DM 699,00

SPRACHEN, TOOLS

AC Basic	DM 412,00
Amiga C Compiler	DM 479,00
C-Monitor V. 2.00	DM 195,00
Cambridge Lisp	DM 479,00
CLi mate	DM 78,00
Fortran 77	DM 649,00
Gizmoz V2.0	DM 155,00
Grabbit	DM 69,00
Lattice C Compiler V3.1	DM 438,00
M2 Amiga Modula2	DM 340,00
Macro Assembler	DM 179,00
Manx Aztec C Com.	DM 1120,00
Manx Aztec C Devel.	DM 648,00
Manx Aztec C Personal	DM 445,00
MCC Pascal	DM 248,00
Modula-2 Commercial	DM 619,00
Modula-2 Developers	DM 349,00
Modula-2 Standard	DM 219,00
Multi Forth-83	DM 578,00
Shell CLi	DM 118,00
Toolkit	DM 99,00
True Basic Interpreter	DM 349,00
UCSD-Pascal	DM 348,00
Zing! (CLi deluxe)	DM 179,00

BÜCHER

Amiga 500 für Einsteiger	DM 39,00
Amiga Basic	DM 59,00
Amiga für Einsteiger	DM 49,00
Amiga Maschinensprache	DM 49,00
Amiga Tips und Tricks	DM 49,00
Amiga DOS-Manual	DM 79,90
Amiga DOS Express	DM 79,90
Amiga Progr. Handbuch	DM 69,00
C für Einsteiger (Amiga)	DM 39,00
Das k. Amiga 500 & 2000	DM 29,00
Das Amiga Handbuch	DM 49,00
Das Amiga Grafik Buch	DM 49,00
Grafik auf dem Amiga	DM 49,00
Hardware Ref. Manual	DM 62,50
Intuition Ref. Manual	DM 62,50
Komment. ROM-Listing (cl.)	DM 69,00
Programmers Guide	DM 50,80
Programmers Handbook	DM 99,90
Progr. m. Amiga Basic	DM 59,00
ROM-Kernel Ref.: exec	DM 62,50
ROM-K. Libr. & Devices	DM 88,00
Amiga 500 Schaltpläne	DM 60,00
Amiga 1000 Schaltpläne	DM 70,00
Amiga 2000 Schaltpläne	DM 110,00
Sidecar Schaltpläne	DM 40,00

DEMO DISKS

Draw	DM 12,00
Animator	DM 12,00
Zuma Fonts	DM 12,00
TV Text	DM 12,00
Bard's Tale	DM 12,00
Zing!	DM 12,00
Dynamic CAD	DM 12,00
Digi-View (H.A.M.)	DM 12,00
Perfect Sound	DM 12,00
Sculpt 3D (2 Disks)	DM 24,00
Title Const. Set	DM 12,00
Modula	DM 12,00
dBMan	DM 12,00
Vizawrite	DM 12,00
Logistix	DM 12,00

ACHTUNG! Kein Parteienverkehr!
Versand täglich ab Großversandlager!

Preisänderungen vorbehalten.
Lieferung nach Verfügbarkeit.

Name _____

Vorname _____

Straße _____

PLZ/Ort _____

Hiermit bestelle ich:

Ich zahle per:
beiliegendem
Verrechnungsscheck

Nachnahme(zuzügl.
N.N.-Gebühren)

zuzügl. Versandkosten von DM 6,50

Jumbo Soft
Software Verlag GmbH
Horemansstr.2, 8000 München 19
Tel. 089/1 23 40 65

PUBLIC DOMAIN SERVICE

Für den AMIGA gibt es schon eine Unmenge von Public-Domain-Programmen; manche Anbieter haben über 100 Disketten in Ihrem Programm. Die verschiedenen Sammlungen sind jedoch zum Teil nicht sortiert und in sich sehr unübersichtlich. Die Disketten wurden meist in den USA zusammengestellt (z.B. die 'Fish-Disks'), andere stammen aus den unterschiedlichsten Quellen. Dies hat dazu geführt, daß sich einige Programme auf mehreren Disketten wiederfinden. Hinzu kommt eine Vielzahl von Updates, also verbesserten Programmversionen, die die alten überflüssig machen. Ein weiterer Kritikpunkt an den meisten bestehenden Sammlungen ist die fehlende oder unzureichende Dokumentation, mit denen die Programmsammlungen angeboten werden. In den Listen steht oft nur eine Sammlung von Namen, deren Bedeutung jedoch nur selten deutlich wird. Fast immer fehlen Angaben über die Programmiersprache, den verwendeten Compiler und die notwendige Kickstartversion.

Aus diesem Dilemma soll Ihnen der Public-Domain-Service der KICKSTART helfen. Die Disketten enthalten ausschließlich auf ihre Funktionstüchtigkeit getestete Programme. Die einzelnen Disketten werden nach festen Kriterien zusammengestellt. Jede Diskette hat also einen Schwerpunkt (z.B. Lehrgänge, Bilder-Show, C-Programme, Spiele, u. ä.). Außerdem werden Angaben über die Programmiersprache, den verwendeten Interpreter oder Compiler usw. gemacht.

Diese Aktion, die uns einige schlaflose Nächte gekostet hat, soll Ordnung in die bestehende Public-Domain-Software bringen, damit Sie den größtmöglichen Nutzen daraus ziehen können. Wir hoffen, daß Sie mit der Einteilung und dem Ergebnis zufrieden sind und würden uns über Ihre Anregungen zu diesem Thema sehr freuen. Schreiben Sie uns, wenn Sie einen Verbesserungsvorschlag machen wollen! Wir haben dafür ein offenes Ohr und werden sicher einige dieser Vorschläge umsetzen.

Die Programme der Disketten 1-20 laufen auf allen AMIGA-Computern mit Kickstart/Workbench 1.2. Allerdings sollten 512k Speicher vorhanden sein. Sollten dennoch irgendwelche Einschränkungen gelten, werden wir dies bei den betreffenden Programmen angeben.

Das aktuelle Angebot

PUBLIC DOMAIN SERVICE

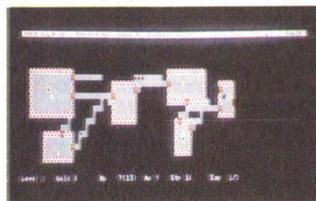
Diskette 1: C SOURCE

Eine Programmsammlung, die besonders dem Anfänger zeigt, wie man Intuition programmiert. Die Beispiele liegen als C-Quellcode und auch als fertige Programme vor, die sofort gestartet werden können.

Diskette 2: Spiele

- YachtC (Würfelspiel für 4 Personen)
- Puzzle
- Missile (verteidigen Sie Ihre Stadt, starker Sound)
- TriClops (sehr schönes 3D-Spiel)
- Breakout (3D-Effekt mit Brille)
- Trek73 (bekannte Star Trek Variante)

Diskette 3: Spiele



HACK: Das bekannte Textadventure, das ursprünglich auf UNIX-Rechnern erstellt wurde, liegt hier als spezielle Grafikversion für den Amiga vor.

Diskette 4: Terminal-Programme

KERMIT: Bekanntes, luxuriöses Terminalprogramm (drei verschiedenen Versionen, mit Source-Code in C).

Diskette 5: Terminal-Programme

- WOMBAT (VT102/52 Emulator, XModem, Autodial)
- VT100 (grafikfähig, Source in C)
- TermPlus (XModem, Source in C)
- DG210 (Data General D-210 Emulator)
- Ahost (XModem, Kermit)

- TeK4010 (XModem, VT100)

Diskette 6: Terminal-Programme

- Speech Term (spricht den empfangenen Text, XModem)
- StarTerm (mit Phone, Duplex, XModem)
- Argo Term
- PD Term (Source in C)
- AmigaDisplay
- Kermit

Diskette 7: UTILITIES

- QuickCopy (gutes Kopierprogramm)
- DirUtil (File-Copy)
- FileZap (File-Monitor)
- DiskZap (Disk-Monitor)
- DiskSalv (Diskettenretter)
- System-Monitor
- CSH (UNIX-ähnlicher Shell)

Diskette 8: Spiele



Monopoly: Sehr schöne Grafik, einfache Mausbedienung (Source in A-Basic).

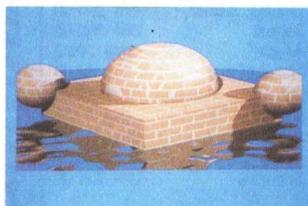
Diskette 9: Grafik-Show

- Grafik-Show mit bekannten Cartoons

Diskette 10: Grafik-Show

JUGGLER DEMO: Ein bewegliches Männchen jongliert mit drei verspielten Kugeln, sehr schöne Demo.

Diskette 11: Grafik-Show



RAY TRACERS: Wunderschöne räumliche Bilder, die auf einer VAX

berechnet und auf den AMIGA übertragen wurden.



Diskette 12: Grafik

- Digitalisierte Bilder mit erstaunlicher Qualität (IFF-Format)

Diskette 13: Grafik

- Sehr schöne Bilder-Show (IFF-Format)

Diskette 14: EDITOR

Bekannter Texteditor MICROEMACS Version 30. Viele Features: Search/Replace/Copy.

Diskette 15: Grafik-Animation

Verschiedene Filme, die mit dem AEGIS-ANIMATOR erstellt wurden, incl. PLAYER zum Abspielen der Filme. INFO: Einige Filme benötigen auf dem AMIGA 1000 mehr als 512 KB Speicher. Bei AMIGA 500/2000 mit 1 MB Speicher muß vorher 'No-FastMem' gestartet werden!

Diskette 16: Sprachen

XLISP 1.7 (neueste Version) mit ausführlicher Anleitung (über 50k)

Diskette 17: Sprachen

MODULA-2: Pre-Release eines Modula-Compilers mit verschiedenen kleineren Beispielprogrammen, die als Source-Code vorliegen.

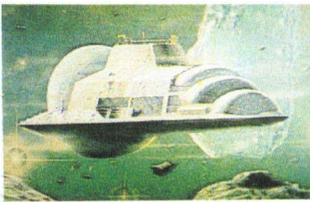
Diskette 18: Grafik



MANDELBROT

PUBLIC DOMAIN SERVICE

Diskette 19: Grafik-Show



– Sehr schöne digitalisierte H.A.M.-Bilder

Diskette 20: Grafik-Show

'Fred the Baker und Rose's Flower Shop': COMIC-Film, der die Multitasking-Fähigkeiten des AMIGA erklärt.

Diskette 21: AMIGA-Tutor

Einführung in die Bedienung des AMIGA 500. Ein farbenfroher Lehrgang, der ganz am Anfang beginnt und mit vielen Bildern und Grafiken die Grundbegriffe des AMIGA erklärt (für Anfänger, komplett in Deutsch).

Diskette 22: Sprachen

MVP-FORTH und C-FORTH
C-Forth ist ein recht leistungsfähiger FORTH-Interpreter, der auch als Quelltext vorliegt.

Diskette 23: Grafik-Show

Viele abwechslungsreiche Motive in verschiedenen Auflösungen, verpackt in einer Grafik-Show (startet automatisch!)

Diskette 24: Grafik-Show



Sehr schöne digitalisierte Frauengesichter (startet automatisch!)

Diskette 25: UTILITIES

CLOCK, PORTAR, MACView, Kickbench, Disassembler, Tracker, Checkmodem, POPCLI (nur Kickstart 1.1) und vieles mehr

Diskette 26/27: GRAFIK-SHOW

Auf zwei randvollen Disketten erleben Sie eine einmalige Dia-Show mit hervorragend digitalisierten futuristischen Bildern in voller PAL-Auflösung. Dazu gibt es stimmungsvolle, sphärische Musik.

Diskette 28: Editoren

Auf dieser Diskette befinden sich einige schöne Editoren (UEDIT, MED, BLITZ) mit dazugehörigen Zeichensatz-Utilities (UeTurbo, Blitzfonts, Pearlfonts).

Diskette 29: UTILITIES

PrtDrvGen: erstellt Drucker-Treiber
DropShadow: jedes Fenster bekommt einen Schatten
MemClear: löscht den Speicher
ScreenSave: speichert den Bildschirm auf Diskette
Compress: komprimiert Programme

Diskette 30: Sound-Demos

Changing Minds, Joan Lui, Miami Vice II, Respectable, Holiday

Diskette 31: SOUND-DEMO

Dieses Programm erzeugt naturgetreue Geräusche, die über die Tastatur, wie auf einem Klavier angespielt werden können.

Diskette 32: SOUND-DEMOS

Mit einer Demo-Version von SoundScape können digitalisierte Musikstücke abgespielt werden. Die Qualität ist wirklich erstaunlich!

Diskette 33: GRAFIK-SHOW



Einige sehr gute, mit Deluxe Video erstellte Filme. Der Benötigte PLAYER befindet sich mit auf der Diskette. INFO: bei AMIGA 500/2000 mit 1 MB Speicher erst 'NoFastMem' starten!

Diskette 34: SPIELE



TUNNEL VISION: werden Sie den Weg durch das Labyrinth finden?
REVERSI: eine spielstarke Version des bekannten Brettspiels
KLONDIKE: ein Patience-Kartenspiel

Diskette 35: UTILITIES

AHA-Copy V1.0 (Nibble- u. Parameter-Copy, IBM-Format, Track 0-81)
ASDG (resetfeste RAM-Disk)
FixDisk, ErrorCk (zur Fehlersuche auf der Diskette)
DiskCat (erstellt eine Übersicht über die Programme Ihrer Disketten)

Diskette 36: CAD

mCAD ist ein wirklich gut gemachtes CAD-Programm, daß jedoch nur im Interlace-Modus läuft. Es bietet die einfachen Zeichenfunktionen und Features wie Zoom, Group, Ungroup, Grid, Move, Rotate). Auf der Diskette befinden sich mehrere Dokumente, die das Programm erklären.

Diskette 37: UTILITES

neue Helfer für Programmierer

Diskette 38: GRAFIK

Ein neuer Mandelbrotgenerator

Diskette 39: GRAFIK-SHOW

neue Bilder

Diskette 40: GRAFIK-DEMOS

verschiedene gute Beispiele für bewegte Grafiken

Versandbedingungen:

Um einen schnellen und problemlosen Versand zu gewährleisten, beachten Sie bitte folgende Punkte:

- Für jede Diskette ergibt sich ein Unkostenbeitrag von DM 10, –.
- Pro Sendung kommt ein Versandkostenbetrag (für Porto und Verpackung) von DM 5, – (Ausland DM 10, –) hinzu.
- Bitte legen Sie außerdem einen Aufkleber mit Ihrer Adresse bei.

Für ausländische Besteller sei erwähnt, daß Sie mit EUROSCHECKS auch in DM bezahlen können.

Wichtig!

Die Software wird nicht auf 'No-name'-Disketten geliefert, sondern auf Qualitätsdisketten der Marke FUJI FILM, mit denen wir sehr gute Erfahrungen gemacht haben.

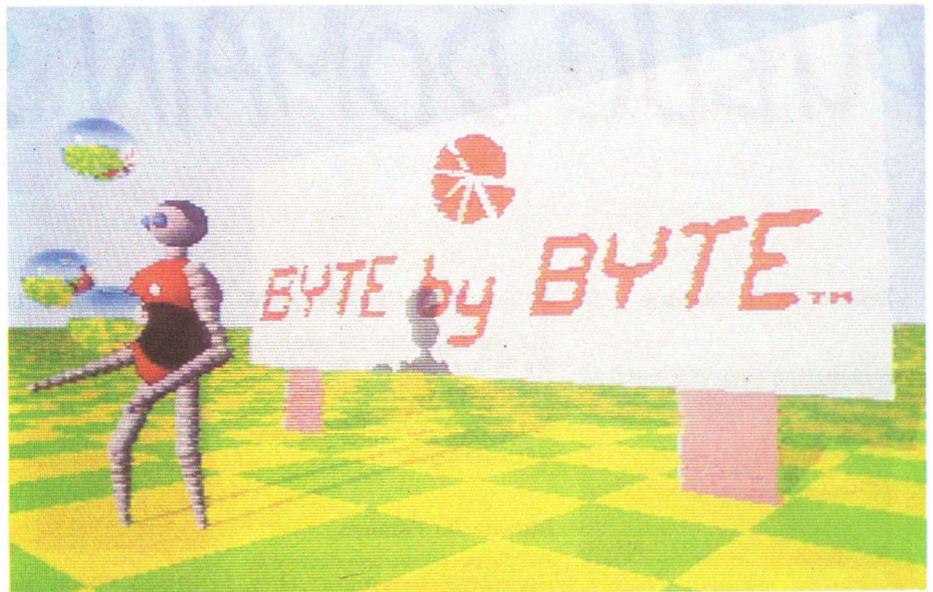
Anschrift:

MERLIN COMPUTER GmbH
KICKSTART Redaktion
Postfach 55 69
6236 Eschborn

Die Diskettenbestellung kann auch telefonisch erfolgen. Der Versand erfolgt dann per Nachnahme.
Tel.: 0 61 96 / 48 18 11 (Mo – Fr von 9 – 17 Uhr)

Die Graphischen Fähigkeiten des Amiga ermöglichen eine neue Generation von Anwender-Software, die bisher nur auf wesentlich größeren und vor allem teureren Systemen implementiert waren.

Ein Beispiel dafür ist 'Sculpt-3D' aus dem Hause 'Byte by Byte'. Sculpt-3D ist ein sogenanntes 'ray-tracing' Programm. Ray-trace heißt, bei der Erstellung eines Bildes aus 3-Dimensionalen Objekten, wird jeder einzelne Lichtstrahl berechnet, was zu sehr naturalistischen Grafiken führt. Die dafür erforderliche Rechenzeit ist lang, daher sei vorab gesagt, daß dieses Programm nicht zu schnellen Ergebnissen führt. Vielmehr ist es für jene gedacht, die sich schon ärgerten, daß Ihr Amiga während des Schla-



LICHTSPIELE

fens oder bei Abwesenheit untätig ist, denn die Berechnung eines Bildes kann viele Stunden dauern. Ray-Trace Demo Bilder gibt es auf Public-Domain Disketten u. a. auch über die KICKSTART. Diese meist auf Digital VAX Systemen berechneten Bilder benötigen auch auf jenem legendären Grafik-System stundenlange Berechnungsphasen. Sculpt-3D präsentiert sich in nicht allzu aufwendiger Aufmachung. In der vorliegenden Version 1.05, befindet sich im ca. 100seitigen DIN A5 Handbuch, welches nicht aus dem Englischen Übersetzt wurde, eine nahezu volle Diskette, und ein Warnschildchen mit der Aufschrift „CAUTION RAY TRACE IN PROCESS“. Damit kann man sich und andere vor dem Benutzen des Amigas warnen, während dieser die 'rays traced'. Die Diskette beinhaltet keine WorkBench, daher muß Sculpt-3D mit einer eigenen WorkBench gestartet werden. Die

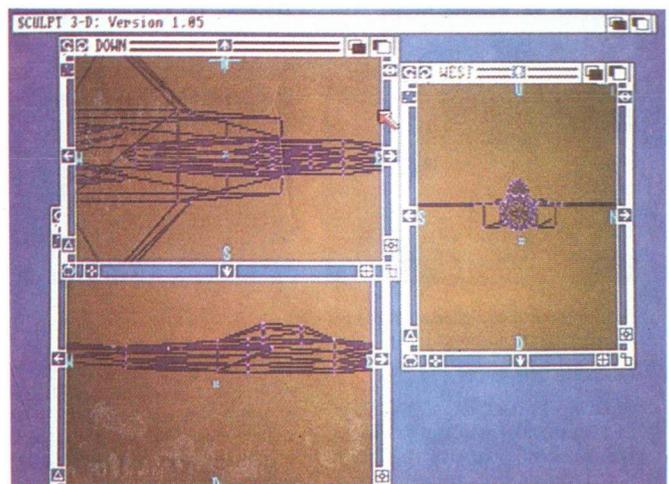
Workbench Diskette sollte nach dem Programmstart in Reichweite bleiben, da diverse 'libs' immer wieder nachgeladen werden muß.

Die Anzahl der Zugriffe auf die Programm Diskette hält sich danach zwar im Rahmen, dennoch sind zwei Laufwerke anzuraten, da weder eine komplette WorkBench noch die Sculpt-3D Diskette ausreichend Platz zum Speichern der eigenen Werke liefert. Dieser kleine Nachteil und das knapp gehaltene Handbuch, welches

zwar alle Funktionen gut dokumentiert, aber kein zusammenhängendes Beispiel für den recht komplexen Arbeitsvorgang liefert, sollen den Preis senken, der in Deutschland bei 198,- DM liegen wird.

Sculpt-3D wird nahezu vollständig mit der Maus bedient, nur die Diskettenoperationen, wie Speichern und Laden, erfordern die Tastatur zur Eingabe der Filenames.

Nach dem Programmstart befindet man sich im 'TRI-VIEW', einem 3-dimensionalen Object-Editor. Das Problem der 3D Objekt Erstellung wurde hier sehr komfortabel gelöst, wenngleich es auch nicht einfach ist.

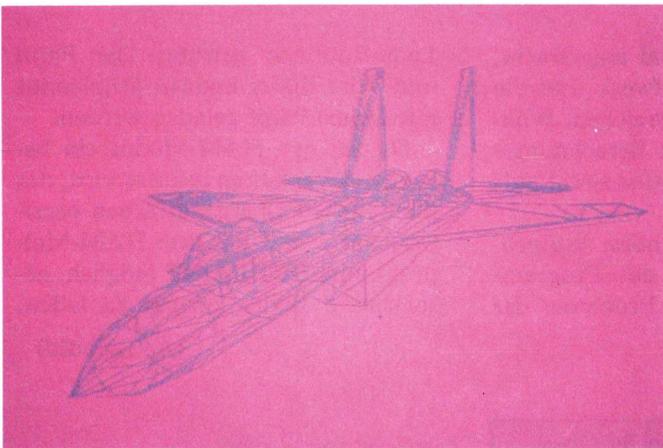


Der 3D Object-Editor TRI-VIEW

Der TRI-VIEW besteht aus drei Windows, wobei jedes die Ansicht auf einer der drei Raumachsen darstellt. Die Blickrichtung eines jeden der drei Windows, ist als dessen Name ständig angezeigt. Die Fenster lassen sich vergrößern, verkleinern und verschieben. Außerdem sind im Fensterrahmen zahlreiche Gadgets vorhanden, die zur Edierung dienen. Veränderungen aus einer Sichtachse werden korrekt auf die Sichtachsen der anderen Windows umgerechnet, so daß

te zu selektieren. Man kann jeden Punkt einzeln, durch Anklicken mit der Maus, oder alle Punkte im Sichtbereich durch Anklicken des entsprechenden Gadgets selektieren. Hat man einen einzelnen Punkt selektiert, kann man die mit ihm zusammenhängenden über das Menü mitselektieren. Ein anderer Menüpunkt öffnet ein kleines Viereck in den Fenstern. Bei Mausclick werden alle Punkte innerhalb des Fensters selektiert.

Selektierte Punkte werden anders-



Die Dimensionen des Werkes lassen sich als WIREFRAME schnell darstellen

man immer die aktuelle Lage der Objekte aus allen Ansichten vor Augen hat. Die Objekte müssen immer aus Dreiecken zusammengesetzt werden.

Die Darstellung der Objekte erfolgt im TRI-VIEW als Gitter-Modell, wobei nichts verdeckt wird.

Um mit der Erstellung eines Objektes zu beginnen, gibt es mehrere Möglichkeiten. Entweder man lädt eine vorhandene 'Scene', bedient sich des Menüpunktes 'Edit/Add', um einen geometrischen Körper zu erzeugen, oder man baut mit 'Tools/Curve' sein erstes Dreieck. Um die vorhandenen geometrischen Körper, wie z. B. Halb-, Kugel, Kreis, Scheibe, Zylinder etc., zu positionieren, klickt man mit der Maus auf eine gewünschte Stelle in einem Fenster. Jetzt liegt das im Fenster vorhandene Fadenkreuz an dieser Stelle. Es ist darauf zu achten, daß das Fadenkreuz auch in den anderen Fenstern positioniert wird. Jetzt kann man 'Edit/Add' aufrufen, und nach einer Abfrage für die Auflösung des Körpers, wird dieser an die markierte Stelle gezeichnet.

Um den Körper nun zu verändern, sind die zu verändernden Gitterpunk-

farbig dargestellt, und können mit den fast gleichen Methoden wieder deselektiert werden.

Mit den selektierten Punkten kann man nun einiges erreichen. Verschieben, vergrößern, drehen etc.. Natürlich kann man auch bestimmte Punkte eines Körpers selektieren und diese dann etwa aus dem Körper herausziehen. Die Verbindungslinien werden dann entsprechend geändert. Nachdem die gewünschten Objekte erstellt sind, kann man deren Oberflächen mit Farben und Struktur, z. B. matt, spiegelnd, gläsern etc., versehen.

Im PAINTING werden auch die Farben sichtbar

Nun ist im Szenario noch der Standpunkt des Beobachters und dessen 'Blickziel' (= freie Übersetzung von Target) festzulegen. Genauso die Position der Lichtquellen, deren Farben und Helligkeit.

Eine Auswahl an Hintergrundkonfigurationen steht nun noch zur Verfügung. Vor der Bildberechnung kann man noch Kameradaten, Bildschirmauflösung, Farbzahl und andere Daten für das eigentliche Bild festlegen. Mit der Auswahl der Daten sollte vorsichtig umgegangen werden, da die Routinen zum Speichern und Laden nicht problemlos funktionieren. Die im Test vorliegende Version 1.05 war nicht in der Lage, eigene Bilder nach dem Speichern korrekt zu laden. Verschiebungen und auch Streifen im geladenen Bild machen dieses unbrauchbar. Auch die auf der Diskette vorhandenen Bilder sind immer nach unten verschoben, so daß die obere Bildschirmkante freibleibt und darunterliegende Screens zu sehen sind. Es bleibt zu hoffen, daß dieses Problem schnell gelöst wird.

Zu jedem Zeitpunkt dieser Erstellung kann man die edierte Gittergraphik in ein Bild umrechnen lassen. Dafür gibt es vier Qualitätsstufen:

1. **WIREFRAME** zeigt innerhalb kurzer Zeit das eigentliche Bild als Gitterkonstruktion, wobei das Arrangement des Beobachters und der Objekte kontrolliert werden kann.
2. **PAINTING** füllt die Flächen aus, zeigt deren Farben und die Einflüsse der Lichtquellen. Die Ausführungszeit ist noch relativ gering, in wenigen Minuten hat man eine weitgehende Kontrolle über sein Werk.



3. **SNAPSHOT** rechnet die Lichtstrahlen schon etwas weiter, aber noch nicht in voller Qualität. Die Ausführungszeit wird schon bedeutender, und liegt teilweise schon jenseits der 60 min Grenze. Jedoch ist es empfehlenswert, komplizierte Arrangements hier nochmals auf die eigenen Vorstellungen hin zu überprüfen, bevor man mit

4. **PHOTO** einen gewaltigen Rechenprozeß in Gange setzt, der schon meist eine ganze Nacht benötigt, und bei großzügigem Umgang mit Lichtquellen, Spiegelflächen und Gitterpunkten beliebig in die Länge gezogen werden kann. Der gleichzeitige Betrieb anderer multitaskingfähiger Programme gestaltet sich schwierig, auch wenn genug Speicher vorhanden sind. Jedoch kann nach einer kurzen intensiven Rechenphase der TRI-VIEW wieder benutzt werden.

Nach der 'Photo' Berechnung erhält man Bilder, in denen Licht und Schatten und der räumliche Zusammenhang der Objekte exakt berechnet sind.

Als letzte Zugabe kann man mehrere komplette Arbeitsvorgänge als 'Script' definieren, was in etwa einem Batch entspricht.

Alle Befehle des Editors und des Berechnungsteils stehen zur Verfügung. Somit können ohne das Zutun des Anwenders, komplette Bildfolgen erstellt werden. Wegen der langen Rechenzeiten stellt sich jedoch die Frage nach dem Sinn, obwohl nun das Problem „Was macht mein Amiga wenn ich im Urlaub bin“ gelöst scheint.

Mindestanforderungen an den Amiga:

Die Speicherkapazität muß mindestens 512 KByte betragen. Jedoch nur mit einem MegaByte kann man hohe Auflösungen mit vielen Farben fahren.



Bild 4:
Nach vielen Stunden,
das End-
produkt

Zwei Laufwerke sind angebracht, jedoch auch ein Laufwerk und ein Diskjockey sollten ausreichen. Während der langwierigen Berechnungsphase sind allerdings keine zwei Laufwerke nötig.

Der ansonsten zumindest semiprofessionelle Anspruch des Programmes wird durch die Probleme der

Lade-Routinen getrübt. Die Paint und Wire Bilder können problemlos mit DeluxePaint geladen werden.

Da für den HAM-Modus ein besonderes Verfahren benutzt wird, das die üblichen Übergangsfarben beseitigt, war es nur mit dem HAM-Malprogramm DIGIPAIN möglich, gespeicherte Bilder korrekt zu laden.

(A. Diebold)

Kurzbewertung:

- + sehr gut gelöste 3d-Objekt Edierung.
- + neue, ungewohnt gute Graphikqualität für den Amiga.
- + relativ günstiger Preis.
- + gute Kontrolle des Objekt Arrangements
- + Sauberer HAM-Modus

- dadurch jedoch auch Inkompatibilität zum IFF-Standard.
- beschränkte Anzahl von Oberflächenstrukturen
- keine Möglichkeit diese selbst herzustellen.
- lange Berechnungszeiten.
- fehlerhafte Bildladeroutinen

Hersteller: BYTE by BYTE

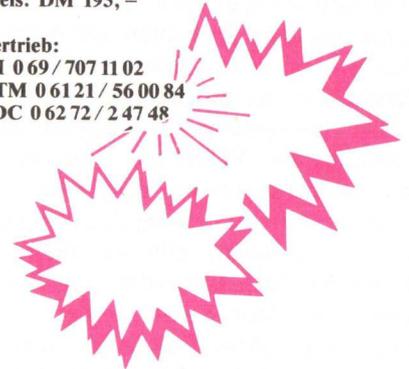
Preis: DM 195,-

Vertrieb:

IM 069/707 1102

DTM 06121/56 00 84

PDC 062 72/247 48



Moviemaker

1/2.

für Filmenamen. Um diese Gadgets herum sind die Gadgets für Lade- und Speicheranweisungen angeordnet. Im rechten Teil des ControlWindows finden Sie eine Leiste mit mehreren Gadgets, die Funktionen aufrufen, welche Ihnen das Erstellen eines Films erheblich erleichtern.

Die Bedienung

Die drei obersten Bewegungsmöglichkeiten im ControlWindow des MM entsprechen, wie schon die Drehbewegungen beim Animator, einer Drehung Ihres Objekts um die drei Raumachsen und um den Mittelpunkt des Koordinatensystems des Objekts, der im Objektdatensatz durch die x-, y- und z-Koordinaten 0,0,0 repräsentiert wird. Darunter folgen die drei Möglichkeiten xmove, ymove und zmove. Hierbei handelt es sich um geradlinige Bewegungen im Raum auf der jeweiligen Raumachse. Die nächsten drei Bewegungen stellen eine Drehung um die x-, y- oder z-Achse und um den Mittelpunkt des Weltkoordinatensystems dar. Dieser und der Mittelpunkt des Objektkoordinatensystems sind nach dem Laden eines Objektes deckungsgleich, so daß eine Zentralsdrehung erst einmal den gleichen Effekt hat, wie eine normale Drehung. Dies ändert sich aber, wenn Sie Ihr Objekt mittels x-, y- oder z-move aus dem Mittelpunkt des Weltkoordinatensystems herausbewegen. Es beginnt dann, um diesen zu kreisen, während es sich unabhängig davon noch um seine eigene Achsen drehen kann.

Leider hat die Veröffentlichung dieses Programms etwas auf sich warten lassen, da es den vorgesehenen Rahmen gesprengt hat. Ursprünglich waren ja nur erweiternde Funktionen zum Animator vorgesehen, doch dieses Konzept ließ sich in Anbetracht einiger dabei entstehender Komplikationen und dem Zwang zu teilweise noch wesentlichen Änderungen nicht verwirklichen; aus diesem Grund wird der Moviemaker als eigenständiges Programm abgedruckt. Dennoch lassen sich einige Routinen aus dem Animator – mit geringfügigen Änderungen – wiederverwenden, was Ihnen einige Tipparbeit ersparen sollte. So können Sie die Funktionen `object_drawer()`, `init_trig()`, `ausstieg()`, `write_int()` und `lies_int()` unter Beachtung und Einfügung einiger Veränderungen übernehmen.

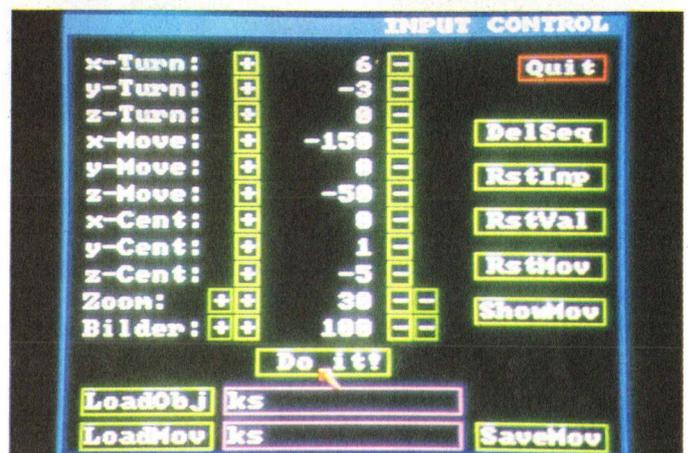
Das Programm

Wie angekündigt, wartet der Moviemaker mit einigen Features auf, die Ihnen neue Möglichkeiten erschließen. So können Sie mit ihm Filme Ihrer 3D-Objekte aufzeichnen und abspeichern, um Sie sich später jederzeit wieder zu betrachten. Auch verfügt der MM über erweiterte Bewegungsmöglichkeiten der Objekte und eine Routine zur korrekten perspektivischen Projektion. In der Be-

rechnung ist er zwar langsamer als der Animator, dafür laufen die Filme mit höherer Geschwindigkeit ab. Der MM ist komplett mausgesteuert (mit Ausnahme der Filmenamen, die über Tastatur eingegeben werden müssen) und bietet zur übersichtlichen Eingabe ein spezielles ControlWindow.

Nach dem Start meldet sich das Programm mit eben diesem ControlWindow. Vor sich sehen Sie im linken Teil des Fensters die verschiedenen Bewegungsmöglichkeiten, dahinter die Gadgets zur Einstellung selbiger und den eingestellten Wert jeweils zwischen dem Plus- und dem Minusgadget. Im unteren Bereich befinden sich zwei Stringgadgets, das Obere für Projektnamen, das Untere

Das Auswahlmeneue von Moviemaker erlaubt eine Vielzahl von Optionen anzuwählen.



Der Zoom verschiebt ihre Sichtposition auf das Objekt zu bzw. davon weg. Die letzte Eingabe in der Leiste gibt die Anzahl der Bilder an, die der MM für die nächste Sequenz berechnen soll.

Der MM ist so ausgelegt, daß Sie Ihre Animationen sequenzweise erstellen. Als erstes müssen Sie ihr gewünschtes Objekt einladen. Klicken Sie hierzu mit der Maus das obere Stringgadget an; der Cursor erscheint, und Sie können den Namen des Objekts eingeben. Klicken Sie nun das LOADOBJ-Gadget an, und der MM lädt das gewünschte Objekt (sollte er es in der aktuellen oder angegebenen Directory nicht finden, teilt er Ihnen das über einen Requester mit). Wie schon beim Animator, müssen sie die Extension (.3d) nicht eingeben. Nun können Sie über die Plus- und Minusgadgets die gewünschten Bewegungen und die Anzahl der Bilder für diese Sequenz eingeben. Ein Klick auf das DOIT-Gadget, und der MM berechnet die Sequenz. Hat er alle Bilder abgearbeitet, so erscheint wieder das ControlWindow. Nun haben sie die verschiedensten Möglichkeiten, fortzufahren. Die Einfachste ist, wenn die erste Sequenz Ihren Vorstellungen entspricht, neue Werte einzugeben

Film erstellt hat, und die letzte Sequenz, das Objekt, auf einmal ruckartig, oder in die falsche Richtung bewegt hat. DELSEQ setzt das Objekt auch auf die Raumposition zurück, auf der es sich vor der Sequenz befunden hat. Zum Löschen eines ganzen Films steht das Gadget RSTMOV zur Verfügung, das beim Anklicken den programminternen Bildzähler zurücksetzt. Dadurch wird das bis dahin bewegte Objekt allerdings nicht auf den Startpunkt zurückgesetzt; dies eröffnet Ihnen die Möglichkeit, ein Objekt an einen beliebigen Punkt zu bewegen, den Bildzähler zurückzusetzen und an der neuen Position Ihren Film zu beginnen: möchten Sie das Objekt auf den Startpunkt (also die 0,0,0-Position im Weltkoordinatensystem) zurücksetzen, betätigen sie einfach das RSTVAL-Gadget, woraufhin alle eingestellten Dreh- und Verschiebungswerte auf Null gebracht werden. Das RSTINP-Gadget dient dazu, die aktuell eingestellten Eingaben im ControlWindow auf Null zurückzusetzen, was bisweilen viel Zeit spart, wenn man z.B. noch verschiedene hohe Werte eingestellt hat, nun aber neu beginnen will. Mittels SHOWMOV können Sie sich jederzeit den bereits erstellten (oder auch

noch keinen Filmnamen eingegeben. Dies ist dann nützlich, wenn man Filme nachbearbeiten will, denn mit dem MM können sie einen schon früher erstellten Film laden und neue Sequenzen anhängen, vorausgesetzt, das Filmarray ist noch nicht voll. Dabei sind Sie nicht gezwungen, das gleiche Objekt weiterzuverwenden. Es ist auch möglich, bei der Erstellung eines Films ein neues Objekt zu laden und an der aktuellen Position fortzufahren, d.h. mit den eingestellten Bewegungen von der letzten Position aus, wodurch im Film interessante Effekte erzeugt werden können. Der MM bietet Ihnen also einige Kombinationsmöglichkeiten seiner Funktionen; andererseits ist wegen dieser getrennten Auslegung der Funktionen darauf zu achten, daß zum Beispiel bei einem gewünschten Neubeginn sowohl der Filmzähler als auch die Drehwerte eines Objekts zurückgesetzt werden. Im großen und ganzen ist die Bedienung aber kinderleicht, und nach ein paar Versuchen kennt man sich mit den Möglichkeiten ausreichend aus, um jedes mit dem MM erreichbare Ergebnis zu verwirklichen.

Ein kleiner Hinweis noch zu den Bewegungsmöglichkeiten: Ein Objekt ist im MM beliebig verschiebbar, der Zoom und die Verschiebungen kennen keine, die Drehwinkel nur die 359-Grad-pro-Bild-Limitierung.

Auch dürfen Ihre Objekte hier beliebig groß sein bzw. werden, aber es sei dennoch zu etwas Vorsicht geraten, denn der MM zeichnet zwar auch Objekte, die wesentlich größer als der Bildschirm sind, wofür extra ein überdimensioniertes Raster zur Verfügung steht, und hat sich in vielen Probeläufen als sehr absturzsicher erwiesen; trotzdem verabschiedete er sich bei zwei oder drei Tests mit einem Not-Enough-Memory-Guru. Dies allerdings läßt sich ganz sicher vermeiden, wenn man auf unkontrollierte Bewegungen, gigantische Objekte oder Durchflüge durch den Mittelpunkt eines Objekts verzichtet.

Die Funktionen

Die meisten Funktionen des MM sind alte Bekannte aus dem Animator, wenn auch einige Modifikationen eingefügt sind. So dient trans-



Das KICKSTART Logo beginnt seinen Bewegungsablauf.

und den MM weiterrechnen zu lassen. Was aber, wenn die Bewegungen nicht so ausgefallen sind, wie man das erwartet hat? Kein Problem, hierfür gibt es auf der rechten Seite das DELSEQ-Gadget. Wenn sie dieses Anklicken, wird die jeweils letzte berechnete Sequenz gelöscht. Diese Funktion ist sehr nützlich, vor allem, wenn man bereits einen längeren

einen nachgeladenen Film) abspielen lassen. Um einen Film zu laden oder zu speichern, klicken sie bitte das untere Stringgadget an, geben den Namen (ohne Extension) ein und betätigen dann das betreffende Gadget.

Zur Übersicht während der Erstellung eines Films stehen die Namen des aktuellen Films immer in den Stringgadgets, es sei denn, Sie haben

form() weiterhin dem Berechnen eines neuen Bildes, durchläuft aber jetzt nicht mehr nur eine Transformationsmatrix, sondern verschiebt nach der Ersten das Objekt um die x-, y- und z-Movewerte und führt danach die Transformationsmatrix für die Zentralsdrehung aus. Da auch die Sichtbarkeitsberechnung etwas komplexer ausgeführt wird, um korrekte perspektivischen Transformation zu gewährleisten, ist es erforderlich, daß transform() mit Float-Variablen arbeitet, statt mit Longs. Außerdem werden in transform(), die als sichtbar erkannten Flächen in das Filmarray eingetragen. Transform() bricht ab, wenn entweder alle Bilder berechnet sind oder das Filmarray vorher voll ist, und gibt letztere Information an object_drawer() weiter, der in diesem Fall auch abbricht.

Neu dazu gekommen ist die Funktion input(), die eigentlich nichts weiter zu tun hat, als Ihre Eingaben zu verwalten und die verschiedenen Unterrountinen aufzurufen.

Die Funktionen film_save() und film_loader() speichern bzw. laden einen Film. Hierzu werden die Amiga-Betriebssystemroutinen eingesetzt, welche die maximal knapp 80 Kbyte Daten in weniger als 20 Sekunden speichern und weniger als 10 Sekunden laden. Film_save() eröffnet vor Ausführung einen Requester mit Exit-Option, was sehr nützlich ist, falls Sie diese Funktion einmal versehentlich aufgerufen haben. Ebenso meldet film_loader() über einen Requester, wenn der gewünschte File nicht gefunden werden kann.

Movie() ist eine einfache Zeichenroutine, welche die Filme aus dem Datenarray abspielt. Vor dem Start



Noch ist keine Ebene verschoben, aber ...

und nach dem Ende des Films stoppt movie() für jeweils eine Sekunde, was unter anderem auch notwendig ist, um einen sauberen refresh des ControlWindows zu gewährleisten. Dieser wird von der Funktion inti_control() eingeführt.

Die Funktionen reset_val(), reset_inp() und deleter() werden (in dieser Reihenfolge) bei Betätigen der Gadgets RSTVAL, RSTINP oder DELSEQ aufgerufen; ihre Auswirkungen sind bereits weiter oben aufgeführt.

Was noch zu beachten wäre

Wie schon der Animator, wurde dieses Programm mit dem Aztec 3.4a Compiler erzeugt. Um es zu compilieren, muß man allerdings die Compilervoreinstellung auf `< +f -E200 >` ändern, da der Compiler sonst vom Compilieren nicht genug Platz für Variablenlisten einräumt. Sollte dieses Problem auf dem Lattice auftreten, muß der Anwender die entsprechende Einstellung vornehmen.

Da alle Routinen für Diskettenzu-

griffe die Betriebssystemroutinen des Amiga verwenden, ist in den Aufrufen peinlich genau darauf zu achten, daß Long-Werte übergeben werden. Vergißt man nämlich nur einmal das groß L, kann dies unangenehme Folgen haben (so hätte bei einem frühen Testlauf der Versuch, 10 Megabyte auf Diskette zu schreiben, beinahe zu einer fatalen Datenvernichtung geführt; glücklicherweise konnte der Schaden auf 16 Read/Write-Errors eingedämmt werden).

Das Format der Objektdatensätze hat sich übrigens um eine Kleinigkeit geändert: Am Ende des Datensatzes müssen jetzt zwei statt einer Farbliste stehen. Der Grund dafür ist, daß Sie so die Möglichkeit haben, die Füllfarbe einer Fläche und deren Umrandungsfarbe getrennt zu bestimmen. Die erste Farbliste gibt die Füllfarben der Flächen an, die Zweite ist für die Umrandung zuständig. Diese Änderung bringt nicht viel Mühe mit sich, aber achten Sie darauf, daß Sie nicht etwa Objekte mit dem alten und dem neuen Datenformat durcheinander bringen.

Leider hat der Moviemaker, zumindest für ein Listing zum abtippen, doch recht umfangreiche Dimensionen angenommen. Es wäre zwar möglich gewesen, ihn noch um ca. 100 Zeilen zu kürzen, aber davon wurde abgesehen, da dieser Kürzung gerade die sehr angenehmen Hilfsfunktionen und Requester, die Fehlerangaben abfangen, zum Opfer gefallen wären, und die Kürzung hätte dennoch nur einen Tropfen auf den heißen Stein bedeutet. Viel Anteil an der Länge tragen auch die zur Initia-



...ein bisschen Zoom und den 'z' Vektor verschoben, läßt das Logo räumlich erscheinen.


```

38: SHORT xyc4[] = {0,0,0,11,40,11,40,0,0,0};
39: UBYTE buff_1[13],buff_2[13],buff_3[13],buff_4[13];
40:
41: struct Border outline =
42: { 0,0,(UBYTE)3,(UBYTE)1,JAM1,5,(SHORT*)xyc1,NULL };
43: struct Border outline2 =
44: { 0,0,(UBYTE)5,(UBYTE)1,JAM1,5,(SHORT*)xyc1,NULL };
45: struct Border bigline =
46: { 0,0,(UBYTE)6,(UBYTE)1,JAM1,5,(SHORT*)xyc2,NULL };
47: struct Border medline =
48: { 0,0,(UBYTE)3,(UBYTE)1,JAM1,5,(SHORT*)xyc3,NULL };
49: struct Border quitline =
50: { 0,0,(UBYTE)2,(UBYTE)1,JAM1,5,(SHORT*)xyc4,NULL };
51:
52: struct IntuiText r_text8 =
53: { (UBYTE)0,(UBYTE)1,JAM1,4,14,NULL,
54:   (UBYTE *)"wird geloescht!",NULL };
55: struct IntuiText r_text7 =
56: { (UBYTE)0,(UBYTE)1,JAM1,4,4,NULL,
57:   (UBYTE *)"Aktueller Film",&r_text8 };
58: struct IntuiText r_text6 =
59: { (UBYTE)0,(UBYTE)1,JAM1,4,14,NULL,
60:   (UBYTE *)"wirklich save?",NULL };
61: struct IntuiText r_text5 =
62: { (UBYTE)0,(UBYTE)1,JAM1,4,4,NULL,
63:   (UBYTE *)" Wollen Sie",&r_text6 };
64: struct IntuiText r_text4 =
65: { (UBYTE)0,(UBYTE)1,JAM1,4,14,NULL,
66:   (UBYTE *)"nicht oeffnen!",NULL };
67: struct IntuiText r_text3 =
68: { (UBYTE)0,(UBYTE)1,JAM1,4,4,NULL,
69:   (UBYTE *)"Kann Datenfile",&r_text4 };
70: struct IntuiText r_text2 =
71: { (UBYTE)0,(UBYTE)1,JAM1,21,14,NULL,(UBYTE *)"ist voll!",
  NULL };
72: struct IntuiText r_text1 =
73: { (UBYTE)0,(UBYTE)1,JAM1,21,4,NULL,
74:   (UBYTE *)"Filmmarray",&r_text2 };
75: struct IntuiText e_text =
76: { (UBYTE)0,(UBYTE)1,JAM1,6,4,NULL,(UBYTE *)"Exit",NULL };
77:
78: struct IntuiText s_text =
79: { (UBYTE)0,(UBYTE)1,JAM1,6,4,NULL,(UBYTE *)"Save",NULL };
80:
81: struct IntuiText o_text =
82: { (UBYTE)0,(UBYTE)1,JAM1,6,4,NULL,(UBYTE *)"Load",NULL };
83:
84: struct IntuiText n_text =
85: { (UBYTE)0,(UBYTE)1,JAM1,6,4,NULL,(UBYTE *)"Nanu!?",NULL
  };
86: struct IntuiText c_text =
87: { (UBYTE)0,(UBYTE)1,JAM1,6,4,NULL,(UBYTE *)"So'n Pech",NULL
  };
88: struct IntuiText doit_text =
89: { (UBYTE)1,(UBYTE)1,JAM1,2,2,NULL,(UBYTE *)" Do it!",NULL
  };
90: struct IntuiText load_text1 =
91: { (UBYTE)1,(UBYTE)1,JAM1,2,2,NULL,(UBYTE *)"LoadObj",NULL
  };
92: struct IntuiText load_text2 =
93: { (UBYTE)1,(UBYTE)1,JAM1,2,2,NULL,(UBYTE *)"LoadMov",NULL
  };
94: struct IntuiText save_text =
95: { (UBYTE)1,(UBYTE)1,JAM1,2,2,NULL,(UBYTE *)"SaveMov",NULL
  };
96: struct IntuiText show_text =
97: { (UBYTE)1,(UBYTE)1,JAM1,2,2,NULL,(UBYTE *)"ShowMov",NULL
  };
98: struct IntuiText quit_text =
99: { (UBYTE)1,(UBYTE)1,JAM1,4,2,NULL,(UBYTE *)"Quit",NULL };
100:
101: struct IntuiText reset_text3 =
102: { (UBYTE)1,(UBYTE)1,JAM1,6,2,NULL,(UBYTE *)"RstInp",NULL
  };
103: struct IntuiText reset_text2 =
104: { (UBYTE)1,(UBYTE)1,JAM1,6,2,NULL,(UBYTE *)"RstVal",NULL
  };
105: struct IntuiText reset_text1 =
106: { (UBYTE)1,(UBYTE)1,JAM1,6,2,NULL,(UBYTE *)"RstMov",NULL
  };
107: struct IntuiText delete_text =
108: { (UBYTE)1,(UBYTE)1,JAM1,6,2,NULL,(UBYTE *)"DelSeq",NULL
  };
109: struct IntuiText minus_text =
110: { (UBYTE)1,(UBYTE)1,JAM1,2,2,NULL,(UBYTE *)"-",NULL };
111: struct IntuiText plus_text =
112: { (UBYTE)1,(UBYTE)1,JAM1,2,2,NULL,(UBYTE *)"+",NULL };
113:
114: struct StringInfo inp_str1 =
115: { (UBYTE *)&buff_1,(UBYTE *)&buff_2,0,13,0,
  NULL,NULL,NULL,NULL,NULL,NULL };
116: struct StringInfo inp_str2 =
117: { (UBYTE *)&buff_3,(UBYTE *)&buff_4,0,13,0,
  NULL,NULL,NULL,NULL,NULL,NULL };
118:
119: struct Gadget read_gad2 =
120: { NULL,78,187,105,9,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,STRGADGET,(APTR)&bigline,
  NULL,NULL,0,(APTR)&inp_str2,NULL,NULL };
121: struct Gadget read_gad1 =
122: { &read_gad2,78,172,105,9,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,STRGADGET,(APTR)&bigline,
  NULL,NULL,0,(APTR)&inp_str1,NULL,NULL };
123: struct Gadget delete_gad =
124: { &read_gad1,190,50,61,12,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&medline,
  NULL,&delete_text,0,NULL,35,NULL };
125: struct Gadget reset_gad3 =
126: { &delete_gad,190,70,61,12,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&medline,
  NULL,&reset_text3,0,NULL,36,NULL };
127: struct Gadget reset_gad2 =
128: { &reset_gad3,190,90,61,12,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&medline,
  NULL,&reset_text2,0,NULL,34,NULL };
129: struct Gadget reset_gad1 =
130: { &reset_gad2,190,110,61,12,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&medline,
  NULL,&reset_text1,0,NULL,33,NULL };
131: struct Gadget super_minus2 =
132: { &reset_gad1,162,138,11,11,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline2,
  NULL,&minus_text,0,NULL,32,NULL };
133: struct Gadget super_minus1 =
134: { &super_minus2,162,126,11,11,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline2,
  NULL,&minus_text,0,NULL,31,NULL };
135: struct Gadget super_plus2 =
136: { &super_minus1,68,138,11,11,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline2,
  NULL,&plus_text,0,NULL,30,NULL };
137: struct Gadget super_plus1 =
138: { &super_plus2,68,126,11,11,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline2,
  NULL,&plus_text,0,NULL,29,NULL };
139: struct Gadget doit_gad =
140: { &super_plus1,90,153,61,12,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&medline,
  NULL,&doit_text,0,NULL,28,NULL };
141: struct Gadget quit_gad =
142: { &doit_gad,210,20,41,12,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&quitline,
  NULL,&quit_text,0,NULL,27,NULL };
143: struct Gadget show_gad =
144: { &quit_gad,190,130,61,12,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&medline,
  NULL,&show_text,0,NULL,26,NULL };
145: struct Gadget save_gad =
146: { &show_gad,190,185,61,12,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&medline,
  NULL,&save_text,0,NULL,25,NULL };
147: struct Gadget load_gad2 =
148: { &save_gad,10,185,61,12,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&medline,
  NULL,&load_text2,0,NULL,24,NULL };
149: struct Gadget load_gad1 =
150: { &load_gad2,10,170,61,12,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&medline,
  NULL,&load_text1,0,NULL,23,NULL };
151: struct Gadget plus_gad11 =
152: { &load_gad1,80,138,11,11,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
  NULL,&plus_text,0,NULL,22,NULL };
153: struct Gadget plus_gad10 =
154: { &plus_gad11,80,126,11,11,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
  NULL,&plus_text,0,NULL,21,NULL };
155: struct Gadget plus_gad9 =
156: { &plus_gad10,80,114,11,11,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
  NULL,&plus_text,0,NULL,20,NULL };
157: struct Gadget plus_gad8 =
158: { &plus_gad9,80,102,11,11,GADGHCOMP,
  RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
  NULL,&plus_text,0,NULL,19,NULL };
159: struct Gadget plus_gad7 =

```

SOFTWARE

```
198: { &plus_gad8,80,90,11,11,GADGHCOMP,
199: RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
200: NULL,&plus_text,0,NULL,18,NULL };
201: struct Gadget plus_gad6 =
202: { &plus_gad7,80,78,11,11,GADGHCOMP,
203: RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
204: NULL,&plus_text,0,NULL,17,NULL };
205: struct Gadget plus_gad5 =
206: { &plus_gad6,80,66,11,11,GADGHCOMP,
207: RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
208: NULL,&plus_text,0,NULL,16,NULL };
209: struct Gadget plus_gad4 =
210: { &plus_gad5,80,54,11,11,GADGHCOMP,
211: RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
212: NULL,&plus_text,0,NULL,15,NULL };
213: struct Gadget plus_gad3 =
214: { &plus_gad4,80,42,11,11,GADGHCOMP,
215: RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
216: NULL,&plus_text,0,NULL,14,NULL };
217: struct Gadget plus_gad2 =
218: { &plus_gad3,80,30,11,11,GADGHCOMP,
219: RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
220: NULL,&plus_text,0,NULL,13,NULL };
221: struct Gadget plus_gad1 =
222: { &plus_gad2,80,18,11,11,GADGHCOMP,
223: RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
224: NULL,&plus_text,0,NULL,12,NULL };
225: struct Gadget minus_gad11 =
226: { &minus_gad1,150,138,11,11,GADGHCOMP,
227: RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
228: NULL,&minus_text,0,NULL,11,NULL };
229: struct Gadget minus_gad10 =
230: { &minus_gad11,150,126,11,11,GADGHCOMP,
231: RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
232: NULL,&minus_text,0,NULL,10,NULL };
233: struct Gadget minus_gad9 =
234: { &minus_gad10,150,114,11,11,GADGHCOMP,
235: RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
236: NULL,&minus_text,0,NULL,9,NULL };
237: struct Gadget minus_gad8 =
238: { &minus_gad9,150,102,11,11,GADGHCOMP,
239: RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
240: NULL,&minus_text,0,NULL,8,NULL };
241: struct Gadget minus_gad7 =
242: { &minus_gad8,150,90,11,11,GADGHCOMP,
243: RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
244: NULL,&minus_text,0,NULL,7,NULL };
245: struct Gadget minus_gad6 =
246: { &minus_gad7,150,78,11,11,GADGHCOMP,
247: RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
248: NULL,&minus_text,0,NULL,6,NULL };
249: struct Gadget minus_gad5 =
250: { &minus_gad6,150,66,11,11,GADGHCOMP,
251: RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
252: NULL,&minus_text,0,NULL,5,NULL };
253: struct Gadget minus_gad4 =
254: { &minus_gad5,150,54,11,11,GADGHCOMP,
255: RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
256: NULL,&minus_text,0,NULL,4,NULL };
257: struct Gadget minus_gad3 =
258: { &minus_gad4,150,42,11,11,GADGHCOMP,
259: RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
260: NULL,&minus_text,0,NULL,3,NULL };
261: struct Gadget minus_gad2 =
262: { &minus_gad3,150,30,11,11,GADGHCOMP,
263: RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
264: NULL,&minus_text,0,NULL,2,NULL };
265: struct Gadget minus_gad1 =
266: { &minus_gad2,150,18,11,11,GADGHCOMP,
267: RELVERIFY:GADGIMMEDIATE,BOOLGADGET,(APTR)&outline,
268: NULL,&minus_text,0,NULL,1,NULL };
269:
270: struct NewWindow graph_window1 =
271: { 0,0,320,200,0,1,MOUSEBUTTONS:GADGETUP,
272: SIMPLE_REFRESH:BORDERLESS,NULL,NULL,NULL,
273: NULL,NULL,320,200,320,200,CUSTOMSCREEN };
274: struct NewWindow graph_window2 =
275: { 0,0,320,200,0,1,MOUSEBUTTONS:GADGETUP,
276: SIMPLE_REFRESH:BORDERLESS,NULL,NULL,NULL,
277: NULL,NULL,320,200,320,200,CUSTOMSCREEN };
278: struct NewWindow control_window =
279: { 30,0,260,200,1,4,MOUSEBUTTONS:GADGETDOWN:GADGETUP,
280: SIMPLE_REFRESH:ACTIVATE:NOCAREREFRESH,&minus_gad1,NULL
,
281: (UBYTE *)" INPUT CONTROL",
282: NULL,NULL,260,200,260,200,CUSTOMSCREEN };
283:
284: USHORT farb_lst[] = {0x000,0xffff,0xf00,0x0f0,0x00f,0x0a2
,
285: 0xf0f,0xff0,0xf6b,0x3a9,0x492,0x666,0xbbb,0x07c,0xc40,0x9
0a};
286: BYTE *area_plane,control_lst[12000];
287: char *filename, *zielchar, zielstring[10];
288: UWORD areabuf[500];
289: int seq_deleted = TRUE;
290: int xlst[16384],ylst[16384];
291: int pkt_zahl = 0,flaech_zahl = 0,zoom = 0;
292: int xturn,yturn,zturn,xdreher,ydreher,zdreher;
293: int xturn2,yturn2,zturn2,xdreher2,ydreher2,zdreher2;
294: int prev_xturn,prev_xturn2,prev_yturn,prev_yturn2;
295: int prev_zturn,prev_zturn2,xver,yver,zver;
296: int check_count = 4,pcnt = 0,bild_counter;
297: int prev_check_count, prev_pcnt;
298: int sichtbar[MAXFL],plst[MAXFL][MAXPPF],pkt_p_flaech[MAXF
L];
299: long sp[MAXP],ze[MAXP],cx[MAXP],cz[MAXP],cy[MAXP];
300: long afarbe[MAXFL],farbe[MAXFL];
301: float sinus[360],cosin[360];
302: float xmove,ymove,zmove,zoomer,prev_zoomer;
303: float prev_xmove,prev_ymove,prev_zmove;
304:
305: main()
306: {
307: init_trig();
308: graph_screens();
309: rp_1 = window_1->RPort;
310: rp_2 = window_2->RPort;
311: rp_3 = cont_wind->RPort;
312: init_control();
313: rp_1->Layer->ClipRect->bounds.MinX = (SHORT)0;
314: rp_1->Layer->ClipRect->bounds.MinY = (SHORT)0;
315: rp_1->Layer->ClipRect->bounds.MaxX = (SHORT)319;
316: rp_1->Layer->ClipRect->bounds.MaxY = (SHORT)199;
317: rp_2->Layer->ClipRect->bounds.MinX = (SHORT)0;
318: rp_2->Layer->ClipRect->bounds.MinY = (SHORT)0;
319: rp_2->Layer->ClipRect->bounds.MaxX = (SHORT)319;
320: rp_2->Layer->ClipRect->bounds.MaxY = (SHORT)199;
321: InitArea(&area_info,&areabuf[0],200L);
322: rp_1->AreaInfo = &area_info;
323: rp_2->AreaInfo = &area_info;
324: rp_1->TmpRas = &temp_ras;
325: rp_2->TmpRas = &temp_ras;
326: area_plane = AllocMem(25600L,MEMF_CHIP);
327: rp_1->TmpRas->RasPtr = area_plane;
328: rp_1->TmpRas->Size = (long)RASSIZE(512,400);
329: rp_2->TmpRas->RasPtr = area_plane;
330: rp_2->TmpRas->Size = (long)RASSIZE(512,400);
331: input();
332: ausstieg();
333: }
334:
335: graph_screens()
336: {
337: int loop,wert;
338:
339: IntuitionBase=(struct IntuitionBase *)
340: OpenLibrary("intuition.library",0L);
341: GfxBase=(struct GfxBase *)
342: OpenLibrary("graphics.library",0L);
343: screen_1=(struct Screen *)OpenScreen(&graph_screen);
344: screen_2=(struct Screen *)OpenScreen(&graph_screen);
345: LoadRGB4(&screen_1->ViewPort,farb_lst,16L);
346: LoadRGB4(&screen_2->ViewPort,farb_lst,16L);
347: graph_window1.Screen = screen_1;
348: graph_window2.Screen = screen_2;
349: control_window.Screen = screen_2;
350: window_1=(struct Window *)OpenWindow(&graph_window1);
351: window_2=(struct Window *)OpenWindow(&graph_window2);
352: cont_wind=(struct Window *)OpenWindow(&control_window);
353: }
354:
355: input()
356: {
357: struct IntuiMessage * inp_msg;
358: struct Gadget * inp_gad;
359: int abbruch = FALSE;
360:
361: CONTINUE
362: {
363: if(inp_msg=(struct IntuiMessage*)GetMsg(cont_wind->User
Port))
364: {
365: if((inp_msg->Class) == GADGETDOWN)
366: {
367: inp_gad = (struct Gadget *)inp_msg->IAddress;
368: switch(inp_gad->GadgetID)
```

```

369: {
370:     case 27:abbruch = TRUE;
371:         break;
372:     case 12:xdreher+= 1;
373:         write_int(xdreher,26L,rp_3);
374:         break;
375:     case 1:xdreher-= 1;
376:         write_int(xdreher,26L,rp_3);
377:         break;
378:     case 13:ydreher+= 1;
379:         write_int(ydreher,38L,rp_3);
380:         break;
381:     case 2:ydreher-= 1;
382:         write_int(ydreher,38L,rp_3);
383:         break;
384:     case 14:zdreher+= 1;
385:         write_int(zdreher,50L,rp_3);
386:         break;
387:     case 3:zdreher-= 1;
388:         write_int(zdreher,50L,rp_3);
389:         break;
390:     case 18:xdreher2+= 1;
391:         write_int(xdreher2,98L,rp_3);
392:         break;
393:     case 7:xdreher2-= 1;
394:         write_int(xdreher2,98L,rp_3);
395:         break;
396:     case 19:ydreher2+= 1;
397:         write_int(ydreher2,110L,rp_3);
398:         break;
399:     case 8:ydreher2-= 1;
400:         write_int(ydreher2,110L,rp_3);
401:         break;
402:     case 20:zdreher2+= 1;
403:         write_int(zdreher2,122L,rp_3);
404:         break;
405:     case 9:zdreher2-= 1;
406:         write_int(zdreher2,122L,rp_3);
407:         break;
408:     case 15:if(xver<1500) xver+= 50;
409:         write_int(xver,62L,rp_3);
410:         break;
411:     case 4:if(xver>-1500) xver-= 50;
412:         write_int(xver,62L,rp_3);
413:         break;
414:     case 16:if(yver<1500) yver+= 50;
415:         write_int(yver,74L,rp_3);
416:         break;
417:     case 5:if(yver>-1500) yver-= 50;
418:         write_int(yver,74L,rp_3);
419:         break;
420:     case 17:if(zver<1500) zver+= 50;
421:         write_int(zver,86L,rp_3);
422:         break;
423:     case 6:if(zver>-1500) zver-= 50;
424:         write_int(zver,86L,rp_3);
425:         break;
426:     case 10:if(zoom>-1500) zoom-= 10;
427:         write_int(zoom,134L,rp_3);
428:         break;
429:     case 31:if(zoom>-1500) zoom-= 100;
430:         write_int(zoom,134L,rp_3);
431:         break;
432:     case 21:if(zoom<1500) zoom+= 10;
433:         write_int(zoom,134L,rp_3);
434:         break;
435:     case 29:if(zoom<1500) zoom+= 100;
436:         write_int(zoom,134L,rp_3);
437:         break;
438:     case 11:if(bild_counter>0) bild_counter-= 2;
439:         write_int(bild_counter,146L,rp_3);
440:         break;
441:     case 32:if(bild_counter>18) bild_counter-= 20;
442:         write_int(bild_counter,146L,rp_3);
443:         break;
444:     case 22: bild_counter+= 2;
445:         write_int(bild_counter,146L,rp_3);
446:         break;
447:     case 30: bild_counter+= 20;
448:         write_int(bild_counter,146L,rp_3);
449:         break;
450:     case 23:filename = (char *)&buff_1[0];
451:         object_loader();
452:         break;
453:     case 24:filename = (char *)&buff_3[0];
454:         film_loader();
455:         break;
456:     case 25:filename = (char *)&buff_3[0];
457:         film_save();
458:         break;
459:     case 26:movie();
460:         Delay(15L);
461:         init_control();
462:         break;
463:     case 28:object_drawer();
464:         Delay(15L);
465:         init_control();
466:         break;
467:     case 33:check_count = 4;
468:         pcnt = 0;
469:         break;
470:     case 34:reset_val();
471:         break;
472:     case 35:if(!seq_deleted) deleter();
473:         break;
474:     case 36:reset_inp();
475:         init_control();
476:         break;
477:     default:break;
478: }
479: }
480: }
481: if(abbruch) break;
482: }
483: }
484:
485: object_drawer()
486: {
487:     int loop,pind,breaker = FALSE,go_on = TRUE ;
488:
489:     prev_xturn = xturn;
490:     prev_yturn = yturn;
491:     prev_zturn = zturn;
492:     prev_xturn2 = xturn2;
493:     prev_yturn2 = yturn2;
494:     prev_zturn2 = zturn2;
495:     prev_xmove = xmove;
496:     prev_ymove = ymove;
497:     prev_zmove = zmove;
498:     prev_zoomer = zoomer;
499:     prev_check_count = check_count;
500:     prev_pcnt = pcnt;
501:     seq_deleted = FALSE;
502:     WindowToBack(cont_wind);
503:     while(bild_counter>0)
504:     {
505:         breaker = transform();
506:         if(breaker) go_on =
507:             AutoRequest(window_2,&r_text1,NULL,&c_text,NULL,NULL,1
508:                 30L,58L);
509:         if(!go_on) break;
510:         BltClear(rp_1->BitMap->Planes[0],8000L,0L);
511:         BltClear(rp_1->BitMap->Planes[1],8000L,0L);
512:         BltClear(rp_1->BitMap->Planes[2],8000L,0L);
513:         BltClear(rp_1->BitMap->Planes[3],8000L,0L);
514:         for(loop=0;loop<flaech_zahl;loop++)
515:         {
516:             if(sichtbar[loop])
517:             {
518:                 SetAPen(rp_1,farbe[loop]);
519:                 SetOPen(rp_1,afarbe[loop]);
520:                 AreaMove(rp_1,sp[plst[loop][0]],ze[plst[loop][0]]);
521:                 for(pind=1;pind<pkt_p_flaech[loop];pind++)
522:                     AreaDraw(rp_1,sp[plst[loop][pind]],ze[plst[loop][pind]]);
523:                 AreaEnd(rp_1);
524:                 sichtbar[loop] = FALSE;
525:             }
526:         }
527:         ScreenToFront(screen_1);
528:         breaker = transform();
529:         BltClear(rp_2->BitMap->Planes[0],8000L,0L);
530:         BltClear(rp_2->BitMap->Planes[1],8000L,0L);
531:         BltClear(rp_2->BitMap->Planes[2],8000L,0L);
532:         BltClear(rp_2->BitMap->Planes[3],8000L,0L);
533:         for(loop=0;loop<flaech_zahl;loop++)
534:         {
535:             if(sichtbar[loop])
536:             {
537:                 SetAPen(rp_2,farbe[loop]);
538:                 SetOPen(rp_2,afarbe[loop]);
539:                 AreaMove(rp_2,sp[plst[loop][0]],ze[plst[loop][0]]);
540:                 for(pind=1;pind<pkt_p_flaech[loop];pind++)
541:                     AreaDraw(rp_2 ,sp[plst[loop][pind]],ze[plst[loop][pind]]);
542:                 AreaEnd(rp_2 );

```

SOFTWARE

```
542:   sichtbar[loop] = FALSE;
543:   }
544: }
545: ScreenToFront(screen_2);
546: bild_counter-= 2;
547: }
548: BltClear(rp_2->BitMap->Planes[0],8000L,0L);
549: BltClear(rp_2->BitMap->Planes[1],8000L,0L);
550: BltClear(rp_2->BitMap->Planes[2],8000L,0L);
551: BltClear(rp_2->BitMap->Planes[3],8000L,0L);
552: WindowToFront(cont_wind);
553: }
554: }
555: init_trig()
556: {
557:   int loop;
558: }
559: for(loop=0;loop<360;loop++)
560: {
561:   sinus[loop] = (float)(sin(loop*PI/180));
562:   cosin[loop] = (float)(cos(loop*PI/180));
563: }
564: }
565: }
566: ausstieg()
567: {
568:   FreeMem(area_plane,25600L);
569:   CloseWindow(window_1);
570:   CloseWindow(window_2);
571:   CloseWindow(cont_wind);
572:   CloseScreen(screen_1);
573:   CloseScreen(screen_2);
574:   CloseLibrary(GfxBase);
575:   CloseLibrary(IntuitionBase);
576:   exit(TRUE);
577: }
578: }
579: transform()
580: {
581:   int loop,anz_sichtbar,inloop,array_voll = FALSE;
582:   long horbuf,verbuf;
583:   float horpos,verpos,z,x,y,hz,hx,hy;
584:   float va_x,va_y,va_z,vb_x,vb_y,vb_z;
585:   float trans_x[MAXP],trans_y[MAXP],trans_z[MAXP];
586:   float normv_x,normv_y,normv_z,s_prod;
587:   float sichtv_x,sichtv_y,sichtv_z;
588: }
589:   xturn+= xdreher;
590:   if(xturn<0) xturn+= 360;
591:   if(xturn>359) xturn-= 360;
592:   yturn+= ydreher;
593:   if(yturn<0) yturn+= 360;
594:   if(yturn>359) yturn-= 360;
595:   zturn+= zdreher;
596:   if(zturn<0) zturn+= 360;
597:   if(zturn>359) zturn-= 360;
598:   xmove+= (float)xver;
599:   ymove+= (float)yver;
600:   zmove+= (float)zver;
601:   xturn2+= xdreher2;
602:   if(xturn2<0) xturn2+= 360;
603:   if(xturn2>359) xturn2-= 360;
604:   yturn2+= ydreher2;
605:   if(yturn2<0) yturn2+= 360;
606:   if(yturn2>359) yturn2-= 360;
607:   zturn2+= zdreher2;
608:   if(zturn2<0) zturn2+= 360;
609:   if(zturn2>359) zturn2-= 360;
610:   zoomer+= (float)zoom;
611:   anz_sichtbar = 0;
612:   for(loop=0;loop<pkt_zahl;loop++)
613:   {
614:     hx = (float)cx[loop];
615:     hy = (float)cosin[xturn]*cy[loop] - sinus[xturn]*cz[loop];
616:     hz = (float)sinus[xturn]*cy[loop] + cosin[xturn]*cz[loop];
617:     x = cosin[yturn]*hx - sinus[yturn]*hz;
618:     y = hy;
619:     z = sinus[yturn]*hx + cosin[yturn]*hz;
620:     hx = cosin[zturn]*x - sinus[zturn]*y;
621:     hy = sinus[zturn]*x + cosin[zturn]*y;
622:     hz = z;
623:     hx+= (float)xmove;
624:     hy+= (float)ymove;
625:     hz+= (float)zmove;
626:     x = hx;
627:     y = cosin[xturn2]*hy - sinus[xturn2]*hz;
628:     z = sinus[xturn2]*hy + cosin[xturn2]*hz;
629:     hx = cosin[yturn2]*x - sinus[yturn2]*z;
630:     hy = y;
631:     hz = sinus[yturn2]*x + cosin[yturn2]*z;
632:     x = cosin[zturn2]*hx - sinus[zturn2]*hy;
633:     y = sinus[zturn2]*hx + cosin[zturn2]*hy;
634:     z = hz;
635:     z = (z-(FAK-zoomer));
636:     if((z>-1.0) && (z<=0.0)) z = -1.0;
637:     if((z>0.0) && (z<1.0)) z = 1.0;
638:     trans_x[loop] = x;
639:     trans_y[loop] = y;
640:     trans_z[loop] = z;
641:     horpos = (-x/z)*256.0;
642:     verpos = (-y/z)*256.0;
643:     sp[loop] = (long)(160.0+horpos);
644:     if(sp[loop]>414L) sp[loop] = 414L;
645:     if(sp[loop]<-94L) sp[loop] = -94L;
646:     ze[loop] = (long)(100.0+verpos);
647:     if(ze[loop]>298L) ze[loop] = 298L;
648:     if(ze[loop]<-98L) ze[loop] = -98L;
649:   }
650:   for(loop=0;loop<flaech_zahl;loop++)
651:   {
652:     va_x = trans_x[plst[loop][0]] - trans_x[plst[loop][1]];
653:     va_y = trans_y[plst[loop][0]] - trans_y[plst[loop][1]];
654:     va_z = trans_z[plst[loop][0]] - trans_z[plst[loop][1]];
655:     vb_x = trans_x[plst[loop][2]] - trans_x[plst[loop][1]];
656:     vb_y = trans_y[plst[loop][2]] - trans_y[plst[loop][1]];
657:     vb_z = trans_z[plst[loop][2]] - trans_z[plst[loop][1]];
658:     normv_x = ((va_y * vb_z) - (va_z * vb_y));
659:     normv_y = ((va_x * vb_z) - (va_z * vb_x));
660:     normv_z = ((va_x * vb_y) - (va_y * vb_x));
661:     sichtv_x = -(trans_x[plst[loop][1]]);
662:     sichtv_y = (trans_y[plst[loop][1]]);
663:     sichtv_z = -(trans_z[plst[loop][1]]);
664:     s_prod = ((sichtv_x*normv_x)+(sichtv_y*normv_y)+
665:               (sichtv_z*normv_z));
666:     if(s_prod<0.0)
667:     {
668:       sichtbar[loop] = TRUE;
669:       anz_sichtbar+= 1;
670:     }
671:   }
672:   if((check_count<11900) && (pcnt<16184))
673:   {
674:     control_lst[check_count] = anz_sichtbar;
675:     check_count+= 1;
676:     for(loop=0;loop<flaech_zahl;loop++)
677:     {
678:       if(sichtbar[loop])
679:       {
680:         control_lst[check_count] = pkt_p_flaech[loop];
681:         check_count+= 1;
682:         control_lst[check_count] = (int)farbe[loop];
683:         check_count+= 1;
684:         control_lst[check_count] = (int)afarbe[loop];
685:         check_count+= 1;
686:         for(inloop=0;inloop<pkt_p_flaech[loop];inloop++)
687:         {
688:           horbuf = (sp[plst[loop][inloop]]-160L);
689:           verbuf = (ze[plst[loop][inloop]]-100L);
690:           xlst[pcnt] = (int)horbuf;
691:           ylst[pcnt] = (int)verbuf;
692:           pcnt+= 1;
693:         }
694:       }
695:     }
696:   }
697:   else array_voll = TRUE;
698:   return(array_voll);
699: }
700: }
701: object_loader()
702: {
703:   int count,loop,length;
704:   char diskfile[18], *append = ".3d\0";
705:   struct FileHandle *load_point;
706: }
707:   length = strlen(filename);
708:   for(loop=0;loop<13;loop++)
709:     diskfile[loop] = filename[loop];
```

```

710: for(loop=length;loop<(length+4);loop++)
711:   diskfile[loop] = append[loop-length];
712: load_point = Open(&diskfile[0],MODE_OLDFILE);
713: if(load_point)
714: {
715:   pkt_zahl = lies_int(load_point);
716:   flaech_zahl = lies_int(load_point);
717:   for(count=0;count<pkt_zahl;count++)
718:   {
719:     cx[count] = (long)lies_int(load_point);
720:     cy[count] = (long)lies_int(load_point);
721:     cz[count] = (long)lies_int(load_point);
722:   }
723:   for(count=0;count<flaech_zahl;count++)
724:     pkt_p_flaech[count] = lies_int(load_point);
725:   for(count=0;count<flaech_zahl;count++)
726:     for(loop=0;loop<pkt_p_flaech[count];loop++)
727:       plst [count][loop] = lies_int(load_point);
728:   for(count=0;count<flaech_zahl;count++)
729:     farbe[count] = (long)lies_int(load_point);
730:   for(count=0;count<flaech_zahl;count++)
731:     afarbe[count] = (long)lies_int(load_point);
732:   Close(load_point);
733: }
734: else
735: {
736:   AutoRequest(cont_wind,&r_text3,NULL,&n_text,NULL,NULL,1
40L,58L);
737:   init_control();
738: }
739: }
740:
741: int lies_int(file_point)
742: struct FileHandle *file_point;
743: {
744:   int wert,loop,pos_zaehl;
745:   pos_zaehl = 0;
746:
747:   do
748:   {
749:     Read(file_point,zielchar,1L);
750:     if(zielchar[0] == '-')
751:     {
752:       zielstring[pos_zaehl] = zielchar[0];
753:       ++pos_zaehl;
754:     }
755:     if(isdigit((int)zielchar[0]))
756:     {
757:       zielstring[pos_zaehl] = zielchar[0];
758:       ++pos_zaehl;
759:     }
760:   }
761:   while((zielchar[0] != ';') && (zielchar[0] != '\0'));
762:   wert = atoi(zielstring);
763:   for(loop=0;loop<10;loop++)
764:     zielstring[loop] = '\0';
765:   return(wert);
766: }
767:
768: write_int(int_wert,output_ypos,ziel_port)
769: int int_wert;
770: long output_ypos;
771: struct RastPort *ziel_port;
772: {
773:   char output_string[8];
774:   int int_wert_buffer,zahl,s_factor,dcnt;
775:   int pos_zaehl,int_wert_stellen;
776:   pos_zaehl = 0;
777:   int_wert_stellen = 5;
778:   if(int_wert<0)
779:   {
780:     output_string[pos_zaehl++] = '-';
781:     int_wert_buffer = -int_wert;
782:   }
783:   else
784:     int_wert_buffer = int_wert;
785:   for(dcnt=10000;dcnt>=1;dcnt/=10)
786:   {
787:     if(int_wert_buffer/dcnt)break;
788:     int_wert_stellen -= 1;
789:   }
790:   switch(int_wert_stellen)
791:   {
792:     case 5:s_factor = int_wert_buffer / 10000;
793:           output_string[pos_zaehl++] = 48 + s_factor;
794:           int_wert_buffer -= s_factor * 10000;
795:     case 4:s_factor = int_wert_buffer / 1000;
796:           output_string[pos_zaehl++] = 48 + s_factor;
797:           int_wert_buffer -= s_factor * 1000;
798:     case 3:s_factor = int_wert_buffer / 100;
799:           output_string[pos_zaehl++] = 48 + s_factor;
800:           int_wert_buffer -= s_factor * 100;
801:     case 2:s_factor = int_wert_buffer / 10;
802:           output_string[pos_zaehl++] = 48 + s_factor;
803:           int_wert_buffer -= s_factor * 10;
804:     case 1:s_factor = int_wert_buffer;
805:           output_string[pos_zaehl++] = 48 + s_factor;
806:           int_wert_buffer -= s_factor;
807:           break;
808:     case 0:output_string[pos_zaehl++] = 48;
809:   }
810:   output_string[pos_zaehl] = '\0';
811:   Move(ziel_port,95L,output_ypos);
812:   Text(ziel_port,"",6L);
813:   Move(ziel_port,(long)(143-(8*pos_zaehl)),output_ypos);
814:   Text(ziel_port,&output_string[0],(long)pos_zaehl);
815: }
816:
817: film_save()
818: {
819:   int *checkpoint,length,loop,save_it;
820:   char diskfile[19], *append = ".mov\0";
821:   struct FileHandle* save_pnt;
822:
823:   save_it = AutoRequest(cont_wind,&r_text5,&s_text,&e_text
,
824:                       NULL,NULL,145L,58L);
825:   init_control();
826:   if(save_it)
827:   {
828:     length = strlen(filename);
829:     for(loop=0;loop<13;loop++)
830:       diskfile[loop] = filename[loop];
831:     for(loop=length;loop<(length+5);loop++)
832:       diskfile[loop] = append[loop-length];
833:     checkpoint = (int *)&control_lst[0];
834:     *checkpoint = check_count;
835:     checkpoint = (int *)&control_lst[2];
836:     *checkpoint = pcnt;
837:     save_pnt = Open(&diskfile[0],MODE_NEWFILE);
838:     length = Write(save_pnt,&control_lst[0],(long)check_cou
nt);
839:     length = Write(save_pnt,&xlst[0],(long)pcnt*2);
840:     length = Write(save_pnt,&ylst[0],(long)pcnt*2);
841:     Close(save_pnt);
842:   }
843: }
844:
845: film_loader()
846: {
847:   int length,loop,load_it;
848:   char diskfile[19], *append = ".mov\0";
849:   struct FileHandle *load_pnt;
850:
851:   load_it = AutoRequest(cont_wind,&r_text7,&o_text,&e_text
,
852:                       NULL,NULL,145L,58L);
853:   init_control();
854:   if(load_it)
855:   {
856:     length = strlen(filename);
857:     for(loop=0;loop<13;loop++)
858:       diskfile[loop] = filename[loop];
859:     for(loop=length;loop<(length+5);loop++)
860:       diskfile[loop] = append[loop-length];
861:     load_pnt = Open(&diskfile[0],MODE_OLDFILE);
862:     if(load_pnt)
863:     {
864:       length = Read(load_pnt,&check_count,2L);
865:       length = Read(load_pnt,&pcnt,2L);
866:       length = Read(load_pnt,&control_lst[4],(long)(check_co
unt-4));
867:       length = Read(load_pnt,&xlst[0],(long)pcnt*2);
868:       length = Read(load_pnt,&ylst[0],(long)pcnt*2);
869:       Close(load_pnt);
870:       seq_deleted = TRUE;
871:     }
872:   }
873:   else
874:   {
875:     AutoRequest(cont_wind,&r_text3,NULL,&n_text,
876:               NULL,NULL,140L,58L);
877:     init_control();
878:   }
879: }
880:

```

```

881: movie()
882: {
883:   int control_cnt = 4 , pcnt = 0;
884:   int loop, inloop, flaechen, punkte;
885:   long fill_color, outline_color;
886:
887:   WindowToBack(cont_wind);
888:   Delay(50L);
889:   while(control_cnt < check_count)
890:   {
891:     BltClear(rp_1->BitMap->Planes[0], 8000L, 0L);
892:     BltClear(rp_1->BitMap->Planes[1], 8000L, 0L);
893:     BltClear(rp_1->BitMap->Planes[2], 8000L, 0L);
894:     BltClear(rp_1->BitMap->Planes[3], 8000L, 0L);
895:     flaechen = control_lst[control_cnt];
896:     control_cnt++;
897:     for(loop=0; loop < flaechen; loop++)
898:     {
899:       punkte = control_lst[control_cnt];
900:       control_cnt++;
901:       fill_color = (long)control_lst[control_cnt];
902:       control_cnt++;
903:       outline_color = (long)control_lst[control_cnt];
904:       control_cnt++;
905:       SetAPen(rp_1, fill_color);
906:       SetOPen(rp_1, outline_color);
907:       AreaMove(rp_1, (long)(xlst[pcnt]+160L), (long)(ylst[pcnt]
+100));
908:       pcnt++;
909:       for(inloop=0; inloop < (punkte-1); inloop++)
910:       {
911:         AreaDraw(rp_1, (long)(xlst[pcnt]+160L), (long)(ylst[pcn
t]+100));
912:         pcnt++;
913:       }
914:       AreaEnd(rp_1);
915:     }
916:     ScreenToFront(screen_1);
917:     BltClear(rp_2->BitMap->Planes[0], 8000L, 0L);
918:     BltClear(rp_2->BitMap->Planes[1], 8000L, 0L);
919:     BltClear(rp_2->BitMap->Planes[2], 8000L, 0L);
920:     BltClear(rp_2->BitMap->Planes[3], 8000L, 0L);
921:     flaechen = control_lst[control_cnt];
922:     control_cnt++;
923:     for(loop=0; loop < flaechen; loop++)
924:     {
925:       punkte = control_lst[control_cnt];
926:       control_cnt++;
927:       fill_color = (long)control_lst[control_cnt];
928:       control_cnt++;
929:       outline_color = (long)control_lst[control_cnt];
930:       control_cnt++;
931:       SetAPen(rp_2, fill_color);
932:       SetOPen(rp_2, outline_color);
933:       AreaMove(rp_2, (long)(xlst[pcnt]+160L), (long)(ylst[pcnt]
+100));
934:       pcnt++;
935:       for(inloop=0; inloop < (punkte-1); inloop++)
936:       {
937:         AreaDraw(rp_2, (long)(xlst[pcnt]+160L), (long)(ylst[pcn
t]+100));
938:         pcnt++;
939:       }
940:       AreaEnd(rp_2);
941:     }
942:     ScreenToFront(screen_2);
943:   }
944:   Delay(50L);
945:   BltClear(rp_2->BitMap->Planes[0], 8000L, 0L);
946:   BltClear(rp_2->BitMap->Planes[1], 8000L, 0L);
947:   BltClear(rp_2->BitMap->Planes[2], 8000L, 0L);
948:   BltClear(rp_2->BitMap->Planes[3], 8000L, 0L);
949:   WindowToFront(cont_wind);
950: }
951:
952: init_control()
953: {
954:   SetAPen(rp_3, 1L);
955:   Move(rp_3, 10L, 26L);
956:   Text(rp_3, "x-Turn:", 7L);
957:   Move(rp_3, 10L, 38L);
958:   Text(rp_3, "y-Turn:", 7L);
959:   Move(rp_3, 10L, 50L);
960:   Text(rp_3, "z-Turn:", 7L);
961:   Move(rp_3, 10L, 62L);
962:   Text(rp_3, "x-Move:", 7L);
963:   Move(rp_3, 10L, 74L);
964:   Text(rp_3, "y-Move:", 7L);
965:   Move(rp_3, 10L, 86L);
966:   Text(rp_3, "z-Move:", 7L);
967:   Move(rp_3, 10L, 98L);
968:   Text(rp_3, "x-Cent:", 7L);
969:   Move(rp_3, 10L, 110L);
970:   Text(rp_3, "y-Cent:", 7L);
971:   Move(rp_3, 10L, 122L);
972:   Text(rp_3, "z-Cent:", 7L);
973:   Move(rp_3, 10L, 134L);
974:   Text(rp_3, "Zoom:", 5L);
975:   Move(rp_3, 10L, 146L);
976:   Text(rp_3, "Bilder:", 7L);
977:   write_int(xdreher, 26L, rp_3);
978:   write_int(ydreher, 38L, rp_3);
979:   write_int(zdreher, 50L, rp_3);
980:   write_int(xver, 62L, rp_3);
981:   write_int(yver, 74L, rp_3);
982:   write_int(zver, 86L, rp_3);
983:   write_int(xdreher2, 98L, rp_3);
984:   write_int(ydreher2, 110L, rp_3);
985:   write_int(zdreher2, 122L, rp_3);
986:   write_int(zoom, 134L, rp_3);
987:   write_int(bild_counter, 146L, rp_3);
988: }
989:
990: reset_val()
991: {
992:   xturn = 0;
993:   yturn = 0;
994:   zturn = 0;
995:   xturn2 = 0;
996:   yturn2 = 0;
997:   zturn2 = 0;
998:   xmove = 0;
999:   ymove = 0;
1000:   zmove = 0;
1001:   zoomer = 0;
1002: }
1003:
1004: reset_inp()
1005: {
1006:   xdreher = 0;
1007:   ydreher = 0;
1008:   zdreher = 0;
1009:   xdreher2 = 0;
1010:   ydreher2 = 0;
1011:   zdreher2 = 0;
1012:   xver = 0;
1013:   yver = 0;
1014:   zver = 0;
1015:   zoom = 0;
1016: }
1017:
1018: deleter()
1019: {
1020:   xturn = prev_xturn;
1021:   yturn = prev_yturn;
1022:   zturn = prev_zturn;
1023:   xturn2 = prev_xturn2;
1024:   yturn2 = prev_yturn2;
1025:   zturn2 = prev_zturn2;
1026:   xmove = prev_xmove;
1027:   ymove = prev_ymove;
1028:   zmove = prev_zmove;
1029:   zoomer = prev_zoomer;
1030:   check_count = prev_check_count;
1031:   pcnt = prev_pcnt;
1032:   seq_deleted = TRUE;
1033: }

```

1000 Berlin



RUNOW
Büroelektronik
Keithstraße 26 · 1000 Berlin 30
☎ 26 111 26

2000 Hamburg

Bit Computer Shop
Osterstraße 173 · 2000 Hamburg 20
Telefon: 040/494400

Createam
Computer Hard & Software
Bramfelder Chaussee 300 · 2000 Hamburg 71
Telefon Sa. Nr. 040/6415091

Gerhard u. Bernd Waller GbR
Computer & Zubehör-Shop
Kieler Straße 623
2000 Hamburg 54
☎ 040/570 60 07 + 570 52 75



GMA
Systemhändler
Wandsbeker Chaussee 58
2000 Hamburg 76

Computer
Hardware · Software · Zubehör



Lilienstraße 32
(beim Mönckebergbrunnen)
2000 Hamburg 1
Tel. 040/338616

SYSTEMSHOP

2160 Stade



BERGBAU
Büromaschinen · EDV-Systeme
Neue Straße 5, 2160 Stade
Telefon: (04141) 23 64 + 23 84

2390 Flensburg



ecl electronic
computer
laden ohg
Norderstr 94-96 D-2390 Flensburg
(0461) 28181 & 28193

2800 Bremen

Berthge & Strutz KG
St.-Jürgen-Straße 46-50
2800 Bremen 1
Tel. 04 21- 70 00 57-59

2940 Wilhelmshaven

Radio Tiemann
Commodore-Systemfachhändler
Markstr. 52
2940 Wilhelmshaven
Telefon 0 44 21- 2 61 45

2960 Aurich

FRANZ-JOSEF KIRFEL
GEORGSWALL 18a
2960 AURICH 1
TELEFON 0 49 41- 6 23 55

3000 Hannover

COM DATA
Am Schiffgraben 19 · 3000 Hannover 1
Telefon 05 11- 32 67 36

3250 Hammeln

Witte Bürotechnik
Inh. Günther Kessels
Deisterstr. 53
3250 Hameln 1
Tel. 0 51 51- 1 20 22/ 1 20 23

3470 Höxter

Schidlack + Sohn KG
Am Markt 8 · 3470 Höxter
Telefon 0 52 71- 79 29/ 29 29

3500 Kassel

Hermann Fischer GmbH
Commodore-Systemfachhändler
Rudolf-Schwander-Str. 5-13
3500 Kassel
Tel. (05 61) 70 00 00



3250 Hameln

CCS Commodore
Computer Schule GmbH
Immermannstr. 65
4000 Düsseldorf 1
Tel. 02 11- 35 32 15

Hier könnte
Ihre Anzeige
erscheinen
Anruf genügt: Heim - Verlag
Tel. 0 61 51 - 5 60 57

4200 Oberhausen

LaSch
das Buch und Software Haus
Inh. Rainer Langner u. Franz Schnitzler GbR
Nohlstraße 76 · 4200 Oberhausen 1
Telefon 02 08/ 80 90 14

4504 GM.-Hütte



jacor
Computershop
Auf geht's
Im Sinus-Elektrofachmarkt
Niedersachsenstr. 9
4504 GM.-Hütte
Tel. 0 54 01/ 4 54 41

4600 Dortmund

Elektronik
Computer
Fachliteratur
AMIGA-Software
4600 Dortmund 1, Güntherstraße 75, Tel. (02 31) 57 22 84



city-elektronik

4650 Gelsenkirchen-Horst



Comtron
Hard- und Software, Literatur
Bauteile, Service, Versand
Groß- und Einzelhandel
Poststr 15 · 4650 Gelsenkirchen-Horst
Tel. 0209/52572

4770 Soest



jacor
Computershop
Auf geht's
In der Familia-Passage
Senator-Schwartz-Ring 24
4770 Soest · Tel. 0 29 21/ 6 38 03

4800 Bielefeld

hardware
software
organisation
service

CSF

CSF COMPUTER & SOFTWARE GMBH
Heeper Straße 106-108
4800 Bielefeld 1
Tel. (05 21) 6 16 63

5000 Köln

BÜROMASCHINEN
braun

AM RUDOLFPLATZ GmbH
5000 KÖLN 1
RICHARD-WAGNER-STR. 39
RUF: 02 21/219171

5060 Bergisch-Gladbach

Computer Center

Buchholzstraße 1
5060 Bergisch-Gladbach
Telefon 0 22 02 - 3 50 53

5200 Siegburg

Computer Center

Luisenstraße 26
5200 Siegburg
Telefon 0 22 41/6 68 54

5500 Trier

bürocenter
LEHR

Güterstr. 82 - 5500 Trier
☎ 06 51 - 2 50 44

Fordern Sie unsere Zubehör-Liste an.

5768 Sundern

C.S.C.

Computer & Software-Center

Hauptstr. 2 · 5768 Sundern
Telefon 0 29 33- 20 46

6000 Frankfurt

GES-COMPUTER
GESELLSCHAFT FÜR EDV UND SOFTWARE mbH

Filiale Frankfurt
Hartmann-Ibach-Str. 63
6000 Frankfurt 60
Tel.: (0 69) 46 20 41

Filiale Hanau
Steinheimer Str. 22
6450 Hanau
Tel.: (0 61 81) 2 48 26

6000 Frankfurt

Hardware
Software
Service

Computer
bc

Büro-Computer + Organisations GmbH

Commodore
TOSHIBA
ATARI OKI DATA

Ihr Partner,
wenn es um
Computer geht

6000 FRANKFURT/M. 1, Deder Weg 7-9, ☎ 069/55 04 56/57

6200 Wiesbaden

Henneveld KG

Schossbergstr. 21
6200 Wiesbaden
Tel. 0 61 21 - 27 70

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



Werbung und EDV GmbH

AUTORISIERTER
COMMODORE
SYSTEM-HÄNDLER



6250 Limburg

wir
bürosysteme vertriebs gmbh

diezer strasse 10
6250 limburg
tel. 0 64 31 - 2 00 30

6380 Bad-Homburg

PDC GmbH
Produkte u. Details Computerverbund

Luisenstr. 115
Ladenpassage Alter Bahnhof
6380 Bad-Homburg
Tel. 0 61 72 - 2 47 48

Bad-Homburgs erster Commodore Computerladen

6457 Maintal

Landolt-Computer

Beratung · Service · Verkauf · Leasing

Commodore Autorisierter Commodore-Händler

Wingertstr. 112 · 6457 Maintal/Dörnigheim
Telefon 0 61 81 - 4 52 93

6500 Mainz

Henneveld KG

Münsterstr. 15
6500 Mainz
Tel. 0 61 31 - 24 01

6520 Worms

Georg Steinmetz oHG

Neumarkt 4 + 10 · 6520 Worms
Telefon 0 62 41 - 68 68

6680 Neunkirchen

Shop 64

Computer GmbH

Saarbrücken * Saarlouis
Homburg * St. Ingbert

Neunkirchen
06821/23713
Commodore Systemhändler

6750 Kaiserslautern



6800 Mannheim

GAUCHI+STURM

Computersysteme + Textsysteme

6800 Mannheim 24

Casterfeldstraße 74-76
☎ (06 21) 85 00 40 · Teletex 6 211 912

Computer-Center
am Hauptbahnhof GmbH

L 14, 16-17
6800 Mannheim 1
Tel. (06 21) 2 09 83/84

6900 Heidelberg

dacor
Computershop

Auf geht's

Im Familia-Center
Hertzstraße 1 · 6900 Heidelberg
Tel. 0 62 21 / 30 24 37

6940 Weinheim

dacor
Computershop

Auf geht's

Im Mult-Zentrum
Berliner Platz 1 · 6940 Weinheim
Tel. 0 62 01 / 60 01 89

8460 Schwandorf

Büro- u. Datensysteme GmbH

Ettmannsdorfer Str. 8
8460 Schwandorf
Tel. (0 94 31) 2 04 78 / 4 18 43

8500 Nürnberg

Orgaplus Datenverwaltung G. Gailer KG

Fürther Str. 54 - 56
8500 Nürnberg 80
☎ 09 11 - 27 06 20

8600 Bamberg

BÜRO- ZENTRUM A+R KUTZ

Am Kranen Tel. 09 51 / 2 78 08
8600 Bamberg

8700 Würzburg

SCHILL

BÜROTEAM

Hardware · Software
Service · Schulung
computer center
am Domlnkanerplatz
Ruf (09 31) 5 04 88

Schweiz

CH-8021 Zürich

Senn Computer AG



Langstrasse 31
Postfach CH-8021 Zürich
Tel. 01 / 241 73 73
Telex 814 193 seco

CH-4054 Basel

Wir sind Amiga-Freaks.

Unsere Öffnungszeiten:
Von Dienstag bis Freitag
9.30 - 12.30 und 14.00 - 18.30 und
am Samstag 9.30 - 16.00

SYSAG

COMPUTERCENTER

Basel: Tel. 061/39 25 25 · Holestrasse 87 · 4054 Basel
Aarau: Tel. 064/22 63 33 · Kasernenstrasse 26 · 5000 Aarau

7000 Stuttgart

»If AMIGA, go to Schreiber«
Stuttgart's starker Computer-Laden.

SCHREIBER COMPUTER

Alte Poststraße 2
7000 Stuttgart-1
Tel. 07 11 / 22 70 99

Neu · Neu · Neu · Neu
Im SUBWAY
BREUNINGER City

7140 Ludwigsburg

B D T

BÜRO-DATEN-TECHNIK-VERTRIEBS GMBH

Kurfürstenstraße 18 · 7140 Ludwigsburg
Telefon 07 141 - 2 50 74

7500 Karlsruhe

MCT

Ges. f. Microcomputer

Troilingerstr. 3 · 7500 Karlsruhe 41
Tel. 07 21 / 47 27 95

7700 Singen

Schellhammer GmbH

Freibühlstr. 21 + 23
7700 Singen
☎ 0 77 31 / 82 02 - 00

7890 Waldshut-Tiengen

hettler-data

service gmbh

Lenzburger Straße 4
7890 Waldshut-Tiengen
Telefon 0 77 51 / 30 94

7900 Ulm

EDV-Systeme Frauenstr. 28
Software- 7900 Ulm/Donau
erstellung Tel. 07 31 / 2 80 76
Schulung Telex 712 973 csulm-d



7920 Heidenheim

Das Büro

Kastorstr. 11
7920 Heidenheim/Brenz
☎ 0 73 21 - 4 40 15

7990 Friedrichshafen

KUMATRONIK Datentechnik

Schwabstr. 19
7990 Friedrichshafen
Telefon 0 75 41 - 7 20 41

8000 München

Ludwig

COMPUTER + BÜROTECHNIK

COMPUTER · SOFTWARE · PERIPHERIE
BERATUNG · TECHN. KUNDENDIENST
INGOLSTÄDTER STR. 62L
EURO-INDUSTRIE-PARK · 8000 MÜNCHEN 45
TELEFON 089 / 3113066 · TELETEX 898341

8120 Weilheim

COMPUTER STUDIO HUTTER GMBH

MÜNCHNER STR. 12 · 8120 WEILHEIM
TELEFON 08 81 / 12 23

8330 Eggenfelden

Hot Space

Computer-Centrum
R. Lanfermann
Schellenbrückstraße 6
8330 Eggenfelden
Telefon 0 87 21 / 65 73

Altöttinger Straße 2
8265 Neuötting
Telefon 0 86 71 / 7 16 10

Innstraße 4
8341 Simbach
Telefon 0 85 71 / 44 10

8390 Passau

Zimmermann elektroland

8390 Passau
Kohlbruck 2a
☎ 08 51 / 5 20 07

8400 Regensburg

Computer-Laden Karl Steinmetz

Gewerbepark C 62 · 8400 Regensburg
Telefon 09 41 - 4 82 99

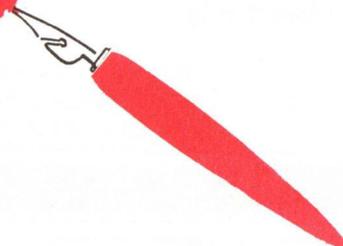
Zimmermann elektroland

8400 Regensburg
Dr.-Gessler-Str. 8
☎ 09 41 / 9 50 85

Es ist da!

VIZAWRITE

Deskriptor



Der erste Eindruck

Nach dem Starten erscheint die Textverarbeitung in angenehmen Farben, aber ansonsten in der vom AMIGA gewohnten Weise, mit Fenstern und einer Menüleiste. Auffällig ist die Kopfleiste, die ähnlich gestaltet ist, wie bei TEXTCRAFT, der Textverarbeitung von Commodore und zeigt die Ausrichtung des Textes und die Tabulatoren. Außerdem enthält sie eine Skala, die im ersten Moment als Spaltenlineal gedeutet werden könnte, das nur 50 Zeichen zuließ. Doch zum Ersten ist es eine Einteilung in Inches, und zum Zweiten benutzt VIZAWRITE auch auf dem Bildschirm Proportionalschrift und somit kann keine Aussage über die Anzahl der Buchstaben, die in eine Bildschirmzeile passen, gemacht werden. Im Schnitt sind es um die 80 Zeichen, die sich damit darstellen lassen.

Textverarbeitung

Weil es nicht fehlen darf, werden an dieser Stelle die 'Standardfunktionen' vorgestellt, die mehr oder weniger zu jeder brauchbaren Textverarbeitung dazugehören. Standardmäßig wird jeder Text zwischen den zwei äußeren Tabulatormarken linksbündig angeordnet. Mit einem Tastendruck ist aber auch Zentrieren, die Ausrichtung nach rechts, oder der, besonders im Zusammenspiel mit der Proportionalschrift, professionelle Blocksatz. Diese Textausrichtung wird auch dann noch eingehalten, wenn mitten in einen Absatz etwas eingefügt wird. Der nachfolgende Text wird immer in die nachfolgenden Zeilen geschoben, so

Schon vor Monaten wurde von dem englischen Softwarehaus VIZA ein neues Textverarbeitungsprogramm für den AMIGA angekündigt, doch der interessierte Kunde sollte sich noch einige Zeit gedulden können. Die Firma VIZA SOFTWARE ist vielen Homecomputerbesitzern schon bekannt, denn vor einigen Jahren hatten Sie das beliebte VIZAWRITE für den COMMODORE 64 herausgebracht. Diese Textverarbeitung war und ist immer noch, eine der Besten für diesen Computer. Nicht zuletzt deswegen, wurde in das neue Produkt für den AMIGA die größten Erwartungen gesteckt – die Werbung tat ein übriges. Doch jetzt muß sie beweisen, welchen Wert sie hat.

daß keine überlangen Zeilen entstehen können.

In der bereits erwähnten Formatzeile, die beliebig oft in einem Text auftreten kann, werden auch die Tabulatoren eingestellt. Es gibt mehrere Arten von Tabulatoren. Rechts, Mitte und Links ordnen den Text entsprechend ihrer Position an. Der Dezimaltabulator setzt bei Zahlen den Dezimalpunkt untereinander.

Obwohl nicht jeder vom Einsatz einer Maus bei der Textverarbeitung angetan ist, so ist sie doch hervorragend geeignet, um Bereiche zu markieren. Ein Doppelklick, und schon ist eine ganze Zeile angewählt. Zusammen mit einer weiteren Taste kann diese Funktion auch auf Sätze, Abschnitte und sogar das ganze Dokument angewendet werden. Ein solcher Bereich ist schnell kopiert, ver-

schohen oder aber gelöscht. Sehr wichtig ist auch die Möglichkeit, die Schriftart, oder den Zeichensatz innerhalb dieser Markierung zu verändern.

Wer schon einmal in einem längeren Text nach einem Wort gesucht hat, das fälschlicherweise falsch geschrieben wurde, der wird es zu schätzen wissen, wenn ihm eine Funktion zum Suchen und Ersetzen zur Verfügung steht. Die Arbeit wird dann, auf Wunsch auch mit einer Rückfrage, selbständig erledigt, wobei der Fehler hoffentlich beseitigt wurde, ohne einen Neuen einzubauen

...

Nicht alltäglich ist die Möglichkeit, Texte mit einem Passwort zu versehen. Dies ist unter Umständen eine nützliche Angelegenheit, wobei man das Passwort nicht vergessen sollte denn sonst ...

Wer innerhalb eines Textes häufig ein bestimmtes Wort oder einen Ausdruck benötigt, der kann diesen als Baustein ablegen. Auch Bilder können in diese Datei übernommen werden. Um einen Baustein in den momentanen Text zu übernehmen, muß man diesen Menüpunkt anwählen, oder sein Fenster hervorholen und das entsprechende Objekt anklicken.

Desktop Publishing

Der Begriff Desktop Publishing wird im Handbuch des öfteren erwähnt, allerdings sollte man von VIZAWRITE auf diesem Gebiet nicht allzu viel erwarten, letztlich erfüllen selbst spezielle Desktop Publisher auf dem AMIGA nicht zufriedenstellend die Wünsche der Anwender.

Sicherlich hat VIZAWRITE ein

paar Funktionen, die in diese Richtung gehen, aber es fehlen doch Dinge wie Mehrspaltenverarbeitung und Textdarstellung neben eingebundenen Grafiken. Gerade für die, in punkto Grafik, verwöhnten AMIGA-Besitzer ist die Einbindung von Bildern und deren Ausdruck inmitten von Text, eine interessante Angelegenheit. VIZAWRITE erlaubt es in diesem Zusammenhang auch, Bilder horizontal oder vertikal zu stauchen bzw. zu dehnen. Die Bilder müssen mit einem IFF-kompatiblen Grafikprogramm erstellt werden, wobei man allerdings auf die Verwendung von farbig ausgefüllten Flächen verzichten sollte, da diese nicht in Grauwerte, sondern lediglich in Schwarz umgewandelt werden. Hoffentlich findet sich in diesem Zusammenhang bald jemand, der ein gutes Konvertierprogramm für Farbbilder in Schwarzweiß erstellt.

Eine ebenfalls sehr schöne Option, ist die Fähigkeit von VIZAWRITE, jeden Zeichensatz des AMIGA darzustellen und zu verarbeiten. Auf der Diskette sind sieben verschiedene Fonts mit zwei bis drei Größen enthalten. Die Größen reichen dabei von 8 bis 20 Punkten, wobei aber auch größere denkbar sind. Mit diesen Zeichensätzen können, zusammen mit den verschiedenen Schriftarten (fett, kursiv, unterstrichen, hoch- und tiefgestellt), schon gute und abwechslungsreiche Gestaltungen vorgenommen werden. Besonders die verschlungenen Schriftarten eignen sich hervorragend dafür, sie für Einladungen oder Informationsschriften zu verwenden.

Ausdruck

Die Möglichkeiten, einen Ausdruck des erstellten Schriftstücks zu erhalten, sind sehr vielfältig. Drei Arten werden dabei prinzipiell unterschieden: mit einer bestimmten Schriftbreite, proportional oder direkt vom Bildschirm. Die ersten beiden Arten haben zur Folge, daß Texte in Normalschrift oder NLQ ausgedruckt werden und Bilder entsprechend im Grafikmodus. Die verschiedenen Zeichensätze des AMIGA verschwinden auf diese Weise genauso, wie die Aufteilung des Bildschirms. Grafik und Text des Aus-



VIZAWRITE, die Textverarbeitung

Es geht auch größer

außerdem stehen die systemzeichensätze zur Verfügung

auch klein ist es fein

hallo hier bin ich

ich bin winzig

Wer kann das noch lesen ?

So, jetzt ist aber Schluß mit dem Test ...

Lange hat man darauf gewartet: VIZAWRITE DESKTOP ist endlich fertig und was noch besser ist, es wird bereits in Deutschland ausgeliefert. Sicherlich hatte der ein oder andere die Hoffnung schon aufgegeben, dieses Programm jemals in Händen zu halten.



drucks, sind allerdings entsprechend den Qualitäten der verwendeten Drucker recht sauber. Doch vor allem überrascht die hohe Geschwindigkeit des Ausdrucks. Der eigentlich wichtigste Ausdruck wird mit der Option 'Bildschirm' erreicht. Hierbei wird alles so ausgegeben, wie es auch auf dem Bildschirm zu sehen ist. Dabei kommen sowohl die kunstvollen Schriftarten, als auch die Grafiken zur Geltung. Auch dieser Ausdruck benötigt nur wenig Zeit, so daß bei einem NEC P6 ein Probeausdruck in einer geringeren Qualität keinen merklichen Zeitgewinn bringt.

Für die Gestaltung des Ausdrucks sind neben den grafischen Elementen auch noch die Kopf- und Fußzeilen zuständig. Ein dort eingegebener Text erscheint auf jeder Seite des Ausdrucks und beinhaltet deshalb Daten, wie Seitennummer, Autorenname und ähnliches. Das Seitenformat, also in erster Linie die Papierlänge und die Breite der Ränder, wird in einem separaten Menü festgelegt. Im eigentlichen Druckmenü bestimmt man dann nur noch den Seitenbereich, die Anzahl der Kopien und die Druckqualität.

Eine sehr leistungsstarke Fähigkeit von VIZAWRITE ist der Ausdruck von Serienbriefen (Mail Merge). Dies bedeutet, daß ein Brief mit verschiedenen Adressen, die aus einer Adressdatei stammen, ausgedruckt wird. Dieser Vorgang geschieht, nach der Vorbereitung des Briefes und der Datei, vollautomatisch und erspart dadurch viel Arbeit und Zeit.

VIZAWRITE

VIZAWRITE DESKTOP gehört mit seinen Fähigkeiten sicherlich zu den zur Zeit besten Textverarbeitungsprogrammen. Besonders die Kombination Blocksatz mit Propor-

- + mehrere unabhängige Dokumente gleichzeitig bearbeiten
- + Proportionalchrift
- + nutzt alle Systemzeichensätze
- + Serienbriefschreibung (Mail Merge)
- + schneller Ausdruck
- + deutsche Anleitung und Menüführung
- + kein Kopierschutz

An Firma
Fritz & Kunze GmbH
Bahnhofsgasse 34
6787 Wundershausen

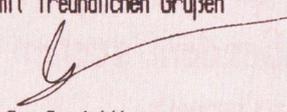
Sehr geehrte Damen und Herren,

mit großem Gefallen nahmen wir Ihre Sommerkollektion entgegen. Besonderes Gefallen fanden unsere Einkäufer an dem Modell 'Tina' und 'Bernhard'. Gerade im Modegeschäft ist es wichtig, die Kunden mit aggressiv neuen Ideen und knalligen Farbe zu überzeugen. Wir bestellen deshalb nach unseren Geschäftsbedingungen folgende Ware:

Menge	Artikel	Preis
100	Tina 36-46	89,90
300	Bernhard 38-58	67,68

Nach Rücksprache mit Herrn von Knüllow, möchten wir Sie bitten uns Ihr Angebot im Bereich Krawatten zu unterbreiten.

mit freundlichen Grüßen



i.A. Cardetti

tionalchrift prägt den guten Standard des Programms. Ein wichtiger Punkt sind auch die verschiedenen Systemzeichensätze und die Fähigkeit, Bilder und Grafiken in den Text zu integrieren. Für Firmen und Vereine ist die Serienbriefschreibung ein hilfreicher Punkt, der einiges an Zeit und Arbeit sparen hilft. Leider hat VIZAWRITE keine Silbentrennung, was jedoch weitgehend durch die Proportionalchrift ausgeglichen wird, weil bei entstehenden Lücken aufgerückt wird. Die Bedienung der gesamten Textverarbeitung ist ein-

fach, die Funktionen sind übersichtlich in Menüs zusammengefaßt und gestatten somit ein einfaches Anwählen. Der leichte Einstieg in das System wird auch durch die Anleitung unterstützt, die übersichtlich gegliedert und sehr flüssig ins Deutsche übersetzt wurde. Dabei zeigt sich, daß es durchaus möglich ist, gleichzeitig mit der Einführung in den deutschen Markt, sowohl eine gute deutsche Anleitung mitzuliefern, als auch die Texte des Programms zu übersetzen.

(MN)

- teilweise etwas langsam
- keine Silbentrennung
- kein Text neben Grafiken
- keine Tastenbelegung
- schlechte Grafikkonvertierung

Hersteller: VIZA SOFTWARE
Anbieter: DTM 0 61 21/56 00 84
IM 0 69/7 07 11 02

Preis: DM 198,-

Konfiguration:
AMIGA 500/1000/2000,
(minimal) ein Laufwerk

Eprommer für alle Amigas lieferbar

Brennt 2716 – 27011 mit Standard und Fast Algorithmus

Software mit Kabel DM 98,-
Eprommer komplett DM 298,-

MTR 512

- 512 KByte statische Ram/Epromkarte
- in 64 KB Schritten aufrüstbar
- durchgeschleifter Bus
- Adressbereich beliebig einstellbar

Der Preis ab DM 98,-

Zweitlaufwerke für alle Amigas ab DM 270,-

Multifunktionskarte

- 72 frei programmierbare I/O Leitungen
- batteriegepufferte Echtzeituhr
- Experimentierfeld
- beliebig anreihbar ab DM 98,-

Software

New CLI Mate DM 76,-
Profimat DM 99,-
Superbase DM 228,-
Beckertext DM 198,-
Textomat DM 99,-



Bald lieferbar: 512 KB Karte mit Uhr für Amiga 500

Ralf Tröps

Computertechnik · Pingsdorferstr. 141 · 5040 Brühl
Tel. 0 22 32 / 1 30 63 und 4 71 05 ☎

PADERCOMP – Walter Ladz

Erzbergerstr. 27 · 4790 Paderborn · Tel. 0 52 51 - 3 63 96

FLOPPYSTATIONEN

PADERCOMP FL 1 **398,-** –
3.5", 1 Mb, eingeb. Netzteil, NEC-Laufwerk, Abm. 240x105x40 mm,
anschlußfertig, graues Metallgeh. Testbericht ATARI-Magazin 2/87,
Seite 70

PADERCOMP FL 2 **698,-** –
Doppellaufwerk übereinander, sonst wie FL 1

PADERCOMP FL 3 **348,-** –
Zweitlaufwerk für Amiga

NEC FD 1036 A 3.5", 1 MB, 32 mm Bauhöhe ... **225,-** –
neuestes Modell, keine Modifizierung für ATARI ST erforderlich.

Industrie Floppystecker 7,90
ST Kabel an Shugart-Bus 3.5" 29,90
Monitorstecker 6,90

Die aktuellen Preise erfragen Sie unter der PCB-Mailbox Tel. 0 52 52-34 13 (300 Bd., 8N1)

DRUCKER

STAR NL 10 incl. Interface, Dt. Handbuch **575,-** –
NEC P6, 24 Nadeln, 216 Z/s, DIN A 4 **1148,-** –
NEC P7, 24 Nadeln, 216 Z/s, DIN A 3 **1498,-** –
STAR NB 24/15, 24 Nadeln, 216 Z/s, DIN A 3 **1798,-** –
Seikosha SL-80AI, 24 Nadeln, 135 Z/s, A 4 **848,-** –

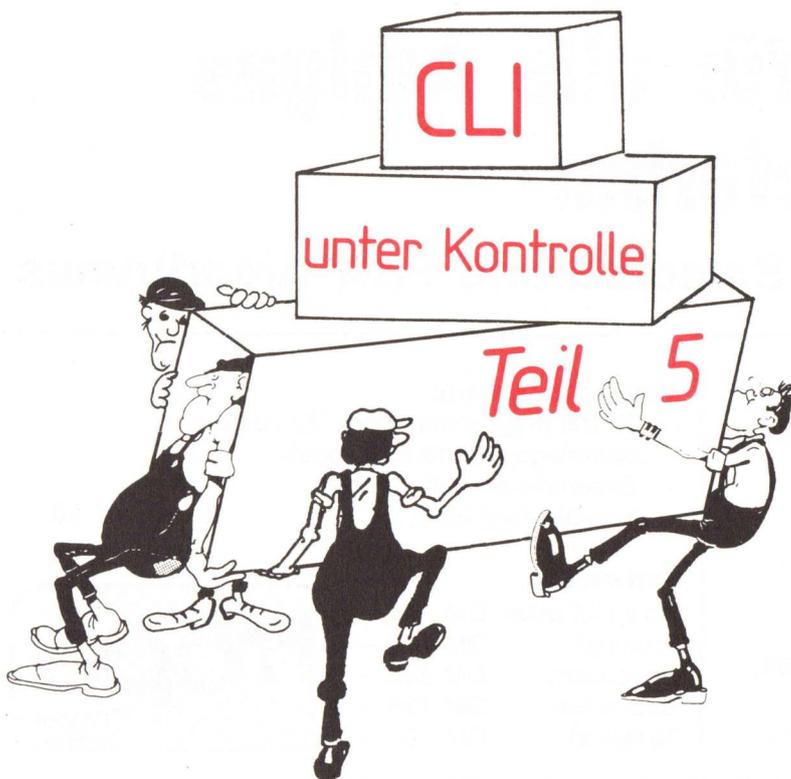
ZUBEHÖR

FL-1 Adapter, zum Anschluß von 2 FL-1 **49,-** –
Multimataic, Cut Sheet Feeder für P6 **598,-** –
Dataphon S21d-2, 300 Baud, FTZ-Nr. **238,-** –
Dataphon S21/23, 300 bzw. 1200/75 Baud, BTX **319,-** –
Disk Box SS-50, f. 50 3.5" Disketten **19,90**
Media Box 1, f. 150 St. 3.5" Disketten **34,90**
Druckerkabel ST **34,90**
Monitor-Ständer dreh-, schwenk- und kippbar **25,-** –
Joy-Star, Super Joystick mit 6 Microschaltern **17,90**
Preisliste **kostenlos**

MONITORE

EIZO Flexscan 8060 S, alle Aufl., kein nachjustieren,
820x620 Punkte, 0.28 dot pitch **1548,-** –
NEC Multisync, alle drei Auflösungen **1398,-** –
Mitsubishi Freescan, Modell 1471 **1598,-** –
Kabel EIZO/NEC an ATARI ST **75,-** –
Monitor-Umschaltbox **47,-** –

Bestellungen per Nachnahme oder Vorkasse ab 30 DM. Auslandslieferungen nur gegen Vorkasse. Eingetragenes Warenzeichen: ATARI ST. Die Preise können günstiger liegen. Rufen Sie an! Händleranfragen erwünscht.



Immer weiter geht es mit dem Kurs, immerhin haben wir schon das 5. Kapitel erreicht und noch immer kein Ende vorzusehen. Teil 4 hat sogenannte Datei-Befehle aufgegriffen. In dieser KICKSTART-Ausgabe möchten wir auf Befehle der CLI-Steuerung näher eingehen.

Befehle der CLI-Steuerung

Befehle der CLI-Steuerung haben immer eine bestimmte Auswirkung auf das CLI. Ändern des Prompts, oder des Stacks gehören, ebenso dazu, wie das Starten eines neuen Tasks. Aufgreifen möchten wir in Kurs 5 die Befehle 'break', 'cd', 'endcli', 'newcli', 'prompt', 'run', 'stack', 'status' und 'why'.

Fangen wir ohne große Worte gleich an.

Break

Was bewirkt der 'break' Befehl? Er setzt Unterbrechungskennungen in einem Task. Mit Hilfe von 'break' kann in einem parallel laufenden Task ein AmigaDOS Bedienungs-Interrupt aufgerufen werden. Als AmigaDOS Bedienungs-Interrupt stehen dem Benutzer gleich vier zur Verfügung, die durch die Tastenkombinationen CTRL-C, CTRL-D, CTRL-E und CTRL-F erzeugt werden können. CTRL-C unterbricht hierbei die AmigaDOS Befehle, CTRL-D Stapeldateien (mit dem Befehl 'execute'

gestartet), CTRL-E und CTRL-F werden nur selten bei speziellen Befehlen benötigt und müssen nicht weiter erläutert werden.

Der Syntax von 'break' ist wie folgt festgelegt:

break Task [all][c][d][e][f]

'Task' repräsentiert hierbei den entsprechenden Prozeß, 'all' wird gesetzt, wenn alle Interrupts erzeugt werden sollen 'c', 'd', 'e' und 'f' für den bestimmten Interrupt.

Ein Beispiel:

In einem zweiten Task wird gerade eine Stapeldatei abgearbeitet, jetzt wollen Sie diesen Ablauf vom ersten Task aus unterbrechen, dazu geben Sie einfach

break 2 d

ein und die Stapeldatei wird abgebrochen. Die gleiche Auswirkung wird erzielt, wenn Sie mit dem Mauszeiger den entsprechenden Task anwählen und CTRL-D drücken. 'break' ermöglicht es lediglich, von einem anderen Task den Ablauf zu unterbrechen.

Weiter mit dem 'cd'-Befehl

Change Directory, cd

Es soll kein Zugriff auf eine Compact Disk (CD) vorgenommen werden, sondern das Verzeichnis gewechselt werden. Zum Syntax:

cd [Verzeichnis]

'Verzeichnis' stellt hier das Unterverzeichnis, in das gesprungen werden soll, dar. Befinden Sie sich beispielsweise im Hauptdirectory der Workbench-Diskette und geben

cd c

ein, so wird das 'c'-Verzeichnis zum Aktuellen, welches dann mit dem Befehl 'dir' aufgelistet werden kann. Ebenso ist es möglich, im Anschluß den Verzeichnisbaum weiter herunter zu klettern.

cd devs/printers

bewirkt, wenn man sich im Hauptverzeichnis der Workbench-Diskette befindet, daß man sich nun im Verzeichnis 'printers' befindet. Dieser Befehlsausdruck ist aber nur vom Hauptdirectory aus möglich. Befindet man sich jedoch beispielsweise im 'c'-Verzeichnis der Workbench und möchte in das 'printers'-Verzeichnis, stehen dem Benutzer zwei Möglichkeiten, um sein Ziel zu erreichen zur Auswahl. Erstens, durch die Befehlsfolge

cd / cd devs/printers

'cd /' bewirkt jeweils ein Zurückkehren in das nächst höhere Verzeichnis der Hierarchie vom momentan Aktuellen aus. Durch Setzen von zwei oder drei Schrägstrichen (/) springt man dementsprechend höher in der Hierarchie.

In unserem Beispiel, also in der Haupt- oder Rootdirectory. Von da aus wird erst in das 'printers' Directory verzweigt. Der zweite, kürzere Weg, ergibt sich folgendermaßen:

cd df0:devs/printers

Sicherlich haben Sie bemerkt, daß eine Laufwerksbezeichnung in dem Befehl aufgetaucht ist. Mit der Angabe des gewünschten Laufwerks entfällt der 'cd /' Befehl. Der Befehlsausdruck kann in jedem Verzeichnis durchgeführt werden, egal wie tief man sich im Verzeichnis-Baum befindet, man 'findet' sich nach Eingabe immer im Verzeichnis 'printers' wieder.

cd df1:

wechselt immer auf das zweite Laufwerk.

cd ram:

in auf die RAM-Disk.

cd dh0:

oder

cd jh0:

Auf die Festplatte. Beachten Sie, daß immer der Doppelpunkt (:) hinter den Laufwerken stehen muß:

'cd' ohne eine Verzweigungsangabe gibt immer den Pfad des aktuellen Directories aus.

Neuer CLI-Task

Der CLI-Befehl 'newcli' gestattet es, einen weiteren CLI Task zu starten und öffnet dabei ein neues Fenster. Zum Syntax:

newcli [con:x/y/Breite/Höhe/[Titel]]

Mit dem Ausdruck 'con:x/y/Breite/Höhe' kann die Dimensionierung des neuen Fensters festgelegt werden. 'x' und 'y' bestimmen hierbei die neuen, oberen Fenster Koordinaten, 'Breite' und 'Höhe' die Abmessungen. Alle Angaben erfolgen in Pixelwerten.

'Titel' beschreibt einen neuen Namen des CLI-Fensters, der im Fensterbalken ausgegeben wird.

Ohne Angabe einer Dimensionierung wird der neue CLI Task in einem Standard Fenster erzeugt.

newcli con:0/10/400/100/

erzeugt ein Fenster mit der Anfangsposition 0,10 und der Breite von 400 Bildpunkten und der Höhe von 100 Bildpunkten.

newcli con:0/10/400/100/CLI2

erzeugt ein Fenster mit den selben Abmessungen wie oben beschrieben, nur mit der zusätzlichen Ausgabe eines neuen Namens, in unserem Beispiel 'CLI2'.

Man sollte noch erwähnen, daß mit dem Befehl 'newcli' aufgerufene Tasks, vollkommen unabhängig zu anderen Tasks arbeiten, so wie alle Tasks des Amigas. Also kein Zusammenhang vom Aufgerufenen besteht. Das neue CLI-Fenster kann genauso gehandhabt werden, wie das alte CLI-Fenster. Um nun von einem Fenster in das andere zu wechseln, müssen Sie mit dem Mauszeiger das gewünschte Fenster anwählen, was dadurch in aktiven Zustand gesetzt wird. Ein aktives Fenster erkennen Sie immer daran, daß der Titel deutlich zu lesen ist. Beim inaktiven Fenster ist der Titelbalken 'ghostet' dargestellt. Ein CLI Fenster kann wieder geschlossen werden, dafür steht der 'endcli' Befehl zur Verfügung. Dessen Syntax ist recht einfach:

endcli

Der gleiche Ausdruck bewirkt ein Beenden des CLI Tasks, in dem man sich gerade befindet. Vorsicht ist mit dem Befehl geboten. Existiert kein weiterer CLI Task und die Workbench ist nicht geladen (lodwb), darf der Befehl auf keinen Fall durchgeführt werden. Ansonsten steht der Benutzer alleine dar, keinerlei Eingaben können mehr gemacht werden und nur ein Ausweg bleibt offen. Das erneute Booten ist unumgänglich.

Eigene Zeilenmarke!

Eine eigene Zeilenmarke kann man sich mit dem Befehl 'prompt' erstellen. Dieser sehr einfache Befehl ohne komplexe Optionen besitzt nachstehenden Syntax:

prompt [Zeilenmarke]

'Zeilenmarke' stellt hierbei die neue Marke dar. Beim Öffnen des CLIs ist die Marke immer gleich, nämlich '1>'. '1' repräsentiert die Nummer des CLI Tasks. Bei jedem weiteren Aufrufen eines CLI Tasks mit dem 'newcli' Befehl, wird aus der '1' eine '2' usw.

prompt Marke

bewirkt eine Änderung der Zeilenmarke von '1>' zu 'Marke', dementsprechend

prompt?

Das hat eine Besonderheit. Nach dieser Eingabe wird 'PROMPT:' auf den Bildschirm ausgegeben. Der Rechner erwartet jetzt die neue Eingabe der Zeilenmarke. Geben Sie nur

prompt

ein, wird die Marke immer auf '>' gesetzt. Wollen Sie Zeichenketten mit Leerzeichen als Marke verwenden, müssen Sie die Zeichen nach dem 'prompt' in Anführungsstriche setzen. Beispiel:

prompt "KICKSTART, die Amiga Zeitschrift"

Ansonsten wird die Marke mit diesem Ausdruck vertauscht. Als Letztes möchte ich noch erwähnen, daß es möglich ist, auch die neue Marke mit einer Tasknummer zu versehen, dazu geben Sie lediglich den Ausdruck '%n' zusätzlich mit ein.

prompt %n -

ändert die Zeilenmarke des ersten CLI TASKS auf '1 - -'.

Programme im Hintergrund!

Mit Hilfe des 'run' Befehls ist es möglich, Programme als Task im Hintergrund ablaufen zu lassen. Im Gegensatz zur Workbench, wo alle gestarteten Programme als Task arbeiten, macht es das CLI nicht. Im CLI laufen alle Programme im selben Task, wenn man nicht den 'run' Befehl zu Hilfe nimmt.

Der Syntax:

run Befehl[+ Befehl +

.
.
.

Befehl]

'Befehl' steht hierbei entweder für ein Programm, oder aber für ein CLI-Kommando. Mit '+' ist es möglich, weitere Befehle anzuhängen, die ebenfalls als Hintergrund ablaufen. Die höchste Priorität besitzt immer der Grundtask, demnach Task 1. Geben Sie während der Abarbeitung einen weiteren Befehl ein, wird dieser

GRUNDLAGEN

somit ausgeführt. Nach Beendigung des Befehls oder Programms schließt sich der Text automatisch wieder. Keine weiteren Befehle sind notwendig.

Beispiel:

```
run copy Datei1 to ram: quiet +
cd ram: +
dir
```

'Datei1' wird auf die RAM-Disk kopiert, ohne eine Kopierangabe (qui-

et), danach wird das Verzeichnis gewechselt und aufgelistet. Das ganze läuft als Hintergrundtask ab, und der Benutzer kann sich anderen Aufgaben widmen.

Das war's

Mit dem 'run' Befehl möchte ich den CLI-Kurs Teil 5 beenden. Ich hoffe, Sie haben neue Einblicke gewonnen. In der nächsten KICKSTART-Ausgabe möchte ich Befehle

aufgreifen, die nicht im AmigaDOS Handbuch aufgegriffen werden. Es handelt sich dabei um Befehle, die mit der Workbench Diskette Version 1.2 in Umlauf gekommen sind. Ein sicherlich hochinteressanter Teil, da wenig Informationen über diese Befehle im Umlauf sind.

Bis zur nächsten KICKSTART-Ausgabe.

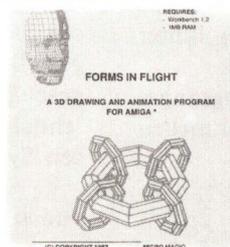
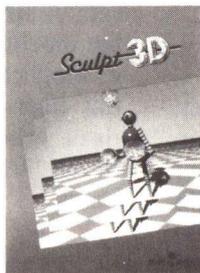
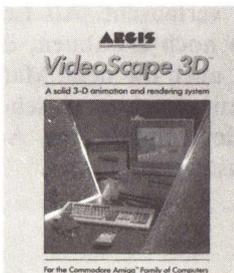
(AK)

NUR VOM FEINSTEINEN:

Videoscape 384,-
Das Nonplusultra an Animationsprogrammen.
Demo-Disk 15,-

Sculpt 3D 194,-
Erstellen Sie 3D-Ray-Tracing-Bilder von bestechender Brillanz.

Arazoks Tomb 89,-
Das Horror Adventure von Aegis. Nur für Erwachsene.



Forms i. Flight 165,-
Setzen Sie die rot-grün-Brille auf und lassen Sie Stereo-Grafiken aus dem Bildschirm fliegen.

Disney 3D Anim. 595,-
Der fantastische Trickfilm-Generator mit HAM-Modus!
Demo-Disk 20,-



Poststr. 25, 6200 Wiesbaden
☎ 061 21/56 0084



Basaltstr. 58
6000 Frankfurt
☎ 069/7 07 11 02

Vesalia Soft & Hard Entwicklung · Produktion

Amiga und Amiga-Zubehör

Amiga 500	1098,-
Farbmonitor Amiga 1081 mit Monitorkabel	729,-
Farbmonitor Philips CM 8833 mit Stereoton u. Kopfhörerbuchse, RGB analog, > 12 MHz Bandbreite	729,-
Monitorverbindung (RGB analog)	27,-
Druckerverbindung Amiga 500 u. 2000	27,-
Amiga 1000	27,-
Midi INTERFACE 500/2000	98,-
Midi INTERFACE 1000	98,-
Sound Sampler	198,-
Sound Scope Sampler mit Software	249,-
Digi-View Digitizer mit Software	379,-
EASLY Zeichentablett für A 1000 arbeitet z. B. mit Page-Setter, Deluxe Paint II, AEGIS-Images	839,-

AMIGA-Laufwerke

VLW-1 3,5" Laufwerk (TEAC FD 35 FN) Amiga-farbenes Metallgeh. abschaltbar	299,-
VLW-2 3,5" Laufwerk (TEAC FD 35 FN) Amiga-farbenes Metallgehäuse, abschaltbar mit durchgeführtem Floppybus	319,-
VLW-3 3,5" Slimline-Laufwerk (NEC 1036 A) Amiga-farbenes Metallgehäuse, abschaltbar farblich passende Blende	329,-
VLW-4 3,5" Slimline-Laufwerk (NEC 1036 A) abschaltbar, mit durchgeführtem Floppybus, Amiga-farbenes Metallgehäuse, farblich passende Blende	349,-
VLW-5 3,5" Doppel-Slimline (2 x 1036 A) einzelnd abschaltbar, Amiga-farbenes Metallgehäuse, farblich passende Blenden	628,-
VLW-6 Mit durchgeführtem Floppybus	648,-
VLW-7 3,5" Slimline-Laufwerk (NEC 1036 A) helle Blende, mit Zubehör und Einbauanleitung, bereits modifiziert als internes Laufwerk für Amiga 2000	278,-
VLW-8 5,25" Laufwerk (TEAC FD 55 FR) Amiga-farb. Metallgehäuse, abschaltbar, farblich passende Blende, durchgeführter Bus	449,-

Alle Laufwerke mit **12 Monaten Garantie!**
Keine Wartezeiten - tägliche Auslieferung

Amiga Speichererweiterungen

512 KB für Amiga 500 mit Uhr	249,-
Gleiche Karte abschaltbar	259,-
2 MB Golem RAM-Box für Amiga 1000. autokonfigurierend, Amiga-farbenes Metallgehäuse, durchgeführter Systembus, Ein-/Aus-Schalter, erweiterbar	948,-

Software Auszug

BeckerText	199,-
Butcher	72,-
BootBoy	68,-
Bootblock Generator mit Sound	49,-
Calligrapher Schriftgestaltung	179,-
Datamat	99,-
Deluxe Music 1.2	185,-
Deluxe Paint II	239,-
Deluxe Video 1.2	185,-
Digi Paint deutsche Version	117,-

Diga Kommunikationsprg.	125,-
Grabbit	72,-
Flow Idea Processor	169,-
Logistix Tabellenkalk. u. Grafik	255,-
Master CAD 3D Cad Programm	219,-
Publisher 1000	334,-
Prism Zeichenprg. 4096 Farben	149,-
Page-Setter (Umlaute)	238,-
Superbase	227,-
Sonix	129,-
Sculpt 3D zum erst. von 3D Bildern	183,-
Textomat	99,-
Videoscape 3D	322,-
Turbocopy V 2.1 neu für alle Laufwerke	49,-
White Lightning neu für ein Laufwerk	49,-
Word Perfect 4 Disketten	739,-

Händleranfragen erwünscht

Vesalia-Versand

G. Does · Marienweg 40 · 4230 Wesel
Tel. 02 81/6 54 66 u. 6 22 05

Ladenverkauf:
Kornmarkt 23 · Rathauspassage

COMPUTER SERVICE GbR

Michael & Joachim Maier · Abt. MA · Postfach 13 04 · 7913 Senden · Tel. 0 73 07-62 30

TERRORPODS (D) 65,-	GARRISON (D) 65,-	UNINVITED (D) 69,-	FLIGHT SIMULATOR II (D) 119,-
ROADWAR 2000 (D) 65,-	PORTAL (D) 79,-	CHESSMASTER 2000 (D) 85,-	GUILD OF THIVES (D) 65,-
KNIGHT ORC (D) 59,-	SINDBAD AND THE THRONE OF FALCON (D) 79,-	PORTAL (D) 79,-	BUREAUCRACY (D) 89,-
ARAZOKS TOMB (D) 89,-	S.D.I. (D) 65,-	DEFENDER OF THE CROWN (D) 69,-	CRUNCHER FACTORY (D) 25,-
DEJA VU (D) 79,-	TASS TIMES IN TONE TOWN (D) 69,-	LEATHER GODDES OF PHOBOS (D) 69,-	GOLDRUNNER (D) 65,-
DEMOLITION (D) 25,-	THE PAWN (D) 65,-	SHANGHAI (D) 69,-	KARATE KID II (D) 65,-
FLIP FLOP (D) 25,-	HARDBALL (D) 65,-	STARGLIDER (D) 65,-	Katalog mit Programmbeschreibungen gegen DM 0,80 in Briefmarken
JEWELS OF DARKNESS (D) 55,-	BARBARIAN (D) 65,-	SUPER HUEY (D) 59,-	

Versandkosten: bis 150,- DM Vorauskasse 3,50 DM Nachnahme 6,- DM
ab 150,- DM keine Versandkosten

Hardware und Zubehör bitte gesondert anfragen.
Telefonische Bestellungen sind auch abends und an den Wochenenden möglich (Anrufbeantworter)



ABO



ABO

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte
mit
60 Pf.
frankieren

Heim-Verlag

Heidelberger Landstr. 194

6100 Darmstadt-Eberstadt

Telefon 0 61 51 / 5 60 57



Einzelheft- u. Monatsdisketten Bestellung



Einzelheft- u. Disketten Service

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte
mit
60 Pf.
frankieren

Heim-Verlag

Heidelberger Landstr. 194

6100 Darmstadt-Eberstadt

Telefon 0 61 51 / 5 60 57



Kleinanzeigen



Kleinanzeigen

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte
mit
60 Pf.
frankieren

Heim-Verlag

Heidelberger Landstr. 194

6100 Darmstadt-Eberstadt

Telefon 0 61 51 / 5 60 57

Ja, bitte senden Sie mir die Amiga-Computer Fachzeitschrift ab _____
für mindestens 1 Jahr (11 Hefte) zum ermäßigten Preis von jährlich DM 70,- frei Haus.
(Ausland: Nur gegen Scheck-Voreinsendung DM 90,- Normalpost.)
Der Bezugszeitraum verlängert sich nur dann um ein Jahr, wenn nicht 6 Wochen vor Ablauf des Abonnements gekündigt wird.

Name _____

Vorname _____

Straße/Nr. _____

PLZ _____ Ort _____

Datum _____ Unterschrift _____

Gewünschte Zahlungsweise bitte ankreuzen

Bequem und bargeldlos durch Bankeinzug

Konto-Nr. _____ BLZ _____

Institut _____ Ort _____

Ein Verrechnungsscheck über DM _____ liegt bei.

Vorkasse per Post-Einzahlung (Zahlkarte)

Garantie:
Diese Bestellung kann ich schriftlich innerhalb einer Woche (rechtzeitige Absendung genügt) widerrufen. Dies bestätige ich durch meine 2. Unterschrift.

Datum _____ Unterschrift _____

KICKSTART können Sie direkt beim HEIM-VERLAG zum Einzelheft-Preis von DM 7,- (zuzüglich Gebühr für Porto und Verpackung) nachbestellen. Bearbeitung nur gegen beigefügten Scheck über den entsprechenden Betrag (keine Überweisung).

Jan.	Febr.	März	April	Mai	Juni	Juli/Aug.	Sept.	Okt.	Nov.	Dez.

1987 = DM

1988 = DM

+ Gebühr für Porto u. Verp. _____ = DM _____

Scheck in Höhe _____ zus. DM _____ liegt bei

Disketten Service

Alle Programme, die in KICKSTART veröffentlicht wurden, sind auf Disketten erhältlich. Die Disketten enthalten die Programme von jeweils 2 KICKSTART-Ausgaben. Bestellen Sie durch ankreuzen die gewünschten Disketten

Preis je Diskette 19,- DM	Juli/Aug.	Sept./Okt.	Nov./Dez.	Jan./Febr.	März/Apr.
	87	87	87	88	88

Lieferung: gegen beigefügten Scheck zuzügl. 5,- DM Versandkosten

Bitte veröffentlichen Sie für mich folgende Kleinanzeige in der angekreuzten Rubrik

Biete an Hardware Software Ich suche Hardware Software Tausch Kontakte Verschiedenes

30 Buchstaben je Standardzeile - incl. Satzzeichen und Wortzwischenräume.
Groß- und Kleinbuchstaben verwenden, fettgedruckte Wörter unterstreichen.

Bearbeitung nur gegen Vorausscheck über den entsprechenden Betrag (keine Überweisung)

privat = DM 7,- je Zeile incl. MwSt. gewerblich = DM 15,- je Zeile + MwSt. Chiffregebühr = DM 10,- Scheck über DM _____ ist beigefügt

Bei Angeboten: Ich bestätige, daß ich alle Rechte an den angebotenen Sachen besitze.

Datum _____ Unterschrift _____

Absenderangaben auf der Rückseite nicht vergessen



Kontaktkarte



Kontaktkarte

Bitte Adresse der Firma, bei der Sie Informationen, oder etwas bestellen möchten, auf der rechten Seite eintragen ----->

Absender
(Bitte deutlich schreiben)

Vorname/Name

Beruf

Straße/Nr.

PLZ/Ort

Telefon Vorwahl/Rufnummer

Postkarte

Bitte freimachen

Firma

Straße/Postfach

PLZ Ort



Meinungskarte



Meinungskarte

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte freimachen

Merlin Computer GmbH
KICKSTART Redaktion
Industriestraße 26
6236 Eschborn



PD Bestellung



PD Bestellung

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte freimachen

Merlin Computer GmbH
KICKSTART Redaktion
Industriestraße 26
6236 Eschborn



- Ich bitte um weitere Informationen
 - Ich gebe folgende Bestellung auf
- in Bezug auf Ihre Anzeige in Kickstart Heft _____ Seite _____

Menge	Produkt/Bestellnummer	DM	gesamt DM

Datum, Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)



Abgesandt am:



Firma: _____

Bemerkungen:

An  finde ich toll/nicht so toll:



Meinungskarte



PUBLIC DOMAIN SERVICE

Ich bestelle folgende PD-Disketten:
(Siehe PD Service in dieser Ausgabe)

Zahlung erfolgt:

- per Scheck
- per Nachnahme

Je Diskette fügen Sie bitte einen Betrag von DM 10,- bei,
für Porto und Verpackung je Sendung DM 5,- (Ausland DM 10,-)

Datum

Unterschrift



PD Bestellung

Datum

TERRORPODS

Nach dem erfolgreichen Barbarian veröffentlicht Psygnosis jetzt Terrorpods. Ein erwarteter Abglatsch von Barbarian ist es nicht geworden, sondern eine völlig andere Idee liegt Terrorpods zugrunde. Das Geschehen des Aktion-Strategie Games spielt sich auf dem fernen Planeten Colian ab, große Temperaturunterschiede machen das Wohnen dort ungemütlich, aber wegen des enormen Reichtums an Bodenschätzen ist der Planet von unschätzbarem Wert. Dementsprechend sind die Industrieanlagen zahlreich, neben Minen existieren Aufbereitungsanlagen, Warenlager, Pumpstationen, und etliches mehr. Die Aufgabe des Spielers besteht nun darin, die feindlich gesinnten Terrorpods zu eliminieren, welche leider die Industrieanlagen zerstören, doch das ist leichter gesagt als getan, die Feuerkraft des Fahrzeugs, mit dem Sie über den Planeten streifen, ist begrenzt, ebenfalls die Brennelemente, die das Gefährt vorwärts bewegen. Außerdem genügt eine Salve nicht die Terrorpods zu vernichten, man vertreibt sie nur. Eine stärkere Waffe muß dazu herangezogen werden, doch die benötigt viel Energie. Während man sich die unliebsamen Terrorpods vom Leibe hält, müssen immer wieder Warenlager, Fabriken und Tankstellen angefahren werden, um seinen Energiehaushalt aufzufrischen. Dazu dient übrigens ein kleines Roboterfahrzeug, das vom Mutterschiff ausgesetzt wird, um die verschiedenen Stützpunkte anzulaufen. Nicht nur die Terrorpods werden einem gefährlich, sondern auch ab und zu auftauchende Torpedos, denen man aber wegen Energieersparnis ausweichen sollte, was übrigens sehr leicht geht.

Terrorpods besticht durch die Vielzahl von Möglichkeiten, eine Karte kann zur besseren Orientierung herangezogen oder zerstörte Anlagen wieder aufgebaut werden. Doch damit nicht genug, neben dem komplexen Spiel stimmt auch das drumherum. Das Spiel kann abgespeichert und zu einem späteren Zeitpunkt fortgesetzt



Nur mit einer speziellen Zielvorrichtung können die Terrorpods endgültig vernichtet werden, leider geht die Energie für diese Aktion des öfteren aus.

werden, ebenso kann der Schwierigkeitsgrad zwischen sechs Stufen gewählt werden. Der Highscore wird natürlich auch abgespeichert. Zu Beginn des Spiels kann man unter verschiedenen Nationalitäten wählen, dementsprechend sind auch die Kommentare während des Spiels in der jeweiligen Landessprache. Die deutschen Kommentare sind zwar zum Teil etwas holprig, was aber nicht sonderlich stört. Das Scrolling ist gut, bedenkt man, daß der Hintergrund langsamer scrollt als der Vordergrund.

Lediglich der Sound ist etwas zu

kurz gekommen, was aber bei diesem Spiel nicht sonderlich ins Gewicht fällt. Zum Lieferumfang gehört neben der Diskette, ein Anleitungsbogen und ein Poster, welche die Terrorpods in Aktion zeigt.

Fazit:

Wer nicht nur ballern will, sondern auch Strategieelemente liebt, dem kann zu diesem Spiel nur angeraten werden. Die ausgezeichnete Grafik sowie die Spielmotivation sprechen nur für das Spiel. Psygnosis hat mit Terrorpods wieder einen Hit gelandet der sicher seinen Weg in der Amigawelt machen wird.



DEFCON-3, jetzt wird es schon langsam gefährlich für den Terrorpods Jäger.

IDEE:

SOUND:

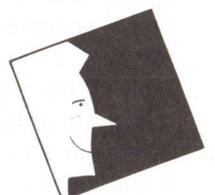
GRAFIK:



Vertrieb:

DTM, Tel. 0 61 21 / 56 00 84
IM, Tel. 0 69 / 707 11 02

Preis: 79,- DM



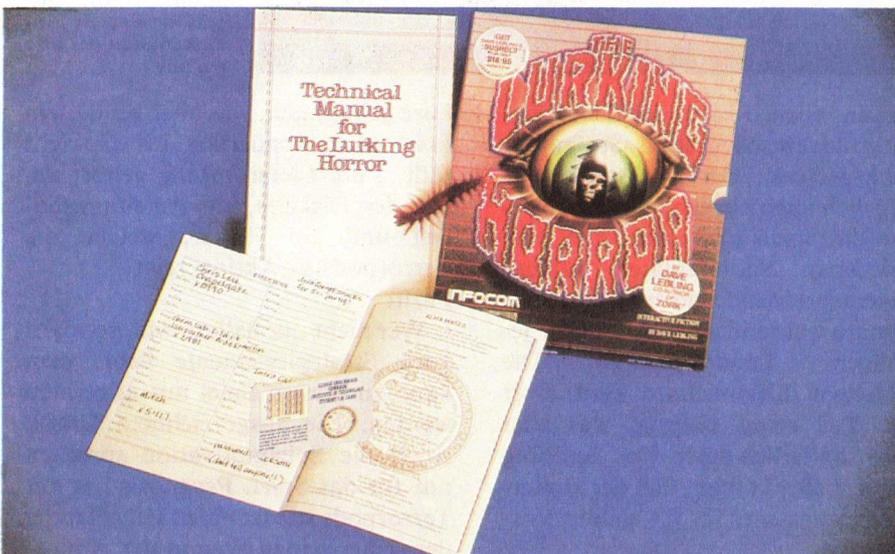
ADVENTURES

The Lurking Horror und Stationfall im Test

Nach Douglas Adams geradezu genialen 'Bureaucracy' sind 'The Lurking Horror' und 'Stationfall' die nächsten, wirklich sehr guten und

anspruchsvollen Text-Adventures aus dem Hause Infocom. Dieser Name steht schon seit der legendären Zork-Trilogie für Text-Adventure an Spit-

zenklasse. Auch mit den beiden Neuerscheinungen zeigt sich, daß dieser Ruf gerechtfertigt ist.



Mit 'The Lurking Horror' hat Infocom endlich dem Wunsch vieler Fans nachgegeben und ein Horror-Adventure geschaffen. Ort der Handlung sind die Gebäude der fiktiven G.U.E. Universität. Draußen wütet ein entsetzlicher Schneesturm und der Spieler ist mehr oder weniger eingesperrt, was nicht von Nachteil ist, da man noch 20 Seiten für eine wichtige Klausur zu schreiben hat. Kaum hat man sich jedoch an seinen Terminal gesetzt und hat, nach Eingabe des Codewortes, das nur im Handbuch steht (Der beste Kopierschutz für die nicht geschützten Programme von Infocom sind immer noch die Handbücher.), angefangen den bisher getippten Text zu lesen, geht der Horror auch schon los.

Auf dem Bildschirm finden sich Daten aus einem Alchemisten Labor in der Uni ein. Da geht es um Geister-

beschwörung und schwarze Magie. Geheimnisvolle Kräfte zwingen den Studenten den Text zu lesen, bis er schließlich in einen erschreckend realistischen Alptraum fällt. Nach dem er aufgewacht ist, sind alle Daten gelöscht und er muß sich erneut auf die Suche danach machen. Von einem Hacker bekommt er einen wertvollen Tip. Der Spieler muß jetzt in die Keller der Uni vorstoßen, in die er sich bisher noch nie gewagt hat, da sich allerhand merkwürdige Geschichten darüber erzählt werden. Aber nicht nur im Keller lernt man das Gruseln. Ein Mann der eine Bodenwachsmaschine in einem der Korridore fährt, stellt sich bei genauerer Untersuchung als Zombie heraus. Aber auch hier setzt sich das relativ gewaltlose Prinzip von Infocom durch, es hat wenig Sinn zu versuchen, den Zombie mit roher Gewalt umzubringen.

Wenn man das Spiel länger spielt, breitet sich eine gruselige Atmosphäre aus, die bei dem Grusel-Adventure 'Moonmist' nicht so recht aufkommen wollte. Leute mit schwachen Nerven sollten 'The Lurking Horror' also nicht zu später Stunde spielen. Der Autor Dave Lebling hat ein Programm geschaffen, das einen Ehrenplatz in seiner bisherigen Erfolgsliste einnehmen kann. Er schuf Hits wie 'Spellbreaker', 'Starcross' und 'Suspect'. An der Zork-Trilogie war er als Co-Autor beteiligt.

Die üblichen Nachteile bei Spielen von Infocom, Preis und Sprache sind auch hier wieder gegeben. Der Preis ist vielleicht durch das neue Design der Verpackung und die Zugaben in der Packung (außer Anleitung und Diskette) zu entschuldigen. Die Sprache, recht anspruchsvolles Englisch, gehört ebenfalls dazu und ist schon fast ein Markenzeichen. Aber mit durchschnittlichen Englischkenntnissen und einem dicken Wörterbuch läßt sich 'The Lurking Horror' wie auch alle anderen Infocom-Adventure recht flüssig spielen. Außerdem, und das sei an die Adresse der Schüler gerichtet, lernt man, wenn man mehrere Infocom-Adventures spielt, geradezu spielend Englisch.

Möchte man etwas Anspruchsvolles auf den Bildschirm bekommen, dem sei 'The Lurking Horror' wie auch jedes andere Infocom-Adventure empfohlen.

Zum Abschluß noch ein paar Tips für die Anfänger:

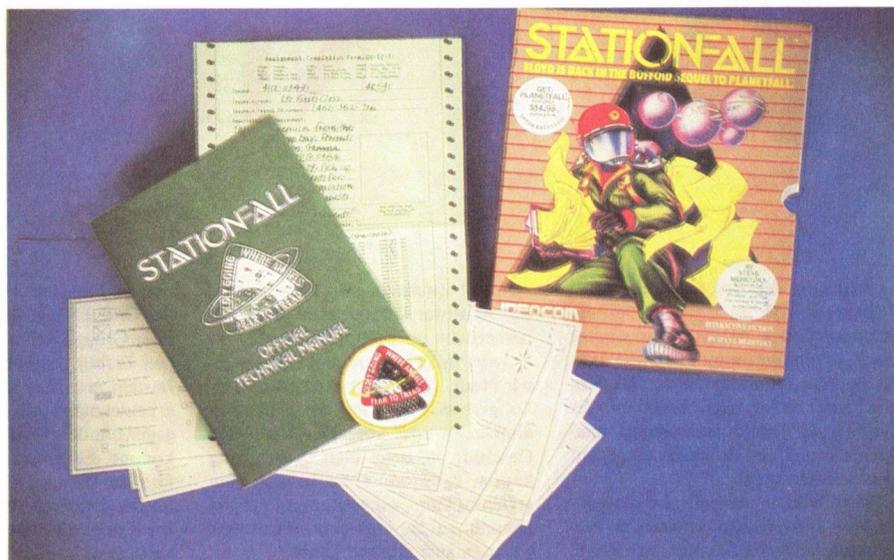
- Der Hacker rückt den entscheidenden Schlüssel nur raus, wenn er etwas vernünftiges zu Essen bekommt.
- Der Zombie hat Schwierigkeiten auf gewachstem Boden zu laufen.
- Cola ist koffein-haltig. Und das hat aufmunternde Wirkung.
- Das Monster, welches die Hand begiert, hat Angst vor dem Stein.
- Die Ratten 'beruhigt' man, indem man das Ventil am Rohr im richtigen Augenblick aufschlägt.
- Den Abschiedsbrief des Studenten zeigt man dem Professor.

Auch mit 'Stationfall' hat Infocom bzw. Steve E. Meretzky einen langersehnten Wunsch der großen Fangemeinde von 'Planetfall' erfüllt. Vor vier Jahren brachte Infocom das Science Fiction Adventure 'Planetfall' heraus. Nicht nur wegen des sehr guten Begleitcharacters, der je in einem Adventure von Infocom vorkam, nämlich dem genial kindischen, verspielten und urkomischen Roboter Floyd, forderten die Fans schon bald einen Nachfolger. Die lange Wartezeit hat jetzt ein Ende.

Nachdem sie den Planeten Resida in 'Planetfall' vor dem Untergang gerettet haben, sind sie bis zum Leutnant erster Klasse befördert worden. Das Leben sollte jetzt aufregender werden, als es bisher als Raum(schiff)pfleger war. Aber sie haben nur eine dieser Arbeiten gegen eine andere eingetauscht.

Jetzt müssen sie Schreibtischarbeit verrichten, bei der es das Aufregendste ist, wenn sie einen Ausflug zur Raumstation Gamma Delta Gamma 777-G 59/59 machen dürfen, um sich um Formulare zu kümmern.

Und das ist genau das, was sie heute tun müssen. Für alles, was sie hierzu brauchen, werden natürlich, wie könnte es anders sein, Formulare benötigt, die auch in der Packung enthalten sind.



Sind sie nach dem freudigen Wiedersehen mit Floyd endlich bei der Raumstation Gamma Delta Gamma 777-G 59/59 angekommen, wird es schon merkwürdig. Kein Empfangskomitee, nichts.

Die gesammte Station ist verlassen. Sie bekommen Hinweise auf eine Pyramide, die die ganze Menschheit bedroht. Ihre Aufgabe ist es nun, das Universum zu retten.

Im Laufe der Zeit zeigen sich die drastischen Auswirkungen der Pyramide. Zum Beispiel fängt Floyd an, sich sonderbar zu benehmen (unter anderem rülps er und entschuldigt sich nicht dafür). Oder es funktionieren Maschinen nicht mehr wie erwartet und verschwinden sogar. Am Ende des Spiels, so traurig es ist, muß die Fangemeinde für immer von Floyd abschiednehmen. Man bekommt einen Vertreter namens Oliver.

Für einen Teil des Spiels entfällt das Kartenzichnen. Ein Plan der Raumstation liegt bei. Die angedockte Raumstadt darf der eifrige Abenteurer aber wie immer mit Papier und Bleistift erkunden.

Stationfall ist gagreicher als 'The Lurking Horror'. Schon allein die Wiedereinführung der Fußnoten,

'Hitchhiker'-Fans werden sich erinnern, und der Kommentar des Autors dazu ist sehenswert. Ebenso wird die Bürokratie auf die Schippe genommen.

Die meisten Probleme sind leicht zu lösen. Ein paar harte Nüsse sind aber auch noch da, die ja das Salz in der Suppe ausmachen.

'Stationfall' ist ein unbedingtes Muß für alle begeisterten 'Planetfall' Spieler. 'Stationfall' ist sicher eines der besten, wenn nicht das beste Programm, daß Infocom je herausgebracht hat.

Hier noch ein paar Tips für Anfänger:

- Das zerknitterte Formular muß man pressen.
- Den Strauss lockt man mit Futter zu dem streikenden Automaten und erschrickt ihn.
- Der Aufbewahrungsort des Futters ist in einem Brief beschrieben.
- Die seltsame Kreatur lockt man mit Hilfe der Spraydose in die Kapelle, wo man mit dessen Hilfe an den Stern herankommt.
- Die Thermosflasche isoliert auch gegen Wärme.

(Andreas Paul)

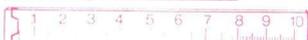
Vertrieb: INFOCOM

Preis: je 89,-

IDEE:

SOUND:

GRAFIK:





THE WRATH OF NIKADEMUS

Schon in der Beschreibung steht: 'Phantasia III zu spielen, sei ein schweres Stück Arbeit'. Das stimmt auch, es gibt bestimmt hundert Punkte die man beachten sollte, will man das Spiel erfolgreich bestehen. Es wird wohl jedem einleuchten, daß hier nur auf ein Bruchteil eingegangen werden kann.

Lord Nikademus will also die ganze Welt beherrschen. Um das zu verhindern kommen Sie nun zum Zuge. Zuerst muß eine Gruppe von sechs 'Abenteurern' zusammengestellt werden. Da gibt es eine ganze Reihe von Kreaturen, mit unterschiedlichen Fähigkeiten, unter denen man wählen kann. Da sind einmal die Menschen, die in allen Sparten ganz gut sind. Dann gibt es noch Zwerge die kleiner und langsamer wie die Menschen, jedoch zäher sind.

Elven die geschmeidig und intelligent, aber schwach sind. Gnome die nicht ganz so stark sind wie Zwerge, aber auch nicht so plumb. Halflinger, die zwar klein sind, aber dafür schnell, zäh und intelligent. Zu diesen Figuren gibt es nun die 'Zufälligen Kreaturen', d. h. wenn man diese Sparte anklickt bekommt man eine Kreatur zugewiesen. Unter den 'Zufälligen Kreaturen' sind Gnolls (stark, zäh, häßlich, dumm), Goblins (ähnlich Gnome), Kobolde (schnell, zäh, klein, dumm), Eidechsen-Mensch (großer Schwanz, große Klauen), Minotaurs (stark, zäh, langsam, dumm), Ogres (groß, häßlich), Orcs (intelligent, stark), Pixies (schnell, intelligent, schwach), Sprites (schnell), Trolls (groß, stark, zäh, dumm, ungeschickt, häßlich).

Jetzt kann sich jeder Abenteurer in einer von sechs Klassen spezialisieren, nämlich: Kämpfer, Mönch, Priester, Rancher, Dieb oder Zauberer. Auch hier hat jede Klasse seine Eigenschaft. Einen oder mehrere Kämpfer braucht man sowieso in der Gruppe, Mönche können gut Handel treiben, Priester sind durchschnittliche Kämpfer, haben aber einige heilende Zaubersprüche auf Lager, Rancher sind gut in Kämpfen und wissen auch einige heilende Zaubersprüche, Diebe sind gewandt und gute Bogenschüt-

zen und Zauberer sind zwar schwache Kämpfer können aber ihre Zaubersprüche im Kampf einsetzen.

Jedes Spiel beginnt und endet in einer Stadt. Jede Stadt sieht gleich aus. Da gibt es eine Bank wo man Geld abnehmen oder auszahlen kann, ein Geschäft in dem man Ausrüstung kaufen kann, ein Haus in dem man erfahren kann wie weit man im Spiel gekommen ist und eine Wirtschaft wo man sich stärken kann.

Nun geht es mit der Gruppe aus der Stadt. Schon nach den ersten Metern kann eine Begegnung mit feindlichen Monstern stattfinden. Sie kön-

nen die Monster erst mal grüßen. Hat der Groß Erfolg, dann können Sie mit ihrer Gruppe weiter gehen. Hat er keinen Erfolg, kommt es zum Kampf. Besteht man den Kampf erfolgreich, erbeutet man Gold oder Ausrüstungsgegenstände. Verliert man dagegen, kommt die ganze Gruppe vor den Großen Dämon. Dieser entscheidet, ob der jeweilige Abenteurer zerstört wird, oder als Untoter weiter mitwirken kann.

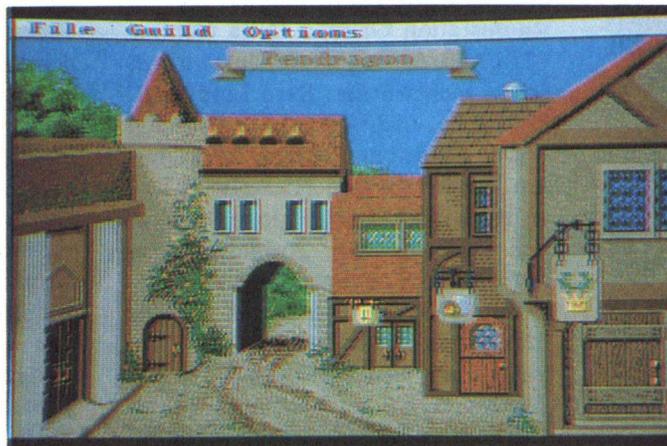
Nun gibt es noch die Verließe, in denen es auch so manche Überraschung gibt, die ich aber nicht verraten will.

Ist man von einem erfolgreichen Ausflug in eine Stadt zurückgekehrt, sollte man sein erbeutetes Gold am Besten anlegen, zum Beispiel in Ausrüstungsgegenständen, oder auf die Bank bringen.

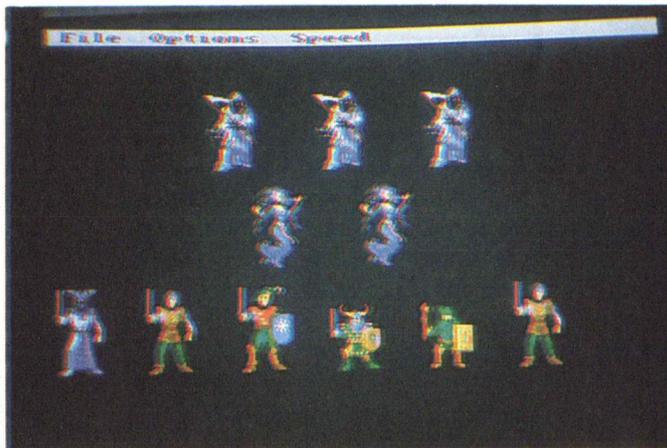
Verletzte Mitglieder der Gruppe müssen versorgt, oder ersetzt werden, um einen neuen Ausflug zu starten.

The Wrath of Nikademus ist ein spannendes Abenteuerspiel. Beachtung verdient auch die Vielfalt der Figuren, die gute Grafik und die vielen Spielvarianten.

(Andreas Suchy)



In der Stadt wird das Team zusammengestellt und dann beginnt die Reise ins Ungewisse.

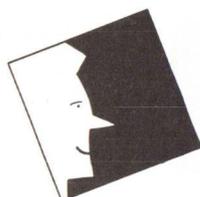


Auch wenn die Helden in der Mehrzahl sind, haben sie nur gut ausgerüstet eine Chance gegen die Mächte der Finsternis.

IDEE:

SOUND:

GRAFIK:



Vertrieb: IM, DTM

Preis: 89,-



AUTOREN GESUCHT

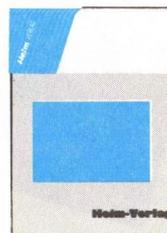
Sie

- ... haben eine gute Programmidee
- ... wollen ein Buch schreiben
- ... kennen eine Menge Tips u. Tricks
- ... möchten Ihre Erfahrungen weitergeben

Wir

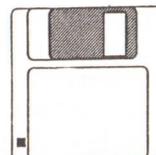
- ... bieten Ihnen unsere Erfahrung
- ... unterstützen Ihre Ideen
- ... sind ein leistungsstarker Verlag
- ... freuen uns von Ihnen zu hören

Buch



+

Programm



Schreiben Sie uns

Heim-Verlag
Kennwort: Autor
Heidelberger Landstr. 194
6100 Da.-Eberstadt
Tel.: 06151/56057

Kaufen — Verkaufen Hardware — Software

Nutzen Sie unseren privaten Kleinanzeigenmarkt.
Eine private Anzeige ist preisgünstig und bringt
den gewünschten Erfolg.

Beispiel: 5 Zeilen à 30 Zeichen kosten nur 35,- DM

Heim-Verlag

Heidelberger Landstr. 194
6100 Darmstadt-Eberstadt
Telefon 06151/56057

Kleinanzeigen

SIDECAR 512 KB—09721/58070

Wer beteiligt sich an Etiketten-
sammelbest. für AMIGA DISK?
TOM — Pf. 1162, 4516 Bissendorf

Public Domain Kopierservice ca.
250 aus USA u. BRD ab DM 4,—,
Infodisk DM 5,— wird verrechnet
v. Trempelmann, Lochnerstr. 24,
4030 Ratingen, Tel. 02102/23371

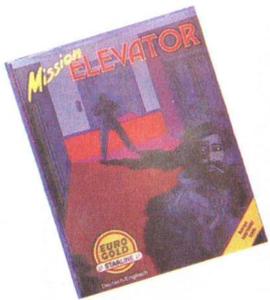
Public Domain Kopierservice Alle
Disk dieser Ausgabe ★ 5.50 DM
Stück und weitere 200 Disk ★ 30
Disk 150 DM ★ Tel.: 02954/1050

★ AMIGA SOFTWARE ★

Write to Rainer Dietmer, Am
sonnigen Hang 4, 6500 Mainz 32

Suchen, tauschen, (ver)kaufen
Amiga Public Domain Info: 10 DM
Beschr. der Prog. über 350 Screens
= wird ersetzt durch 1 Disk extra.
Private usergroup Ait — Ziegelei-
weg 32, 3257 Springe 4

Amiga 500 Besitzer sucht
Tauschpartner im Raum 6000 Ffm.
Tel.: 069/701805



MISSION ELEVATOR

Mission Elevator ist von anderen Rechnern auf den Amiga übertragen worden. Der Grund hierfür lag an dem großem Erfolg des Programms auf den anderen Systemen.

Zur Story: Agent Trevor hat die Aufgabe, in einem von feindlichen Spionen besetzten Hotels, eine Bombe zu entschärfen. Leider besitzt das Hotel 62 Stockwerke, in denen geheime Codes versteckt sind, die wiederum zum entschärfen der Bombe benötigt werden. Die einzelnen Stockwerke sind mit einer nicht geringen Zahl von Aufzügen verbunden. Trevor muß sich nun Stockwerk für Stockwerk nach oben arbeiten, wobei eine übermächtige Anzahl von feindlichen Spionen überwältigt werden muß. Doch damit nicht genug, bestimmte Schlüssel gibt nur der Portier heraus, der zunächst aufgefunden werden muß.

Soweit zur Rahmenhandlung. Mission Elevator ist ein durchaus gelungenes Spiel für den Amiga, die Sprites sind gut animiert und langweilig wird es dem Spieler nie. Viel Liebe im Detail haben die Entwickler dem Programm mitgegeben. So können beispielsweise alle auf den Gängen stehenden Gegenstände untersucht werden, immer neue Überraschungen treten dabei auf. Auch die Pausenfunktion wurde eigenwillig verwirklicht, Trevor muß hierbei ein auf den Hotelgängen stehenden Stuhl ansteuern und darin Platz nehmen. Das Untersuchen der Hotelräume hat seinen besonderen Reiz, denn hier wird mit digitaler Tonausgabe gearbeitet. Kleine Mädchen sagen ‚Guten Abend‘, oder Punker fahren einen böse an, auch Spione lauern hinter verschlossenen Türen. Das Scrolling des Bildschirms ist sauber, kein ‚ruckeln‘ wird bemerkt. Auch die Grafik von Mission Elevator ist gut, nur die Animation der Spielfigur Trevor, die sich nur mit dem Joystick bewegen läßt, könnte etwas besser gestaltet sein. Die Tastatur wird nur zu Beginn des Spiels zur Namenseingabe benötigt.

Ein Nachteil des Programms besteht darin, daß nach erfolgloser Entschärfung der Bombe immer wieder der doch langwierige Vorspann abgespult wird. Der Spieleifer wird etwas gehemmt. Ebenso ist noch zu bemängeln, daß das Programm sich mit keiner Speichererweiterung verträgt. Das Abschalten des eventuell vorhandenen zusätzlichen Speichers (falls möglich) ist unumgänglich, an-

sonsten muß das ‚NoFastMem‘ Programm in Aktion treten.

Den Amiga Besitzern, die schwierige und aktionreiche Spiele lieben, ist mit Mission Elevator sicherlich gedient. Spaß macht es in jedem Fall, sich Stockwerk um Stockwerk nach oben zu kämpfen. Monoton wird es dabei nicht.

(AK)



Eine Dame bietet sich an!



Verbirgt die Standuhr ein Geheimnis?

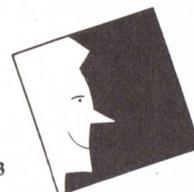
IDEE:

SOUND:

GRAFIK:



Anbieter: PDC, Tel. 06172/53863
Preis: 59.95 DM



THE LAND OF THE LOUNGE LIZARDS

Ist ein sehr eigenwilliges Spiel aus dem Hause SIERRA, das von ECA vertrieben wird. Es handelt sich hierbei um ein Action-Adventure, das durch sein brisantes Thema besticht.

Nichts für kleine Jungs!

Der Spieler übernimmt eine Spielfigur des Namens Larry Laffer, der es nur auf drei Dinge abgesehen hat: Glücksspiel, Alkohol und Girls. Doch jeder Jüngling trachtet einmal nach dem trauten Heim, nicht anders Larry. Drum liegt das Ziel unseres Helden darin, seine Traumfrau zu finden, diese zu ehelichen und in das Land von LOUGE LIZARD zu entfliehen. Doch diese Aufgabe erweist sich in Lost Wages als sehr schwierig. Mit 100 Dollar ausgerüstet muß Larry ausziehen. Zunächst jedoch geht er in eine Bar, um sich mit einigen Drinks zu stärken. Er versucht mit zwielichtigen Damen anzubendeln, hat auch immer sein Mundwasser zur Hand.

Es herrscht eine richtige Saloon-Stimmung, aufgelockert durch Geschehnisse, wie einem Toilettenbesuch u.a.

Larry Laffer ist allerdings nicht befriedigt und verläßt diesen Ort des Geschehens.

Während man auf der Straße auf ein Taxi wartet, wird man von einem Hund mit einem Baum verwechselt. Endlich im Taxi beschließt man, die 100 Dollar zu vermehren, also ab ins Casino. Black Jack und „einarmige Banditen“ stehen zur Auswahl. Hat man seine Dollars vervielfacht, kann endlich ein bestimmtes Etablissement aufgesucht werden (man ist ja auch nur Mann). Um einen 100er erleichtert geht es dann weiter auf die Suche nach der Traumfrau. Die Disco wird auserkoren, doch der Türsteher verlangt eine Clubkarte. Woher nehmen, wenn nicht stehlen? ...

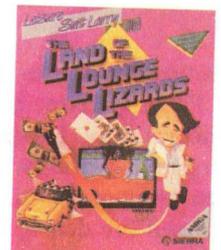
Bis Larry Laffer seine Traumfrau gefunden hat, ist es ein langer Weg und der Spieler wird einige Stunden voller Freude mit dem Programm ha-

ben. Für jüngere Generationen ist es allerdings nicht geeignet, da das Programm kein Blatt vor den ‚Bildschirm‘ nimmt. Um junge User vom Spielen abzuhalten, werden zu Beginn des Programms sechs Fragen gestellt die nicht einfach zu beantworten sind.

Die Grafik des Programms ist für Amiga-Verhältnisse schlecht, trotz Animation von Objekten, ebenso verhält es sich mit dem Sound, nur wenige Klänge ertönen aus dem Lautsprecher, den Spaß, den man

aber mit dem Programm hat, wiegt die schlechte Grafik und den mangelnden Sound wieder auf. Immer neue Gags haben die Programmierer dem Programm mitgegeben. Gesteuert wird Larry wahlweise mit Joystick, Maus oder Tastatur. Ein kleines Begleitheft mit einigen Informationen liegt dem Programm bei.

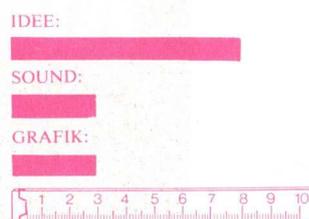
Wer viele lustige Gags mag, nicht zu prüde ist und sich nicht durch die mangelnde Grafik abschrecken läßt, wird viel Spaß an dem Programm haben. (AK)



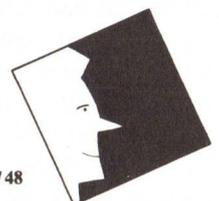
Members Only, ohne Clubkarte kein Einlaß.



Ein Flirt mit Faith genießt auch Larry.



Anbieter: PDC, Tel.: 06172/24748
Preis: 95,- DM



Eine der beliebtesten Spielereien auf dem Amiga – wenn man das noch Spielerei nennen darf – ist die Beschäftigung mit Grafik jeglicher Art. Das beliebteste Programm dafür ist Deluxe Paint II. Die Vorzüge sind seine Bedienungsfreundlichkeit, seine unzähligen Funktionen und die Kompatibilität zu anderen Grafik-Programmen durch das IFF-Format. Dieser Grafik-Kurs soll Tips, Tricks und das nötige Know-How vermitteln, die nicht in der Anleitung stehen. Er ist gleichermaßen für Laien

und für Profis gedacht. Die Beispiele sind so gewählt, daß jeder Leser das finden soll, was er für seine Bilder und Grafiken gebrauchen kann. Natürlich müssen Sie mit Deluxe Paint II schon ein wenig vertraut sein. Schnell entworfene, aber hochqualitative Grafik: Darunter verstehe ich Grafik, die in kurzer Zeit zu zeichnen ist und einen Umfang von Titelbildern über Hintergründe für eigene Spiele bis hin zur Geschäfts- und Kalkulations-Grafik hat.

Wichtig ist nicht nur, wie man Rat-

schläge wirkungsvoll in eigene Bilder umsetzt. Sie sollen auch lernen, Tips und Tricks miteinander zu kombinieren, um noch bessere Effekte zu erzielen und Zeit zu sparen. Durch die verschiedenen Beispiele will ich versuchen, Ihnen ein Gefühl für das richtige Anwenden der Funktionen von Deluxe Paint II zu vermitteln. Sind Sie neugierig? Gut, dann fangen wir gleich mit dem Kurs an!

Auch kleine Auflösungen bringen großes zustande.

TIPS und TRICKS mit Deluxe Paint II

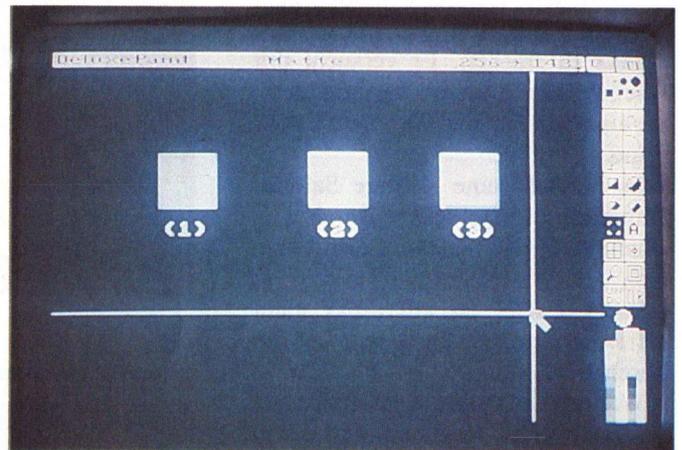
Tip 1

Auflösung: Lores 320x200 Punkte.

Farbenpalette: Sie benötigen zwei Farbverläufe. Den ersten für den Untergrund zwischen drei Farben und den zweiten für die Schrift. Hierfür können Sie den Rest der Farben benutzen. Der Untergrund zu diesem Logo wurde mit Hilfe eines Vierecks in der Größe von 30x30 Punkten erstellt (Bild 1). Ziehen Sie zwei Linien, jeweils in der Farbe, die heller gelegt ist als die Farbe des Vierecks, entlang der linken oberen Seite, und in der Farbe die dunkler gelegt ist, entlang der rechten und der unteren Seite. Lösen Sie nun das Viereck mit Hilfe der Brush-Funktion von der Zeichenfläche ab und löschen Sie den Bildschirm. Drehen Sie jetzt mit Hilfe der Do-Funktion den Brush um die X-Achse nach hinten, stellen Sie die Anti-Antialias-Funktion auf „High“ und füllen Sie den Bildschirm.

Schrift: Sie gehen jetzt in den zweiten Screen und schreiben ein beliebiges Wort auf den Bildschirm (Bild 2.1). Es sollte in einer großen Schriftart geschrieben sein und sich nur über eine Zeile erstrecken (siehe Bild 1). Zeichnen Sie jetzt ein zweites Viereck, das so hoch und so breit ist

Grundform für den Hintergrund



wie der größte Buchstabe des Wortes (Bild 2.2). Ziehen Sie eine Linie in ei-

nem Winkel von 45 Grad von der rechten oberen Ecke in die linke un-



Die Stufen zum Erzeugen der verwendeten Pinsel (brush)

**DAS BUCH ZU DELUXE PAINT II
„PROFESSIONELLES ARBEITEN MIT DELUXE PAINT II“**

Autor: Walter Friedhuber

Auszug aus dem Inhalt: Grundtechniken ★ Schriftgestaltung ★ Schattierungen ★ Perspektivische Darstellung ★

Stencil: Arbeiten mit Schablonen ★ Animationssequenzen ★ 3D-Bilder ★ Fotoretusche ★

Über 100 Bilder und Illustrationen.

Bestellnr. 4001 ★ Buch ★ DM 58,00

Bestellnr. 4002 ★ Disk1 zum Buch: Bilder ★ DM 25,00

Bestellnr. 4003 ★ Disk2 zum Buch: Animation ★ DM 25,00

FÜR AMIGA 500 ★ 100 ★ 2000



THE BEST OF AMIGA ART MACHINE

Eine Diskette randvoll mit den besten Bildern des Teams
1 Diskette ★

Bestellnr. 2000
DM 29,90

NEU: FANTASY DRAMS ★ Bestellnr. 2002 ★ 29,00 DM

AMIGA ART MACHINE COMPUTERMALERSCHULE

Schritt für Schritt die eigene Kreativität entdecken. Handbuch und Bilderspiele unterstützen Sie dabei. Geeignet für alle bekannten Malprogramme wie Graphicraft, Aegis Images, Deluxe Paint.

COMPUTERMALERSCHULE NR. 1: FANTASY
Bestellnr. 3000 ★ Handbuch und Diskette ★ DM 49,00

COMPUTERMALERSCHULE NR. 2: LÄNSCHAFTEN
Bestellnr. 3001 ★ Handbuch und Diskette ★ DM 49,00

COMPUTERMALERSCHULE NR. 3: ANIMATION
Bestellnr. 3002 ★ Handbuch und Diskette ★ DM 49,00

NEU: LANDSCAPE DESIGNER

Professionelle Fractallandschaften selbst erstellen
Bestellnr. 7000 69,00 DM

DESIGNER CONSTRUCTION SET VOL. 1 APOKALYPSE

Vor der hereinziehenden Kälte Afrikas vollzieht sich ein Drama, das zum Übergang des Planeten führt. Über 80 Bilder und Objekte. Bestellnr. 1001 ★ 2 Disketten u. Handbuch ★ DM 98,00

DESIGNER CONSTRUCTION SET VOL. 2 STARBIRD

Eine fantastische Zeitreise durch die Unendlichkeit des Alls. Mit 3D-Sequenzen, digitalisierten Objekten und vielen neuen Tips. Bestellnr. 1002 ★ 2 Disketten u. Handbuch ★ DM 98,00

3D-SHOW

Sensationelle Bilder, die Sie in die faszinierende Welt der dritten Dimension einführen. 3D-Brille liegt bei.

Bestellnr. 5000 ★ 1 Diskette u. 3D-Brille ★ DM 45,00

Verand per Nachnahme
Versandkostenzuschuss DM 7,00
Händleranfragen erwünscht

AMIGA

COMPUTERGRAFIK ★ GABRIELE LECHNER

Planeggerstr. 1 ★ 8000 München 60

Tel. 089/834 05 91

Bürozeiten: 9.00 – 17.30 Uhr

FILIALE: HARALD SCHOTT

Klosterstr. 6

6505 Mainz/Kastel

Tel. 0 61 34 / 67 86 ab 14.00 Uhr





Das Ergebnis aller Bemühungen ist schon beeindruckend

tere Ecke in dem dunkelsten Ton des zweiten Farbverlaufs. Nun ziehen Sie Linien links und rechts von der ersten Linie in den immer heller werdenden Farbtönen. Wenn Sie bei dem Grundton angekommen sind, wiederholen Sie das Ganze, jedoch verschoben, und lassen Sie immer einen Farbton dazwischen aus. Ziehen Sie den Brush mit Hilfe der Line-Funktion größer und nehmen ihn wieder als Brush (Bild 2.3). Dann schützen Sie alle Farben bis auf die Farben des aktuellen Brushes und die Fläche der Schrift. Setzen Sie den Brush nebeneinander auf die Schrift (Bild 2.4), und das Logo ist fertig.

zen Sie die gespiegelte Schrift mit ihrer oberen Kante an die untere Kante der auf dem Untergrund vorhandenen Schrift. Abspeichern und fertig!

Tip 2

Auflösung: Lores 320x200 Punkte.

Farbpalette: Sie brauchen eine Farbpalette von Weiß nach Lila. Ziehen Sie einen Range über diesen Farbverlauf.

Untergrund: Der Untergrund dieses Bildes wurde durch Füllen des Bildschirms in einer mittleren Farbe des Farbverlaufs erzielt. Zeichnen Sie ein Viereck von 20x20 Punkten,

oberen, unteren, rechten und linken Rand. In den Ecken verbinden Sie jeweils die Linien der zueinander führende Ränder (Bild 1). Beginnen Sie dabei am besten mit der äußersten Linie.

Schrift: Gehe Sie nun in den zweiten Screen und schreiben Sie in der hellsten Farbe Ihres Farbverlaufes ein beliebiges Wort auf den Bildschirm. Grenzen Sie es nun mit der Brush-Funktion ein und lösen Sie sie ab.

Effekt: Sie bleiben auf dem zweiten Screen und löschen ihn. Setzen Sie den Brush nun auf den Untergrund und wählen Sie aus dem Modemenu den Color-Modus. Jetzt wählen Sie die dunkelste Farbe aus der Farbpalette und setzen den Brush etwas nach unten und rechts versetzt auf den sich schon auf der Arbeitsfläche befindlichen Schriftzug. Der Brush soll in der Hintergrundfarbe des ersten Screens zwischen die beiden Schriften sein. Jetzt nur noch die fertige Schrift als Brush nehmen und auf den ersten Screen setzen und fertig!

Tip 3

Auflösung: Lores 320x200 Punkte.

Farbpalette: Für diese Bild be-



Wenig Aufwand für gute Effekte

Effekt: Sie nehmen die Schrift als Brush und gehen wieder auf den ersten Screen mit dem Untergrund. Dort setzen Sie die Schrift in die Mitte des Bildes und spiegeln den Brush um die X-Achse. Verzerrten Sie ihn ein bißchen in die rechte untere Ecke. Schützen Sie die zwei Farben an den Rändern des ersten Vierecks und set-



...mit den Rahmen ergibt sich ein gelungenes Bild

in dem Sie Linien von oben nach unten ziehen. Beginnen Sie an den äußersten Rändern mit der dunkelsten Farbe und lassen Sie es nach innen hin heller werden (Bild 2.1). Lösen Sie das Viereck mit Hilfe der Brush-Funktion vom Bildschirm ab, und aktivieren Sie die Line-Funktion. Jetzt ziehen Sie eine Linie, jeweils an dem

nötigen Sie einen Farbverlauf, der fließend von weiß nach lila übergehen sollte. Untergrund: Für den Untergrund dieses Logos müssen Sie ein Viereck mit der Größe von 20x20 Punkten zeichnen. Ziehen Sie nun Linien von der oberen zur unteren Kante des Vierecks, wobei zu beachten ist, daß in der Mitte die hellste



Eine Säule bildet den Ausgangspunkt für die Gestaltung des Hintergrundes

Farbe (also weiß) und nach außen hin dunklere Farben benutzt werden. Schneiden Sie dieses Viereck jetzt mit Hilfe der Brush-Funktion heraus, und wählen Sie aus der Werkzeugleiste die Line-Funktion. Ziehen Sie nun eine Linie vom oberen zum unteren Rand des Bildschirms und nehmen diese als Brush. Löschen Sie den Bildschirm. Aktivieren Sie jetzt die Line-Funktion mit der rechten Maustaste, schalten Sie Spacing ein und stellen es auf 22. Ziehen Sie nun eine Linie von links nach rechts über den Bildschirm.

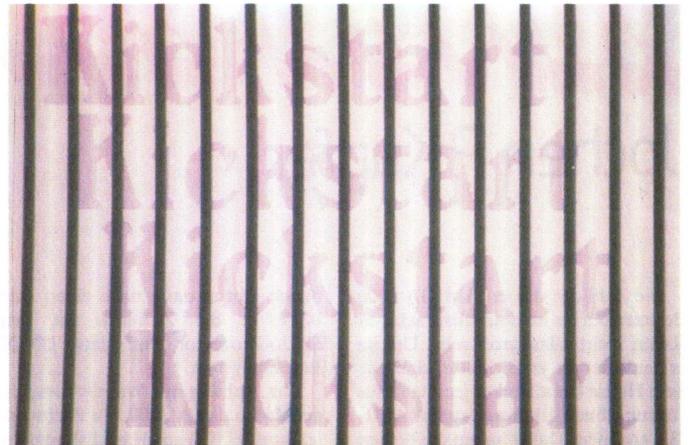
Schrift: Wechseln Sie nun in den zweiten Screen und schreiben Sie ein beliebiges Logo auf den Bildschirm. Diesen lösen Sie als Brush vom Zeichenbrett und wechseln wieder auf den ersten Screen. Effekt: Aktivieren Sie nun in dem Mode-Menü die Shade-Funktion und setzen Sie den Brush auf den Untergrund,

wobei durch Drücken der rechten bzw. der linken Maustaste der Abdruck heller oder dunkler wird.

Tip 4

Auflösung: 320x200 Punkte.

Farbenpalette: Für dieses Bild müssen Sie einen Farbverlauf von Weiß nach Lila definieren. Untergrund: Für den Untergrund zu diesem Bild benötigt man ein Viereck mit der Größe von 20x20 Punkten. Ziehen Sie jetzt Linien von der oberen zur unteren Kante des Vierecks. Aber in der Mitte mit der hellsten Farbe anfangen und nach außen hin dunkler werden lassen. Lösen Sie nun das Viereck als Brush vom Bildschirm, und stellen Sie es im Effects-Menü unter Perspective die Anti-Antialiasing Funktion auf „High“. Gehen Sie jetzt auf die „Do“-Funktion und biegen den Brush um die X-Achse nach



Na, ist das ein Ergebnis!

hinten. Lassen Sie nun den Bildschirm füllen.

Schrift: Gehen Sie jetzt in den zweiten Screen und schreiben Sie eine beliebige Schrift auf den Bildschirm. Anschließend füllen Sie sie mit einem Farbverlauf. Setzen Sie die Schrift nun auf den Untergrund des ersten Screens. Effekt: Spiegeln Sie die Schrift nun mit Hilfe der „y“-Taste. Biegen Sie sie im Brush-Menü in der Bend-Funktion unter Horizontal (Bild 2.3) und schalten im Mode-Menü die Shade-Funktion ein. Setzen Sie nun die gespiegelte Schrift an die untere Kante der sich schon auf dem Untergrund befindlichen Schrift.

Damit sind Sie auch schon am Ende des ersten Teils unseres Kurses gelangt. Im nächsten Heft werden wir einige Tips zu Hintergrundgrafiken für Spiele besprechen.

(Sebastian Dosch)



Erst wird die Schrift bearbeitet...



...und dann auf den Hintergrund gebracht

Buchbesprechung:

„Das große Amiga 2000 Buch“

Jochen P. Kupfer

Die Zahl der stolzen AMIGA 2000 Besitzer dürfte im deutschsprachigen Raum nicht allzu groß sein. Um so erfreulicher ist es für diese, daß aus dem Hause Data Becker jetzt ein Einführungsbuch für den großen AMIGA erschien. Dabei haben die beiden Autoren, Rügheimer und Spanik, es verstanden, ein trockenes Thema durch ihre bekannt flapsige Sprache schmackhaft darzubieten. Denn wer liest sonst schon gerne Handbücher?

Etwa 130 Seiten des Buches beschäftigen sich mit dem Thema „Einstiegen und Durchstarten“. Im 2. Teil, der genauso lang ist, wird die Frage 'Software selbst erstellen oder kaufen?' behandelt. Dieses Kapitel geht auch auf das Thema 'Software und Harddisk' ein. Den weitaus größten Raum beansprucht das 'Aufsteigen mit dem AMIGA 2000'. Hier widmen sich die Autoren den Hoffnungen und Wünschen der AMIGA 2000 Anwender, die ihre Supermaschine gerade wegen dieser schier unbegrenzten Ausbaumöglichkeiten kauften. Dies war sicher eine echte Marktlücke, denn aus der Commodore-Gerüchteküche bekommt man oft nur all zu wenig konkrete Informationen über die tatsächlichen Aufrüstungsmöglichkeiten. Die 2–8 Megabyte Hardwareerweiterungen und der Anschluß an einen Videorecorder werden beschrieben. Der AMIGA 2000 im PC-Betrieb kommt auch nicht zu kurz. PC-Erweiterungskarten und PC-Festplatten werden ausführlich beschrieben.

Der AMIGA-Anwender ohne PC-Ambitionen bekommt eine Menge Informationen über die AMIGA-Harddisk. Das geht bis zum fertigen Installationsprogramm. Als Lecker-

bissen gibt es dann noch einen Ausblick in die Zukunft mit dem 68020-Prozessor und dem UNIX-Betriebssystem.

Der Anhang mit Wandervorschlägen durch den A 2000, ein Fachwortlexikon, Hilfen zur Harddisk und anderes mehr runden das Bild ab.

Fazit:

„Das große AMIGA 2000 Buch“ von C. Spanik und H. Rügheimer ist ein Werk, das eigentlich jedem AMIGA 2000 beiliegen sollte (in grauer AMIGA-1000-Vorzeit war das noch guter Brauch!). Denn gerade der 1. Teil des Buches könnte dem Einsteiger über manche unnötige Hürde hinweg helfen. Wer allerdings die Anfangsschwierigkeiten bereits gemeistert hat, dem bringen die ersten 300 Seiten, außer einem gelegentlichen Aha-Effekt nicht mehr viel. Interessanter sind da schon die Aufstiegsmöglichkeiten, denn hier haben sich die Autoren wirklich verdient gemacht. Ihr lockerer Stil hat das Buch jedoch auf 684 Seiten aufgebläht, so daß es sich zum Nachschlagen weniger eignet. Es ist wohl auch mehr als spannende Story gedacht.

Rügheimer, Spanik:
Das große Amiga 2000 Buch.
DATA BECKER GmbH
684 Seiten, DM 59,-

Anbieter:
PDC, Bad Homburg ☎ 0 62 72/2 47 48
DTM, Wiesbaden ☎ 0 61 21/56 00 84
IM, Frankfurt ☎ 0 69/7 07 11 02

Preis: DM 195,-

auch für diese geeignet. Wie der Titel schon sagt, dreht es sich in dem Buch um die Assembler-Programmierung von PC's unter MS-DOS.

Im ersten Teil wird auf die grundlegende Bedeutung von Maschinensprache eingegangen. Es werden die Unterschiede zu Hochsprachen aufgezeigt und Programmierungswerkzeuge zur Assembler-Programmierung vorgestellt. Der zweite Teil ist

voll und ganz dem Debugger des MS-DOS gewidmet. Darauf folgt dann eine tiefgreifende und umfangreiche Einführung in das Grundwissen zur Assemblerprogrammierung. Des weiteren wird auf den MASM von Microsoft eingegangen. Die Erstellung von Programmen wird anhand von einfachen und verständlich aufgebauten Beispielen gezeigt. In einem späteren Teil des Buches werden dann die speziellen Hilfsmittel des MacroAssemblers behandelt. Alle anderen Kapitel beschäftigen sich mit speziellen Themen der Assemblerprogrammierung: Interruptprogram-

mierung, Zusammenspiel von Assembler und Hochsprachen (hierbei wird besonders auf Basic, C und Turbo Pascal eingegangen), MS-DOS-Befehle und mathematische Operationen in Assembler. Die Interrupts und die Algebra sind hierbei besonders zu erwähnen, da dies Bereiche sind, die dem Neuling (und nicht nur diesem) oft Schwierigkeiten bereiten. In diesem Buch wird darauf ausführlich eingegangen und ein Einstieg in die Materie wird erleichtert. Als Zielgruppe hat dieses Buch nicht nur die Anfänger in Maschinensprache, sondern jeder Assemblerprogrammierer wird bei der täglichen Arbeit gut beraten sein, das Buch griffbereit zu haben.

SCHIEB
PC-Maschinensprache
DATA-BECKER VERLAG
600 Seiten / 49,- DM



„Das große Amiga 2000 Buch“

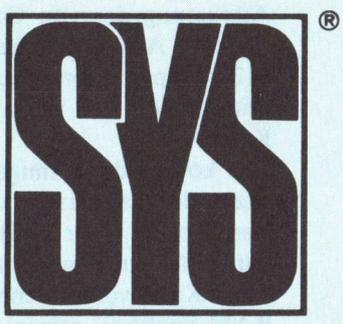
Hier wird heute ein Buch vorgestellt, das sich nur mit einem Teil des AMIGA's befaßt: der PC-Hardware. Das Buch ist zwar nicht speziell für Sidecar oder Bridgeboard geschrieben worden, ist aber durch die IBM-Kompatibilität der beiden,

**KICK
START**
AMIGA ZEITSCHRIFT
Heim-Verlag

Besuchen
Sie uns

MERLIN
computer gmbh

Halle 21 · Stand C15 / C17



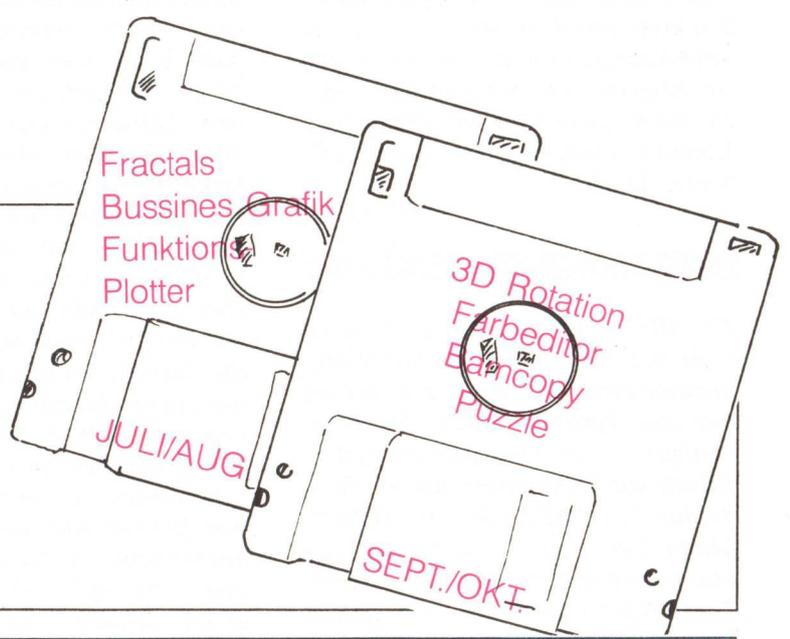
19. – 23. Oktober 87
SYSTEMS 87
München
Messegelände



Disketten-Service

Ab jetzt können Sie alle Programme, die in der KICKSTART veröffentlicht wurden, auf Diskette bestellen.

Juli/August 19,- DM
September/Oktober 19,- DM



Heim-Verlag
Heidelberger Landstr. 194
6100 Darmstadt-Eberstadt
Telefon: (0 61 51) 5 60 57

BESTELL-COUPON
Hiermit bestelle ich
 durch beigefügten Scheck
 per Nachnahme
zuzüglich 5,- DM Versandkosten

Gewünschte Artikel aufführen

Name _____
Straße _____
PLZ/Ort _____



Fortran 77

Ich besitze seit einigen Monaten einen Amiga 500. Da ich ihn auch beruflich (Elektrotechnik) nutze, interessiert mich insbesondere die Möglichkeit, IBM-Programme auf dem Amiga laufen zu lassen (und umgekehrt) bzw. source-codes in Fortran 77 auf dem Amiga zu erstellen und diese dann auf einem IBM-AT (oder XT) zu compilieren, oder laufen zu lassen.

Ich würde es begrüßen, wenn Sie Ihre Erfahrungen in Bezug auf dieses konkrete Problem, aber auch mit der IBM-Kompatibilität des Amiga-500 im Allgemeinen, mir mitteilen, oder in Ihrer Zeitschrift veröffentlichen könnten (vielleicht als Leserbrief). Vielen Dank!

Dirk Kilfitt

Die IBM Kompatibilität eines A500 ließe sich nur mit einem Softwareemulator erreichen. Es ist z.Z. jedoch nur eine Version für den A1000 im Umlauf. Dieser Emulator unterstützt jedoch nur Programme, die im Textmodus des IBM laufen (Word Star, Masm, Debug!! und v.a.m.). Sie können jedoch auf dem Amiga erstellte Texte (Quellcodes) über die serielle Schnittstelle und ein sogenanntes Nullmodemkabel auf andere Rechner übertragen. Sollten beide Rechner räumlich getrennt stehen (einer zu Hause, der andere in der Uni), können Sie ein Programm benutzen, das auf dem Amiga direkt PC bzw. MS DOS Disketten schreiben und lesen

kann. Ein solches Programm ist unter dem Namen Dos2Dos im Handel. Sie benötigen jedoch ein externes 5 1/4" Laufwerk, in welchem Sie die PC Disketten lesen können (die neue Generation IBM Rechner unterstützt jedoch auch 3 1/2" Laufwerke, so daß Sie unter diesen Umständen kein 5 1/4" Laufwerk benötigen, sondern mit einer externen Amiga-floppy arbeiten können). Speziell Ihr Problem mit Fortran 77 betreffend, existiert in Amerika eine Firma, die einen F77 Compiler für den Amiga in ihrem Programm hat. Sie könnten dann die komplette Entwicklung auf dem Amiga vornehmen. Die Firma Absoft ist unter folgender Telefonnummer zu erreichen: 001 313 835 0050.

Die Redaktion

Betr.: Leserbrief zu Nr. 10/87, S. 92, Frage von M. Schiffer

Sehr geehrter Herr Schiffer, um ein Basicprogramm direkt aus der Startup-Sequence heraus anzustarten, braucht man nur folgende Zeile dort aufzunehmen:

[Pfad]AmigaBASIC [Pfad]IhrProgramm
Sie können das natürlich auch mit run... im Hintergrund laufen lassen. Bitte achten Sie bei dem nachzuladenden Maschinenprogramm darauf, daß es an einer geraden Speicheradresse beginnt, sonst gibt es bei einem Aufruf unweigerlich den Guru! Wenn Sie eine externe Floppy und keine interne haben, sind Sie für manche Anwendungsfälle sogar besser gerüstet als mit einer internen: Sie können nämlich jetzt per assign alle möglichen Laufwerke auf ein df1: umleiten, das Sie vorher per assign z.B. mit df2: identifiziert haben. Es gibt ja leider Anwendungsprogramme, die einem nur die Wahl zwischen df0: und df1: lassen; Besitzer von zwei internen Laufwerken und einer Harddisk können letztere von diesem Programm aus nie benutzen. Sie aber können mit assign df1: df2: und assign df0: df1: dieses virtuelle Laufwerk durchaus umlegen!

Dr. Peter Kittel, Commodore Ffm.

Speichererweiterung für Amiga 1000 und Programmiersprache LOGO

Sehr geehrte Damen und Herren, in der Ausgabe Nr. 10 Ihrer Zeit-

schrift wurden die zwei Speichererweiterungen für den Amiga 1000 vorgestellt, die sehr interessant sind, weil diese endlich im Bereich von ca. DM 300,- liegen. Sie haben aber einen Nachteil: es fehlt die Hardwareuhr.

Da die Hardwareuhr schon die originale (von Commodore) Speichererweiterungskarte für den Amiga 500 (auch im Preis von DM 300,-) hat, wäre es bedauerndswert, wenn die Uhr dem Amiga 1000 fehlen würde.

Aufgrunddessen habe ich an Sie folgende Fragen:

1. Gibt es auch Hersteller, die eine Speichererweiterung mit einer Uhr für den Amiga 1000 bei dem Preis ca. 300,- bis 400,- DM anbieten?
2. Warum bietet man eigentlich nicht für den Amiga 1000 eine Anpassung der von Commodore für den Amiga 500 hergestellten 512 kB Speichererweiterungskarte mit Uhr (die von unten montierte Karte) an?
Das könnte dem Amiga 1000 Besitzer den saubersten und billigsten Weg zu einer 1MB Speicherkapazität (zusammen mit schon beim Kauf vorhandenen 512 kB RAM-Speicher) zeigen.
Warum sollte das nicht möglich sein? Die eventuell erforderlichen Lötarbeiten dürften kein Hindernis bringen. Das fordert man auch bei den angebotenen Speichererweiterungen.
3. Dem Amiga-Besitzer macht die Tatsache, daß bis heute die Programmiersprache LOGO für den Computer noch nicht vorhanden ist, große Schmerzen.

Worin liegt hier das Problem? Commodore hat schon für den C-64 diese Sprache sehr gut geschrieben. Wann kann diese Sprache für den Amiga kommen?

Mark Marpelt

Zu Punkt 1:

Soweit uns bekannt ist, im Moment nicht. Es werden jedoch billig Uhrenmodule angeboten (z.B. Timesaver), so daß der Preis bei einer Erweiterungsuhr auf den von Ihnen angestrebten Betrag von 400,- DM kommt.

Zu Punkt 2:

Die angebotene Erweiterungskarte für den A500 enthält keine Adressdekodierung, da sich diese bereits im Rechner befindet. Sie weicht damit grundsätzlich vom Konzept ab, das auf dem A1000 verwendet werden müßte. Sie soll jedoch ein Ansporn an Bastler darstellen. Eine Bauanleitung in dieser Richtung wäre sicher für viele von Interesse.

Zu Punkt 3:

Leider war über die Programmiersprache Logo nichts in Erfahrung zu bringen. Die Redaktion bittet auf diesem Weg die entsprechenden Firmen um Zusendung einer News.

Die Redaktion

Einige Leser haben Probleme mit englischen Sonderzeichen und einem deutschen Tastaturtreiber. Die folgende Tabelle hilft Ihnen hier weiter.

Gewünschtes Zeichen	Tastenkombination
@	<Shift><Alt><2>
.	<Shift><Alt><W>
~	<Shift><Alt><6>
~	<Shift><Alt><[>
{	<Shift><Alt><Ü>
}	<Shift><Alt><*>
[<Alt><ü>
]	<Alt><*>
'	<Alt><ä>

Einige dieser Tasten sind bei den verschiedenen Tastaturen (A500, A1000 und A2000) direkt erreichbar, so daß dann die entsprechenden 'Fingerübungen' entfallen können.

Die Redaktion

Sehr geehrte Redaktion,

hiermit möchte ich einen Lösungsvorschlag für das Problem von Michael Schiffer geben. Herr Schiffer wollte mit einem Basicprogramm eine Passwortabfrage realisieren, die bei Eingabe des richtigen Passwortes ein Maschinenspracheprogramm nachlädt. Das alles sollte sich in der startup-sequence abspielen. Nun die Lösung des Problems:

1. AmigaBasic (Version 1.2!!!), Passwortabfrageprogramm und Maschinenspracheprogramm auf

die gewünschte Diskette kopieren.
2. Die startup-sequence folgendermaßen ändern:

AmigaBasic [Name des Basic-Passwortprogrammes]

[Name des Maschinenspracheprogrammes]

Für das Basicprogramm muß nichts besonderes beachtet werden, außer daß im Falle der richtigen Eingabe ein SYSTEM-Befehl ausgeführt werden muß. Natürlich sollte das Programm noch einige andere Merkmale haben, so sollte z.B. die Tastenkombination Ctrl-C (= Stop) abgefangen werden. Außerdem sollte die Eingabe in einem Fenster ohne Schließ- und Backgadget stattfinden, da man sonst die Eingabe ganz einfach umgehen kann, indem man das Schließgadget anklickt. Hier ist mein Vorschlag:

```
ON BREAK GOSUB ctrlc
BREAK ON
WINDOW 1,"Passwort",(0,0)-(631,185),0

loop:
PRINT "Passwort:";
eingabe:
xs=INKEY$
IF xs=CHR$(13) THEN weiter
as=a$+UCASE$(xs)
PRINT UCASE$(xs);
GOTO eingabe

weiter:
PRINT
IF as="AMIGA" THEN SYSTEM:REM Passwort = AMIGA
x=x+1
IF x=3 THEN falsch
as=""
GOTO loop

falsch:
PRINT "War wohl nix !!!"
WHILE 1
WEND

ctrlc:
RETURN
```

Leider läßt sich selbst dieser Schutz umgehen. Man könnte z.B. die startup-sequence schon mit Ctrl-D abbrechen, bevor das Basicprogramm gestartet wird. Aber besser geht es in AmigaBasic (meiner Meinung nach) nicht. Selbst ein Maschinenspracheprogramm im Boot-Sektor wäre nicht optimal, da man ja einfach mit einer anderen Diskette starten könnte.

Ein weiteres Problem Herrn Schiffers war, daß sein externes Laufwerk nur als df2: anerkannt wird. Abhilfe schafft hier ein einfaches

ASSIGN DF1: DF2:

in der startup-sequence.
in der startup-sequence.

Außerdem wollte Herr Stefan Tandecki wissen, wie sich einige grafi-

sche Tricks in AmigaBasic programmieren lassen. Da die Erklärung wohl etwas zu lang für die Leserecke wäre, möchte ich hier das Buch „Amiga Supergrafik“ aus dem Data-Becker-Verlag empfehlen, in dem diese Dinge (mit Beispielprogrammen) erklärt werden.

Ich hoffe, daß den beiden Herren (und vielleicht auch einigen anderen Amiga-User(innen)n hiermit geholfen ist.

Mit freundlichen Grüßen
Jörn Eyrich

INSERENTEN-VERZEICHNIS

Bellingrath	34
CSJ	53
CWTG	34
DTM	98
Dast	48
Data Becker ..	35, 39, 45, 47
Fischer	25
Hagenau Computer ...	28
Idee-Soft	25
Interplan	50
Jumbo Soft	73
Kingsoft	2
Kupke	116
Lechner	7
Maier	98
Medien-Center	63
PDC	66
Padercomp	95
Philgerma	25
Software2000	25
SoftwareLand	54, 55
Soyka	115
Trumpp	34
Tröps	95
Versalia	98

VORSCHAU:

AKTUELLES:

- **SYS 87**
Was gibt es Neues auf der SYSTEMS in München

HARDWARE:

- *Ein neues GENLOCK der Firma MERKENS stellt sich vor*
- *MONITORE: Neben anderen Modellen wird der neue COMMODORE-Monitor vorgestellt*
- *Außerdem gibt es Tests über SOUND-DIGITIZER von Hagenau Computer, VIDEO-DIGITIZER und DRUCKER (der neue NEC P2020 und der STAR NB 24)*

SOFTWARE:

- *Wieder gibt es eine Menge von Produkten, die ihr Können unter Beweis stellen müssen. Als Tests sind geplant: CITY DESK, MODULA-2, Singel-Pass-Compiler, SILVER Ray-Tracer ANIMATOR, TRUE BASIC und TURBO BASIC (beide für IBM)*

GRUNDLAGEN:

- *Einen Einblick in die faszinierende Hardware des AMIGA bietet der Grundlagenartikel über die Programmierung des CHOPPERS.*

Änderungen vorbehalten!

Ab 13. November an Ihrem Kiosk!

Impressum

KICKSTART

Chefredakteur:

Uwe Bärtels (Chefredakteur) (UB)
Markus Nerding (Stellvertreter) (MN)

Redaktion

Andreas Krämer (AK)
Gerald Carda (GC)
Dipl. Ing. Harald Schneider (HS)
Marcelo Merino (MM)
Harald Egel (HE)

Herausgeber:

„MERLIN“ Computer GmbH
Industriestraße 26
Postfach 55 69
6236 Eschborn
Tel.: 0 61 96 / 48 18 11
FAX: 0 61 96 / 4 11 37

Ständige Mitarbeiter:

Christian Keller (CHK)
Andreas Diebold (AD)
Wolf Dietrich (WD)
Andreas Suchy (AS)

Public Relations:

Claus Peter Lippert

Verlag:

Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt 13
Tel.: 0 61 51 / 5 60 57
FAX: 0 61 51 / 5 56 89

Verlagsleitung:

Hans-Jörg Heim

Anzeigenverkaufsleitung:

Uwe Heim

Anzeigenpreise:

nach Preisliste Nr. 1, gültig ab 1.7.86

Produktion:

Klaus Schultheis
Patricia Illing

Grafische Gestaltung:

Rosina Altkorn

Fotografie:

Klaus Ohlenschläger

Titelgestaltung:

Fabian & Maier

Titelfoto:

Rainer Spirandelli

Satz:

Mohr Fotosatz, Pfungstadt

Druck:

Ferling Druck, Darmstadt

Zeugschriftenhandel:

Zeitschriftenhandel, Kauf- und Warenhäuser,
Commodore-Fachhändler oder direkt beim Verlag

Kickstart erscheint 11 mal im Jahr

Einzelpreis: DM 7,-, ÖS 56,-, SFr. 7,-

Das Jahresabonnement kostet DM 70,- inkl.
Versandkosten + MwSt.

Europ. Ausland DM 90,- inkl. Versandkosten

Alle in der 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 erlaubt.

Programmlistings, 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, Aufbauski-zen, Stücklisten, usw., die zum Nichtfunktionieren oder evtl. zum Schadhafwerden von Bauelementen führen, wird keine Haftung übernommen.

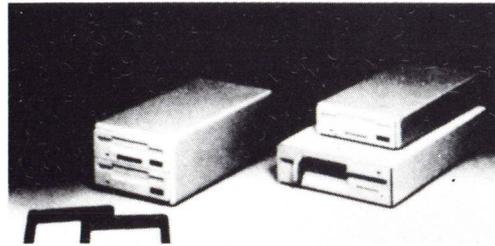
© Copyright Heim Verlag

Hattinger Str. 685
4630 Bochum 5
☎ 02 34/41 19 13
41 19 47

Soyka

DATEN **TECHNIK**

AMIGO	F1 3,5" Einzelfloppy	349,-
AMIGO	F2 3,5" Doppelfloppy	649,-
AMIGO	F5 5,25" Einzelfloppy	489,-
AMIGO	F1 Bausatz	289,-
AMIGO	F2 Bausatz	569,-
AMIGO	F1 Gehäuse	34,90
AMIGO	F2 Gehäuse	39,90
AMIGO	Sound (Mono)	129,-
AMIGO	Sound (Stereo Hifi)	199,-
AMIGO	Bootselector	39,90
AMIGO	Laufwerk NEC 1036a	259,-
AMIGO	F1 Bausatz mit Gehäuse	298,-



NEC 1036 A + Interface + Kabel + Stecker
+ Anleitung zum Selbstbau einer 3,5"

Amiga-Floppy 275,— DM

Amigo Laufwerke

- * 1 Jahr Garantie
- * Kunststoffgehäuse
- * Amigafarbe
- * Voll abgeschirmt
- * abschaltbar
- * Frontblende beige
- * LED Steuerung wie internes LW
- * „sehr zuverlässig“ (Amiga Magazin)

Amiga 500	1148,—
Amiga 2000 mit Monitor	3190,—
Amiga 5000,5 MB-Erweiterung	275,—
Amiga 500 1 MB-Erweiterung abschaltbar	289,—
Amiga 2000 1,5 MB-Erweiterung	279,—
Amiga 1000 2 MB-Erweiterung	948,—
Amiga 500 2 MB-Erweiterung	998,—
Amiga Bücher (Reference Manuals): Intuition, Hardw. Exec Libraries and Devices	je 62,50 88,—
Monitor Eizo 8060 S	1689,—

PREISSCHLAGER

Deluxe Paint II 197,—

Digi View 2.0 297,—

Anwendung			
AC-Basic	387,—	LPD Writer	247,—
AC Fortran	597,—	Macro Assembler	197,—
Acquisition	597,—	Marauder II	77,—
Aegis Sonix 2.0	167,—	Metacomco Pascal	227,—
Amiga C	257,—	Modula II Com.	597,—
Amiga Dos Express	57,—	Modula II Standart	177,—
Analyse	198,—	Modula II Developers	297,—
Analyse Graph	297,—	MusicX*	547,—
Aztec C Dev. V3,4a	597,—	Newio	477,—
Aztec C Com. V3,4a	947,—	Pagesetter Europ. Ver.	327,—
C-Monitor	197,—	Planetarium*	97,—
City Desk	227,—	Power Window	187,—
Cli-Mate 1,2	67,—	Printmaster Plus	107,—
Devpac Assembler	197,—	Pro Video	347,—
Deluxe Music Con. Set	197,—	Pro Write	247,—
Deluxe Paint IIb	197,—	Scribble	197,—
Deluxe Video V1.2	247,—	Seka	137,—
Digi Point	117,—	Shell (CLI deluxe)	137,—
Digi View 2.0	297,—	Superbase (Deutsch)	227,—
Dynamic Cad	997,—	Superbase Personal	297,—
Dynamic Word*	377,—	Toolkit	97,—
Fast Fonts	97,—	True Basic	277,—
Flip Slide	97,—	True Basic Dev. Toolkit	87,—
Fortran	597,—	TV-Text	197,—
Galileo	207,—	UBM Text V2,2	247,—
Gizmoz	127,—	USCD Pascal	167,—
Grabbit	67,—	3D Videoscape	387,—
Instant Music	97,—	VIP Professional	447,—
Lattice C V3.1	347,—	Vizawrite (Deutsch)	197,—
Lattice C prof.	887,—	Zuma Fonts Vol. 1	67,—
Lisp	417,—	Zuma Fonts Vol. 2	67,—
Logistix (Deutsch)	347,—	Zuma Fonts Vol. 3	67,—
		WordPerfect 4.1	747,—

Spielesoftware			
Adventure Constr. Set	67,—	Indoor Sports	87,—*
Alien Fires	84,—	Jet	87,—*
Alternate Reality „The city“	77,—	Kampfgruppe	107,—
Archon II	77,—	Karate Kid II	67,—
Arena	77,—	Karate King	37,—
Autoduell	77,—*	King of Chicago	97,—
Bad Cat	57,—	Land of Legends	97,—*
Barbarian	77,—	Leader Board	77,—
Bard's Tale	97,—	Little Computer People	57,—
Beat It	27,—*	Marble Madness	87,—
Black Cauldron	77,—*	Mindbreaker	27,—
Bureaucrazy	87,—	Mission Elevator	57,—
Challenger	27,—	Night Hawk	57,—
Championship Baseball	87,—	Pac Boy	27,—
Championship Basketball	87,—	Paperboy	87,—*
Championship Football	87,—	Phantasie	77,—
Championship Golf	87,—	Phalanx	27,—
Chessmaster 2000	87,—	Planetarium	97,—*
City Defence	27,—*	Portal	87,—
Cruncher Factory	27,—	Quintette	87,—
Der Hauch des Todes	57,—*	Quiwi	57,—
Defender of the Crown	87,—	Quizam	67,—
Deja Vu	87,—	Return to Atlantis	87,—*
Demolition	27,—	Roadwar 2000	27,—
Dr. Fruit	27,—	Rocket Attack	27,—
Earl Weavers Baseball	97,—	S D I	97,—
Emerald Mine	27,—	Shadow Gate	77,—*
Faery Tale	97,—	Shanghai	77,—
Ferrari Formula One	87,—*	Shooting Star	27,—
Fire Power	47,—*	Sinbad and the thrown	97,—
Flugsimulator II (neue Vers.)	97,—	Sky Fighter	27,—
Fortress Underground	27,—*	Space Battle	27,—
Galactic Invasion	47,—*	Space Fight	27,—
Garrison	67,—	Silicon Dreams	67,—*
Gauntlet	87,—*	Starglider	67,—
Golden Oldies	47,—	Strip Poker	77,—
Goldrunner	87,—	Tenth Frame	77,—*
Grand Slam Tennis	87,—	Terrorpods	67,—
Grand Prix	87,—*	Testdrive	77,—*
Guild of Thieves	87,—	The Final Trip	27,—*
Gunship	77,—*	The Surgeon	117,—
Hardball	87,—	Turbo	47,—*
Hollywood Poker	57,—	Typhoon	67,—
		Uninvited	97,—
		Winter Games	67,—
		World Games	67,—

* = in Kürze lieferbar

**Amiga 2000
4 MB Steckkarte
2 MB bestückt
999,-**

6 Gründe dafür Golem

3,5 Zoll externe Amiga-Laufwerke
Amiga 500/1000/2000



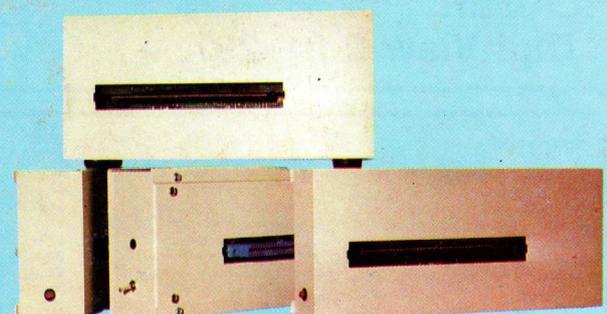
auch als Einschubdrive für Amiga 2000 **299,-**

Drives & Ram Box

1. 100% kompatibel
NEC 1036 a
2. Amiga-farbenes
Metallgehäuse
3. farblich passende
Frontblende
4. durchgeführter
Floppybus
5. Ein-/Aus-Schalter
6. Side-car und
PC-Karten
kompatibel

369,-

externe RAM-BOX Amiga 1000
auch für Amiga 500 (mit Adapter)



Golem 2 MB erweitert den „Amiga“ auf 2,5 MB

1. Autokonfigurierend
(ab Kick 1.2)
2. Amiga-farbenes
Metallgehäuse
3. durchgeführter
Systembus
4. Ein-/Aus-Schalter
5. Erweiterbar
6. Preis

2 MB 998,-

5,25 Zoll Amiga Laufwerk
40/80 Track Umschaltung, Ein-/Aus-Schalter, helle Front-
blende, Busdurchführung, 880 KB Speicherkapazität,
Side-car und PC-Karten kompatibel

479,-

500er Peripherieadapter
für 1000er Peripherie am 500er

59,90

Golem Sound

Ein Audiodigitizer der Extraklasse, kompatibel zu „Perfect
Sound“ u. „Future Sound“ Software. DIN- und Cincheingang,
Empfindlichkeitsregelung für Mikrofonanschluß

138,-

Wir liefern im 3-Tage-Rhythmus

Kupke Computertechnik GmbH
4600 Dortmund, Apelank 28
Tel.: 02 31/85 26 05