

KICK START



Test: True Basic · Logistix · Sound Scape · CLI Shell
Grundlagen: Intuition · CLI · KI · Sound-Sampling
Listings: 3D Fraktals · Business-Grafik · Funktions Plotter

ES IST AN DER ZEIT !

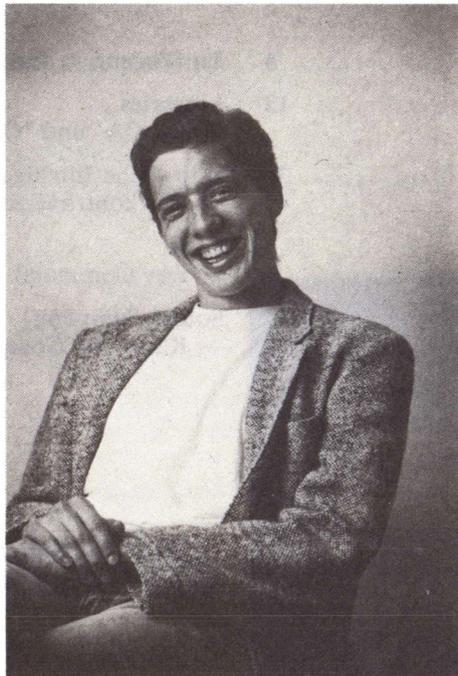
Endlich können wir Ihnen einen leistungsstarken BASIC-Compiler für den AMIGA bieten ! Jetzt können Sie Ihre Basic-Programme bis zu 50 mal schneller machen ! AC/BASIC ist kompatibel zu MICROSOFT-BASIC und zu AMIGA-BASIC - weitere Spracherweiterungen wie z. B. BLOCK IF, CASE, STATIK ! STAND-ALONE-Programme möglich. Sie können endlich Ihre Programme weitergeben oder verkaufen, ohne daß jeder im Besitz Ihrer Source-Programme ist.

AC / BASIC Compiler FÜR DEN AMIGA

AC/BASIC ist ein eingetragenes
Warenzeichen von abssoft

- Kompatibel zu Microsoft-Basic und zu Amiga-Basic
- Unterstützt alle ROM-Routinen vom AMIGA
- 30 - 50 mal schneller als der Basic-Interpreter
- STAND-ALONE-Programme möglich
- Spracherweiterungen wie z. B. BLOCK IF, CASE, STATIK
- Keine Begrenzung von Programmen oder Variablen
- Routinen können in Libraries abgelegt werden
- Unterstützt dynamische und rekursive Unterprogramme
- Läuft auch von der RAM-Disk
- 16 und 32 Bit Integer - 32 und 64 Bit Reals
- keine Zeilennummern nötig
- sequentieller und direkter (random-access) Dateizugriff
- WICHTIG: Benötigt AMIGA-DOS 1.2

DM 398,-



Liebe Leserinnen, liebe Leser!

Dem Phantom auf der Spur?

Diese Frage stellte sich unserer Redaktion im vergangenen Monat häufig. Doch der Geist war ganz realer Natur: Es handelte sich um den jüngsten Sproß der Amiga-Familie. Der A 500 wurde der Fachpresse als Gerät präsentiert, das auf dem Heimcomputer-Markt Fuß fassen sollte. Doch gerade dort war er nicht zu finden. Zwar zeigten sich vereinzelte Exemplare in den Schaufenstern einiger Händler; doch offenbar handelte es sich um mühsam ergattete Renommierstücke: Bei näherer Nachfrage erwiesen sie sich als unverkäuflich.

Jetzt war die Redaktion neugierig geworden. Mit gespannter Erwartung wählte unser Mann die Nummer von Commodore. „In kleinen Stückzahlen ab nächster Woche lieferbar“, lautete die Auskunft. Doch ganz so sicher war man seiner Sache auch bei Commodore nicht: Eine prompt aufgegebenen Bestellung wollte der nette Herr am anderen Ende der Leitung lieber nicht annehmen. Trotzdem scheinen einige Rechner bei den Händlern gelandet zu sein; und schließlich wurden wir in ei-

nem Geschäft in Bad Homburg fündig. Bei den dort verkauften 500ern handelte es sich um echt deutsche, also mit deutscher Tastatur versehene Exemplare (aufgedruckte deutsche Sonderzeichen und nicht etwa Aufkleber wie bei einigen „eingedeutschten“ Modellen des Amiga 1000).

Tatsächlich findet also eine Auslieferung der heißerwarteten Rechner statt. Daß diese Tatsache selbst beim Hersteller nicht bekannt ist, kann man der Firma nicht besonders übelnehmen; und erfahrene Commodore-Besitzer sind das wohl schon gewöhnt.

Die Trennung von Produktion und Verwaltung (in Braunschweig und Frankfurt) verbessert den Informationsfluß nicht gerade, zumal in der sicher hektischen Startphase des A-500-Projektes.

Doch Commodore wird solche Anfangsschwierigkeiten wohl bald überwunden haben. Immerhin ist der neue Amiga 500 ein Gerät, das – als 'Homecomputer' getarnt – in Bereiche vorstößt, die noch vor einigen Jahren 'professionellen' Anwendern in der Industrie vorbehalten waren. Die Fä-

higkeiten, die der Rechner in den Sparten Musik und Grafik zu diesem Preis bietet, lassen eigentlich nur einen Schluß zu: Das Warten hat sich gelohnt.

Gehen wir also frisch ans Werk und erkunden die Möglichkeiten des A 500. Viel Spaß dabei wünscht Ihnen auf diesem Weg die Redaktion. Ihren Leserbriefen mit Anregungen und Kritik (natürlich auch positiver Art) zu bestimmten Themen sehen wir gerne entgegen.

So wünsche ich Ihnen viel Erfolg mit Ihrer Neuerwerbung – und natürlich auch Spaß bei der Lektüre des neuen Heftes, das übrigens ein Doppelheft ist. Wir melden uns also erst wieder Mitte August mit der Septemberausgabe zurück. Und wenn in unserer Entwicklungsabteilung alles planmäßig läuft, kann ich Ihnen für diesen Termin einige Hardware-Lekturen versprechen.

Allen Leserinnen und Lesern wünschen die Mitglieder der Redaktion einen schönen Sommerurlaub!

Gerald Carda

Inhalt:

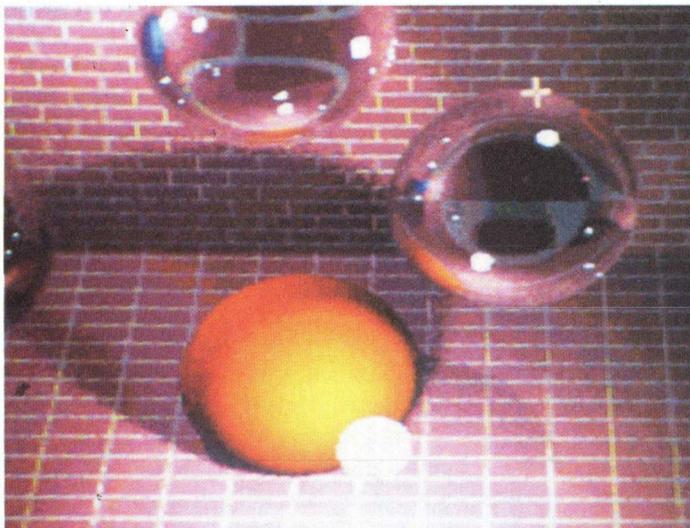
HARDWARE

- Der Wolf im Schafspelz: Amiga 5000 8
- Kontaktfreudig: Fernseher am Amiga 12
- Sound Sampling für den Amiga
– Sound Scape & Future Sound 30



Der Wolf im Schafspelz

Ein Portrait des kleinsten Sprosses der AMIGA-Familie. Werfen Sie einen Blick in sein revolutionäres Innerstes, ohne die Garantie zu verlieren.



Kontaktfreudig

Muß es unbedingt ein Monitor sein, mancher wird sich diese Frage schon gestellt haben. Dieser Test gibt eine Antwort.

GRUNDLAGEN

- Einführung in Intuition, Teil 2 16
- Libraries
– Das 'A' und 'O' des Amiga-Betriebssystems 50
- Künstliche Intelligenz
– Ein kontroverses Thema 78
- CLI
– Der Command Line Interpreter, Teil 2 91
- Sound Sampling
– Klang am Computer 93

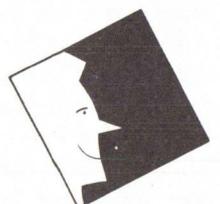
LISTING

- Graph Zeichenmal
– der Funktionsplotter 27
- Übersicht: Zahlenverarbeitung mit Business-Grafik . 58
- 3D-Fractals: Fractale Landschaften 66



Sound Sampling:

Ein ausführlicher Grundlagenartikel führt in dieses interessante Gebiet der Computermusik ein. Passend dazu werden zwei Sound Sampler getestet: Sound Scape & Future Sound.



SOFTWARE

Der Nachbrenner: CLI-Shell	40
True Basic	
– Das 'wahre' Basic	42
Logistix: Ein Text des integrierten Anwenderpakets .	83

True BASIC™

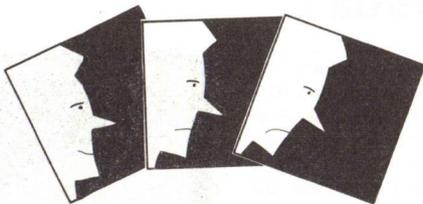
TRUE BASIC

Die Entwickler des 'Urbasics' haben die zeitgemäße Neufassung dieser Sprache vorgestellt. Sie soll zu einem neuen Basic-Standard werden und läuft bereits auf so verschiedenen Rechnern wie IBM und Kompatiblen, MACINTOSH, ATARI ST und jetzt auch auf dem AMIGA.

LOGISTIX

Logistix

Was leistet das integrierte Anwenderpaket auf dem AMIGA. Eignet es sich für den professionellen Einsatz?



TIPS & TRICKS

MS-DOS	
– Zu schnell!!	89

SPIELE

The Feary Tale – The Surgeon – Uninvited	
Deja Vu – Phalanx – Cruncher Factory	
Space Battle – Flip Flop – Demolition	52



Spiele:

Ganz neu: das abenteuerliche 'THE FEARY TALE' und Szenen aus dem Operationssaal in 'THE SURGEON'. Bedienerfreundlich: die Adventurespiele 'DEJA VU' und 'UNINVITED'. Viel Aktion gibt es bei 'PHALANX', 'SPACE BATTLE', 'CRUNCHER FACTORY' und 'DEMOLITION'.



RUBRIKEN

Editorial	3	Einkaufsführer	73
NEWS	6	Leserumfrage	76
Pinboard	36	Inserentenverzeichnis	97
Public Domain Service	38	Impressum	98
Bücher	62	Vorschau	98

NEWS

Wechsel im Commodore -Management

Irving Gould, bisher Vorsitzender des Commodore International Ltd. Aufsichtsrates, wurde mit sofortiger Wirkung zum Chief Executive Officer (CEO) gewählt. Der bisherige Konzernpräsident und CE Officer, Tom Rattigan, hat sein Amt wegen angeblichen Vertragsbruches niedergelegt: Seine Kompetenzen und Zuständigkeiten seien dezimiert worden. Zum neuen Geschäftsführer der amerikanischen Commodore-Gesellschaft wurde Alfred Duncan ernannt. Er war zuvor in führenden Positionen in Kanada und Italien tätig. Neuer Gesamtvertriebsleiter für den amerikanischen Markt ist ab sofort Richard McIntyre, der zuvor die kanadische Vertretung geleitet hatte. Nach diesen Änderungen im Management meint Irving Gould: „Ich bin zuversichtlich, daß wir durch gezielte Maßnahmen die Leistungsfähigkeit des Unternehmens verbessern werden. Es kommt darauf an, in das US-Geschäft neue Impulse und neuen Schwung zu bringen, um so die ungebrochene und weiter zunehmende starke Stellung der europäischen Konzerngesellschaften wirkungsvoll zu ergänzen.“

Commodore weiter im Aufwind

Die deutsche Commodore-Vertriebsgesellschaft kann für das vergangene Quartal die besten Umsatzzahlen seit dem Bestehen der Gesellschaft verkünden: Rund 177 000 Rechner aus dem Hause Commodore wurden verkauft. Geschäftsführer Winfried Hoffmann rechnet mit einer weiteren Steigerung des Umsatzes wegen der ange-

laufenen Produktion und Auslieferung der neuen Amiga-Modelle 500 und 2000. Durch die guten Verkaufszahlen, die Commodore

Deutschland erwirtschaftet hat, wurde auch die positive Entwicklung des Gesamtkonzerns entscheidend unterstützt. Allein seit Januar 1986 hat sich der Wert der Commodore-Aktie mehr als verdoppelt. Umsatzzahlen von Commodore International Ltd. sprechen hier eine deutliche Sprache; im dritten Quartal des laufenden Geschäftsjahres wurden 169,5 Millionen Dollar umgesetzt, wobei der Gewinn sich auf etwa eine Million Dollar ziffert. Im gleichen Zeitraum des Vorjahres hatten die Buchführer noch einen Verlust von 36,7 Millionen Dollar festgestellt.

Neues Spiel mit Supergrafik

Das Softwarehaus Psygnosis, bekannt durch die Sportsimulation Arena, die neuerdings auch in einer Joystickversion erhältlich ist, veröffentlicht demnächst ein neues Spiel mit dem Titel 'Barbarian'. Schon die Demo-Version war ein regelrechter Augenschmaus, denn die phantastischen Animations- und Grafikfähigkeiten des Amiga kommen bei diesem Spiel, das sehr wahrscheinlich ein Psygnosis-Renner werden wird, bestens zur Geltung. Der Spieler schlüpft in die Rolle von „Hegor, dem Barbaren“, der in die Welt Durgan hinauszieht, um den bösen Ty-

rannen Necron zu bezwingen, der das ganze Land in Angst und Schrecken

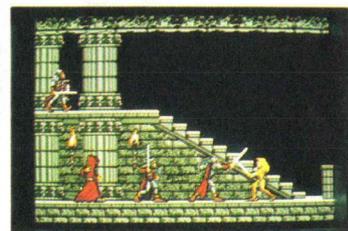
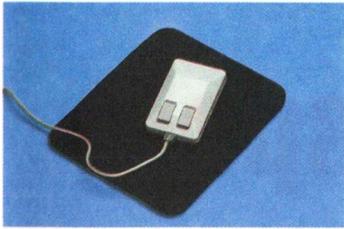


Bild Barbarian

versetzt. Als Belohnung winkt dem tapferen Barbaren die Königskrone. Einen ausführlichen Testbericht werden wir sobald wie möglich in unserem Spieleteil veröffentlichen.



Die Maus im Griff



Wer über mangelnde Haftung der Mausekugel auf der Schreibtischoberfläche klagt, der kann durch eine spezielle Mausunterlage Abhilfe schaffen. Eine 24 x 20 cm große, moosgummi-ähnliche Matte sorgt für ein völlig neues 'Mausgefühl'. Das 'Durchrutschen' der Maus gehört ab sofort der Vergangenheit an. Die geringen Ausmaße der Unterlage reichen zum einwandfreien Arbeiten mit der Maus völlig aus. Erwerben können Sie die Mausunterlage für ca. DM 15,- im Fachhandel.

Down at the Trolls...

...ist ein weiteres Spiel für den Amiga, das voraussichtlich in naher Zukunft erhältlich sein wird. Der legendäre Kampf zwischen Gut und Böse nimmt kein Ende: Unser Held muß diesmal den Zwist zwischen Elfen und Trollen ausbaden, denn die bösen Trolle haben den Schatz der Elfen hinterlistig entwendet. Die Aufgabe besteht darin, den Schatz, der irgendwo im tiefen Reich der Trolle versteckt ist, wieder herbeizuschaffen. Leider erwarten den Helden in den 200 Höhlen morsche Leitern, Steinschläge, tückische Fallen, Feuerbälle und andere widerwärtige Dinge. Ein Magier erweist sich indes als hilfreicher Kumpan im Kampf gegen das Böse; doch wo ist er zu finden?

Als besonderen Leckerbissen weist das Spiel einen Editor auf, mit dem eigene Grotten entworfen und gespeichert werden können.

Go AMIGA Datei

Kurz vor Redaktionsschluß erreichte uns ein Programm, das seit längerer Zeit in verschiedenen Fachzeitschriften angekündigt wurde: Die GO AMIGA Datei. Nach einer Entwicklungszeit von über einem Jahr, so die Entwickler, ist das Programm nun fertiggestellt und ab sofort im Handel. Der erste Eindruck von Leistungsfähigkeit und Bedienerfreundlichkeit ist ausgesprochen gut.

Das Programm kann wahlweise mit Maus oder Tastatur bedient werden. Die einzelnen Features: Editieren der Bildschirmmasken während der Arbeit, vielseitige, komfortable Druckersteuerung von Etiketten bei Listenausdruck, mehrere Bildschirmmasken für ein und dieselbe Datei, frei wählbare Darstellung von Zahlen, internationale Währungen, Mail-Merge-Funktion, Diaschau, Tonschau, Schnittstelle zu anderen Programmen mittels ASCII-Datei sowie Arbeiten mit internem und externem Speicher (RAM bzw. Harddisk).

Die ausgezeichneten Features des Programms konnten zumindest auf ihr Vorhandensein überprüft werden, wobei keinerlei Probleme beim Ausprobieren einzelner Punkte bemerkt wurden. Für 199,- DM ist die GO AMIGA Datei recht preisgünstig; sie wird von der Firma Softwareland, Franklingstr. 27, CH-8050 Zürich hergestellt.

Ein ausführlicher Test des Programms folgt in der nächsten Kickstart-Ausgabe.

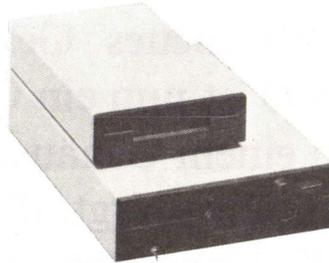
(AK)



Schon die Demo-Version verspricht eine ansprechende Grafik mit vielen gut animierten Sprites. Herausgegeben wird das Spiel von dem Softwarehaus „Rainbow Arts“; der Verkaufspreis war bei Redaktionsschluß noch nicht bekannt.

(AK)

AMIGOS für AMIGA



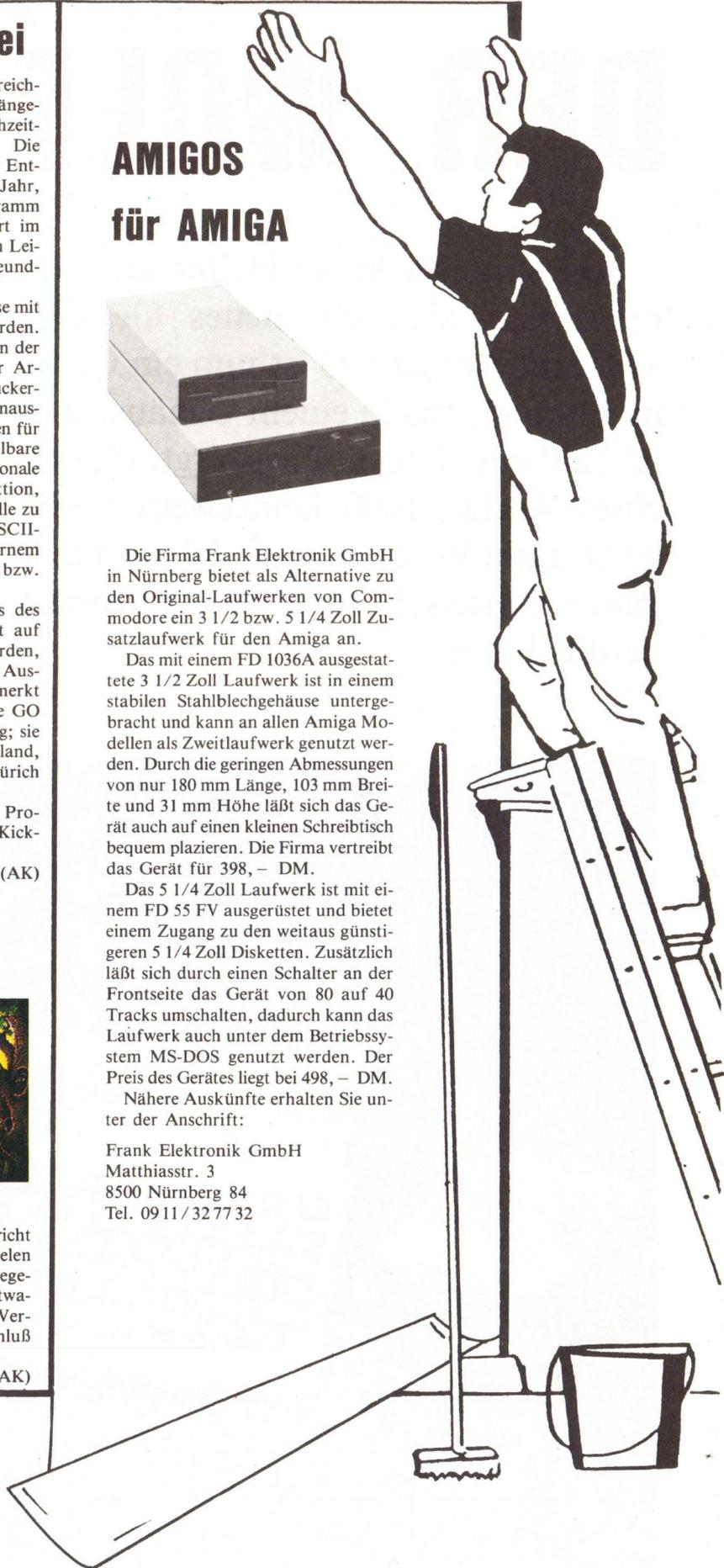
Die Firma Frank Elektronik GmbH in Nürnberg bietet als Alternative zu den Original-Laufwerken von Commodore ein 3 1/2 bzw. 5 1/4 Zoll Zusatzlaufwerk für den Amiga an.

Das mit einem FD 1036A ausgestattete 3 1/2 Zoll Laufwerk ist in einem stabilen Stahlblechgehäuse untergebracht und kann an allen Amiga Modellen als Zweitlaufwerk genutzt werden. Durch die geringen Abmessungen von nur 180 mm Länge, 103 mm Breite und 31 mm Höhe läßt sich das Gerät auch auf einen kleinen Schreibtisch bequem platzieren. Die Firma vertreibt das Gerät für 398,- DM.

Das 5 1/4 Zoll Laufwerk ist mit einem FD 55 FV ausgerüstet und bietet einem Zugang zu den weitaus günstigeren 5 1/4 Zoll Disketten. Zusätzlich läßt sich durch einen Schalter an der Frontseite das Gerät von 80 auf 40 Tracks umschalten, dadurch kann das Laufwerk auch unter dem Betriebssystem MS-DOS genutzt werden. Der Preis des Gerätes liegt bei 498,- DM.

Nähere Auskünfte erhalten Sie unter der Anschrift:

Frank Elektronik GmbH
Matthiasstr. 3
8500 Nürnberg 84
Tel. 09 11 / 32 77 32



DER WOLF im

Schafspelz

Auf dem Markt für Heimcomputer präsentiert sich ein neues Gesicht: Mit dem Amiga 500 ist nun ein Gerät im Handel, das in einem Gehäuse á la 'C 128' eine Leistung verbirgt, die der eines Amiga 1000 keineswegs nachsteht und die vor einigen Jahren noch ganze Schaltschränke mit Hardware gefüllt hätte.



Die Firma Commodore machte sich wohl einige Gedanken über die bescheidenen Verkaufszahlen beim Amiga 1000. Zwei neue Geschwister sollen nun Schwung ins Geschäft bringen: Amiga 500 und 2000. Der 2000er soll ohne Zweifel in den PC Bereich eindringen; er ist ein „offenes“ System, man kann also diverse zusätzliche Hardware (PC-Karte, AT-Karte, Harddisk) mit dem Rechner verbinden. Klar, daß solche Eigenschaften den Preis eines Rechners in die Höhe schnellen lassen. Der neuen Zielgruppe entspricht auch das am professionellen Markt orientierte Design des 2000er-Modells. Der Amiga 500 hingegen zielt, schon durch seinen günstigen Preis von rund 1300 DM, auf den Heimcomputer-Markt. Auch sein Aussehen entspricht weniger einem Profi-Rechner als dem eines Heimcomputers. Die Kompatibilität zwischen den einzelnen Amiga-Modellen bleibt indes gewährleistet; nur die Grundausstattung unterscheidet

die einzelnen Rechner voneinander.

Was gibt's fürs Geld?

Der günstige Preis des Amiga 500 macht ihn auch für Leute erschwinglich, die kein unbegrenztes Bankkonto zur Verfügung haben. Dennoch kann der Junior in der Amiga-Familie mit Leistungsdaten aufwarten, die manchen weit teureren Rechner vor Neid erblassen lassen. In der Grundausstattung besitzt er 512 KByte RAM sowie ein eingebautes 3 1/2 Zoll-Disketten-Laufwerk mit einer Speicherkapazität von 880 KByte. Das Betriebssystem (Kickstart 1.2) befindet sich fest in ROMs und muß nicht mehr, wie beim Amiga 1000, nach Einschalten des Rechners geladen werden. Trotzdem hat die Sache einen kleinen Haken: Die Flexibilität des Betriebssystems wird enorm vermindert, denn eventuell erscheinende Updates des Kickstart können nicht verwendet werden.

Die Softwareproduzenten sind eifrig damit beschäftigt, ihre Programme dem neuen Betriebssystem Kickstart 1.2 anzupassen. Denn unglücklicherweise liefen einige Programme unter der Version 1.2 überhaupt nicht oder nur mit erheblichen Schwierigkeiten. Doch dieses Manko der neuen Systemsoftware wird sicher bald behoben sein, so daß alle Programme einwandfrei ablaufen werden.

Neben der Rechner-Grundeinheit bekommt man für sein Geld ein separates Netzteil; das vom Amiga 1000 bekannte Gebläsegräusch des Kühlventilators entfällt also. Am Netzteil des A 500 findet sich auch der Ein- und Ausschalter; der Anwender muß es also an einem gut zugänglichen Ort platzieren. Eine Maus befindet sich natürlich ebenso im Lieferumfang wie derzeit zwei Disketten, auf denen sich die Workbench und der ABASIC-Interpreter befinden. Geplant ist auch ein „Tutor“ für den 500er: Ein Lernprogramm, das sich

Der im Gegensatz zum ST realexistierende BLITTER, die DMA Logik und der TROPPE sind in diesem Chip enthalten

Portbaustein 8520

„Paula“ besorgt das Handling von Interrupts, den Laufwerken und der RS 232 Schnittstelle

Portbaustein 8520

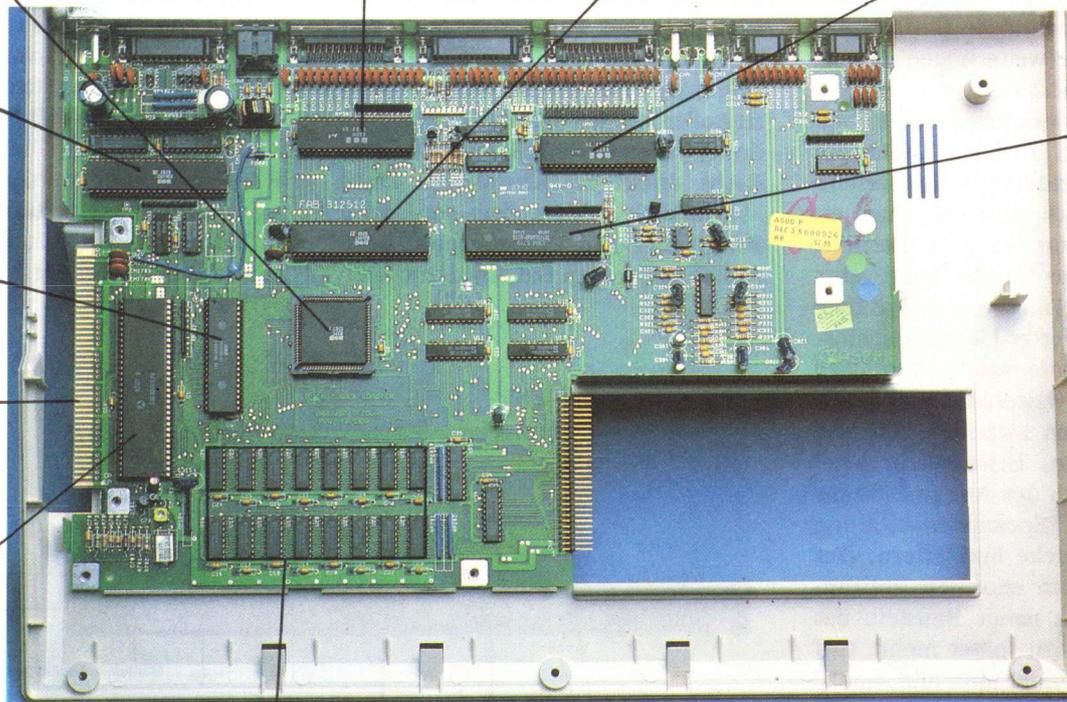
Denise der Grafikmanager des Amigas

Das Kickstart Rom In den 256 KBYTE findet die neue Version 1.2 platz

Am Expansions port steht der Amigabus externen Erweiterungen zur Verfügung

Die altbekannte 68 000 CPU

Das hoch integrierte Gate Array „Gary“ enthält die im Amiga benötigten Logik funktionen



Die 512 KBYTE der Grundversion sind in diesen 16 Chips untergebracht

mit der Arbeitsweise des Gerätes beschäftigt und rasche Erklärungen liefert. Außerdem erhält der Käufer ein in englischer Sprache verfaßtes Buch über den generellen Gebrauch des Amiga, ein AmigaDOS-Handbuch in Deutsch (Erklärung des Command-Line-Interpreters und des Editors) und ein ebenfalls deutsches Werk über den mitgelieferten Amiga BASIC-Interpreter.

Die deutsche Tastatur (deutsche Sonderzeichen) des A 500 hat sich gegenüber dem A 1000 stark verändert; abgesetzte Tasten für die Cursorsteuerung und den Zehnerblock sind nur zwei Unterschiede zum 'alten' A1000.

Das eingebaute 3 1/2 Zoll-Laufwerk ist von einem Unwissenden kaum zu finden. Nur Auswurfknopf und Diskettenschacht veraten die Existenz des Drives an der rechten Seite des Gerätes. Links befindet sich der Expansionsport, der durch eine Klappe verdeckt ist. Entfernt man sie, kommt ein 86poliger Platinenstecker zutage, der von aufwendiger Hardware oder Peripherie genutzt werden kann. Die Ingenieure des A 500 haben jedoch etwas Paradoxes entwickelt: Der Port wurde gegenüber dem A 1000 um 180 Grad gedreht. Durch diesen unsinnigen Einfall können Hardwareerweiterungen des A 1000, wenn überhaupt, nur mit erheblichen Schwierigkeiten am A 500 genutzt werden. Auf der Unterseite des Rechners findet man eine weitere Klappe, die nach dem Entfernen einen Anschluß zur Hauptplatine freigibt. An dieser Stelle kann eine Speichererweiterung von 512 KByte mit akkugepufferter Uhr angeschlossen werden. Diese von Commodore vertriebene Erweiterung soll nach letzten Berichten zwischen 200 und 250 DM kosten. Erstaunlich: Muß doch ein Besitzer des Amiga 1000 für genau dieselbe Speichererweiterung etwa das Vierfache hinblättern. Der Grund für diesen enormen Preisunterschied ist ein neuer Baustein des 500ers, doch dazu später mehr. Auf der Rückseite findet man alle wichtigen Anschlüsse, die ein Computer heutzutage so braucht: Eine Video-signal-Buchse, der RGB-Monitor- und Netzteil-Anschluß, parallele und serielle Schnittstelle für Drucker oder Modem, zwei Tonausgänge für die

Stereoanlage oder den Monitor und, last not least, noch zwei Anschlüsse für Maus, Joystick oder Ähnliches. Auch die Amiga-Entwickler haben aus der Vergangenheit gelernt: Der serielle und parallele Port besitzen nun PC-Standard; die unsinnige Vertauschung von männlichen und weiblichen Steckern sowie die Belegung der Pins wurden geändert.

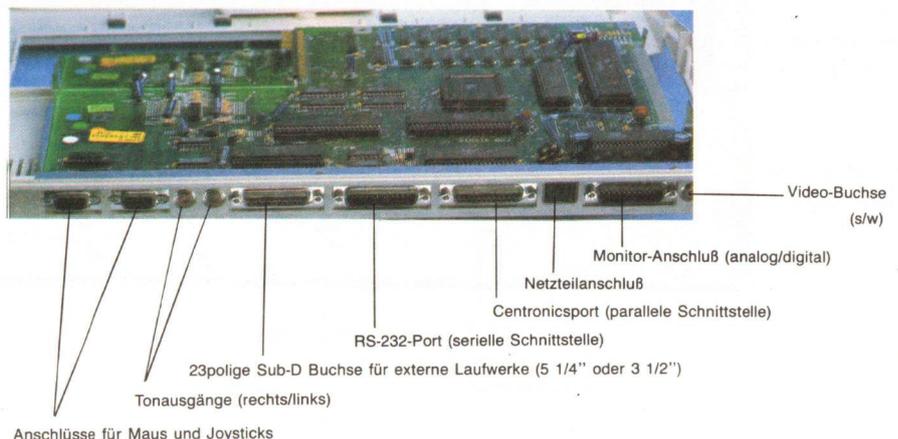
Das Innenleben des A500

Um an das Innenleben des Amiga heranzukommen, muß man nach dem Entfernen des Plastikgehäuses noch eine Metallabdeckung lösen. Nach dem Beseitigen des Gehäuses und der Metallabdeckung erscheint eine sauber aufgeräumte Platine. Ein zu neugieriger Amiga-500-Besitzer sollte jedoch den Garantieverlust nach Öffnen des Gerätes bedenken. Ein quadratischer Baustein fällt sofort ins Auge, der sogenannte Fat-Agnus. Diesen Custom-Chip konnte man auch nach dem vollständigen Zerlegen des A 1000 nicht finden. Der neue Baustein ist für jegliche Art von Speicherzugriff zuständig; zusätzlich vereinigt er die Coprozessoren Copper und Blitter. Durch die Neuentwicklung des Fat-Agnus (aus drei mach einen) wurde enorm viel Platz eingespart. Der geniale Chip beinhaltet außerdem noch einige Logikfunktionen, die es ermöglichen, 1 MByte RAM direkt zu adressieren. Diese Möglichkeit erklärt den geringeren Preis für eine Erweiterung beim A 500: Die teuren Bausteine zur Spei-

cheradressierung sind entbehrlich. Will man jedoch einen Rechner mit mehr als 1 MByte Speicher, dann muß man, wie beim A 1000, einen größeren Betrag über den Ladentisch wandern lassen, da in diesem Fall weitere Logikbausteine unbedingt gebraucht werden. Durch die Verwendung des Fat-Agnus entsteht übrigens noch ein weiterer Vorteil: Der Baustein arbeitet schneller als die ähnlichen im A 1000, doch die Geschwindigkeitssteigerung erreicht keine höheren Dimensionen und fällt kaum ins Gewicht.

Den bulligen 68000-Prozessor findet der Betrachter auf der linken Seite der Platine, direkt neben dem Expansionsport. Es hat den Anschein, als ob das Herzstück des Rechners nichts mit dem 'Rest der Welt' zu tun haben wolle, so 'deplaziert' und unbekümmert liegt er da. Einen weiteren Beitrag zur Bausteinrationalisierung leistet der neu entwickelte Gate-Array-Chip, kurz Gary genannt. Er beinhaltet eine ganze Reihe von zuvor einzelnen Logikbausteinen (A 1000) in einem einzigen 'Tausendfüßler' und schafft somit Platz auf der Platine.

Ein weiterer Chip ist neu auf der A 500-Platine. Er wurde anstelle von acht RAM Bausteinen (256 KByte) eingesetzt, die zuvor für das Kickstart-Betriebssystem reserviert waren. Da beim A 500 das Laden von Kickstart entfällt, können diese RAMs eingespart werden. Dieser einzige Baustein beinhaltet das komplette Betriebssystem 1.2, also sind 256 KByte darin vereinigt.



Die Chips mit den Bezeichnungen Denise und Paula wurden direkt vom A 1000 zum 500er übernommen. Denise kümmert sich immer noch um Grafik, Sprites und Farben; Paula ist nach wie vor mit Diskettenoperationen, Tonausgabe, Interrupts und dem seriellen Port beschäftigt. Hier blieb alles beim Alten.

Kompatibel?

Trotz einiger Änderungen des A 500 gegenüber dem A 1000 bleiben die Rechner untereinander voll kompatibel, das gilt natürlich auch für den A 2000. Die technischen Daten sind zum größten Teil identisch mit denen des A 1000:

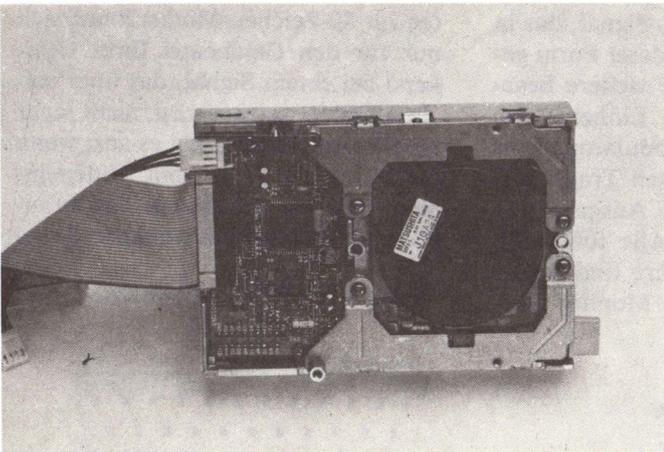
Motorola MC68000 Prozessor, 7.16 MHz Taktfrequenz, maximale Farb-

anzahl von 4096, Grafikauflösungen von 320★256 Bildpunkten in 32 bzw. 4096 Farben, 320★512 ebenfalls in 32 und 4096 Farben, 640★256 in 16 Farben sowie die Auflösung von 640★512 in 16 Farben, 512 KByte RAM, intern erweiterbar auf 1 MByte RAM, extern auf 8 MByte, acht Hardware-Sprites, parallele und serielle Schnittstelle, zwei Anschlüsse für Maus, Joystick, Paddel oder Lightpen, Anschlüsse für RGB-Monitor, Diskettenlaufwerk, Videomonitor und Netzteil, je ein Stereoausgang und ein Systembus, internes 3 1/2-Zoll-Diskettenlaufwerk mit 880 KByte.

Fazit

Für rund 1300 DM bietet der Amiga 500 ein hervorragendes Preis-Leistungsverhältnis. Trotz einiger Mängel (Ein-/Ausschalter am Netzteil) ist das Konzept gelungen. Auf jeden Fall hat der 'kleine' Amiga das Zeug zum großen Einstieg speziell in den Heimcomputerbereich. Auch die Softwareindustrie hat erkannt, daß der 500er wohl ein Renner werden wird und entwickelt entsprechend fleißig. Auf diesem Sektor ist in Zukunft einiges zu erwarten. Bleibt zu hoffen, daß nicht Quantität, sondern Qualität ausschlaggebend sein wird.

(AK)



	<p>AMIGA</p>	
	Lattice C-Compiler neu V3.1	448,-
	MCC-Pascal Metacomco	248,-
	AZTEC C-Compiler AM-p V3.40	448,-
	AZTEC C-Compiler AM-d V3.40	648,-
	AZTEC C-Compiler AM-c V3.40	1148,-
	AC/Fortran77 Abasoft	690,-
	AC/Basic Compiler Abasoft	398,-
	Cambridge LISP Metacomco	490,-
	Modula II Standard V2.0	298,-
	Modula II Developer V2.0	448,-
	Modula II Commercial V2.0	848,-
	MCC Assembler Metacomco neu	198,-
	K-SEKA Assembler Kuma	148,-
	True Basic Interpreter	398,-
TOOLKIT z. B. Pipe, Enlarge	118,-	
SHELL für Kommandointerpret.	148,-	
Online! Terminalprogramm	178,-	
GIZMOS Utilityprogramm	128,-	
UBM-Text Textverarb. dt. V2.2	248,-	
Scribble II Textverarb. engl.	228,-	
Pro Write Text + Grafik	378,-	
Page Setter Desktopprogramm	378,-	
VIP Professional englisch	348,-	
Logistix Tabellenkalk. deutsch	448,-	
Analyse Tabellenkalk. V2.0	228,-	
Superbase Dateiverwaltung	248,-	
dBMAN Dateiverwaltung	398,-	
Deluxe Paint Grafikprogramm	198,-	
Deluxe Paint II Grafikprogramm	278,-	
Deluxe Print (Print Shop)	198,-	
Deluxe Video Construction	198,-	
Deluxe Video Construction II Set	278,-	
Instant Music Kompositionspr.	79,-	
Sonix Musikprogramm V1.4	198,-	
Aegis Animator / Images	348,-	
NEWIO Leiterplattenflecht.	498,-	
Aegis Draw plus CAD-Programm	578,-	
Dynamic CAD ähnlich Auto-CAD	1098,-	
Deja Vu Grafisches Krimispiel	89,-	
Sinbad Abenteuerspiel	99,-	
Defender of the Crown	89,-	
Marble Madness Geschicklichk.	69,-	
Pawn Text + Grafik Adventure	69,-	
Starglider Geschicklichk.	69,-	
Quiwi Quizspiel	69,-	
Archon Geschicklichkeitsspiel	79,-	
Archon II Adventure Spiel	79,-	
The guild of thieves	69,-	
Barbarian Abenteuerspiel	69,-	
Terror polds Strategiespiel	59,-	
Faery tale Abenteuerspiel	119,-	
One-on-One Basketballspiel	79,-	
Deep Space Weltraumabenteurer	89,-	
Uninvited Krimispiel	79,-	
Portal Science Fiction	99,-	
Chessmaster 2000 Schachspiel	99,-	
Wishbringer Infocom Textadv.	79,-	
Bard's Tale Adventure	119,-	
Flight II Sublogic	119,-	
Amiga Users Guide Buch	58,-	
Digi View Digitizer	448,-	
Druckerkabel Centronics	39,-	
Akustikkoppler Dataphon s23	328,-	
Speichererweit. 256KB 120nS	128,-	
Speichererweiter. 768 KB-RAM	498,-	
Speichererw. 1 MB extern	898,-	
Diskettenlaufwerk 3'5 880KB	448,-	
Doppellaufwerk 3'5 2★880KB	848,-	
Disketten. 5 1/4 40/80 Spur	598,-	
Harddisk 20MB f. Amiga 1000	2498,-	
10 Disketten 3'5 2DD in Box	49,-	

Preisliste mit Info anfordern.
Händler fordern Händlerliste an.

PHILGERMA GmbH
Ungererstraße 42 · 8000 München 40
Tel. 0 89 / 39 55 51 von 10.00 – 18.30 Uhr

Besuchen Sie unseren Softwareladen in der Ungererstraße 19. Sie können alle Produkte anschauen und testen.

Kontaktfreudig:

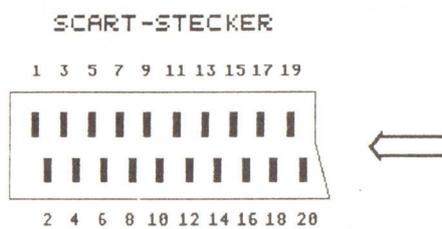


...ist der Amiga, wenn es gilt, sein Wirken optisch ins rechte Licht zu rücken. Er stellt dafür eine stattliche Anzahl von Anschlüssen zur Verfügung, an die mehrere Bildschirme angeschlossen werden können. Diese sollen etwas genauer unter die Lupe genommen werden. Außerdem erfahren Sie, wie man einen Fernseher als Monitor benutzen kann.

Wer sich seinen Amiga einmal von hinten anschaut, wird auf der rechten Hälfte der Rückwand drei Anschlüsse finden. Alle liefern Bildinformationen nach außen, die Art der Signale ist jedoch stark unterschiedlich. Ganz rechts sieht man eine Cinch-Buchse mit der Bezeichnung „Video“, wodurch das Signal schon recht gut klassifiziert ist. Es hat nämlich die gleiche Form, die von einem Videorecorder zur Bildverarbeitung benutzt wird. Daneben sitzt eine 8-polige DIN-Buchse. Auch hier liegt das Videosignal (auch FBAS genannt) an, aber durch die größere Kontaktzahl ist es möglich, die Ton-Information ebenfalls über diesen Steckverbinder zu leiten. Beim Video-Anschluß muß dazu eine eigene Verbindung benutzt werden. Dritter im Bunde ist ein 23-poliger Sub-D-Stecker. An ihm findet man zwei andere Bildinformationsformen: Digital-RGB und Analog-RGB. RGB steht für Rot-Grün-Blau, und wie die Farben aufgeführt sind, werden sie auch übertragen: nämlich getrennt. Das schlägt sich in einer wesentlich besseren Weiterleitung der Bildinhalte nieder, da die Signale keine Möglichkeit haben, sich im Kabel gegenseitig zu beeinflussen. Wenn alle Informationen auf eine Ader übertragen werden, wird das Signal so breitbandig, daß Kabelkapazitäten schon anfangen, die Information zu beschneiden. Im Prinzip unterscheiden sich Digital- und Analog-RGB durch die Anzahl der darstellbaren Farben. Bei Analog-RGB ist die Anzahl der Farben theoretisch unbegrenzt, Digital-RGB kann dagegen nur eine begrenzte Anzahl (8, mit einem Intensitätsbit 16) von Farben darstellen. Aus dem

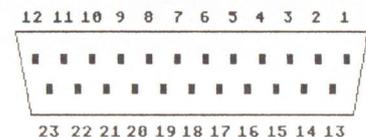
RGB-Signal kann man sehr einfach ein Video-Signal (leider nur monochrom) mischen, in dem man R-, G-, B- und Comp-Sync-Spannungen über Widerstände zusammenschließt. Aber wozu hat schließlich der Amiga einen farbigen Video-Ausgang?

Der Original-Monitor von Commodore wird mit einem Kabel ausgeliefert, das an diesen 23-poligen Stecker angeschlossen wird und die Bilder als RGB-Signale überträgt. Zusätzlich besitzt der Monitor noch einen Video-Eingang. Im Inneren des Monitors wird das Videosignal aufgetrennt und liegt dann ebenfalls als Analog-RGB vor. Das selbe Vorgehen findet man auch im Fernsehgerät, nur durchläuft das Signal, das ja von der Antenne in anderer Form geliefert wird, noch eine weitere Bearbeitungsstufe. Diese Einheit, der Tuner und HF-Demodulator, sendet den hochfrequenten Träger aus und stellt an ihrem Ausgang ein FBAS-Signal bereit. Alle folgenden Stufen bis zur Bildröhre findet man im Fernseher und im Monitor gleichermaßen.

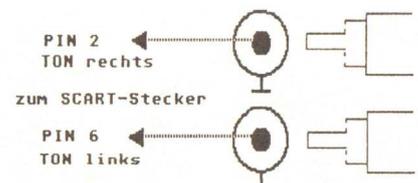


passen würde! Wer einiges technisches Geschick hat und das Wort „LötKolben“ schon einmal gehört hat, kann sich für eine 'Handvoll Dollar' leicht ein passendes Kabel fertigen. Zuerst muß jedoch geklärt werden, welche Anschlüsse der Fernseher überhaupt anbietet. Im einfachsten Fall, der für uns am ungünstigsten ist, besitzt die „Kiste“ nur einen Antennen-Anschluß. Dann muß ein HF-Modulator angeschafft werden, der Videosignale in die Form bringt, in der sie auch von der Antenne angeboten werden. Die Anschaffung eines solchen Gerätes sollte man sich gut überlegen, da ein befriedigende Darstellung von Text (Arbeiten im 80-Zeichen-Modus lohnt sich nur für den Geldbeutel Ihres Optikers) bei einem Signal, das über solche Umwege geleitet wird, nicht mehr möglich ist. Besser sieht es aus, wenn Sie einen Fernseher mit Videoeingang besitzen. In diesem Fall brauchen Sie lediglich das passende Kabel

AMIGA Monitorstecker



AMIGA Audio Ausgang



Die Glotze mißbraucht

Wenn man sich Überlegt, daß ein Monitor eigentlich nur ein abgemagertes Fernsehgerät ist, könnte man auf die Idee kommen, einen TV-Apparat als Monitor einzusetzen. Doch vor das Sehen hat die Industrie die Anschlüsse und Steckverbinder gesetzt. Prinzipiell ist alles kein Problem; wenn nur der 8-polige DIN-Stecker irgendwo in den blöden Fernseher

zur Verbindung des TV-Geräts mit dem Amiga. Da der Amiga mit allen Videoanschlüssen ausgestattet ist, dürften hier die wenigsten Probleme auftreten. Wenn der Fernseher einen 8-poligen DIN-Anschluss oder eine Cinch-Buchse hat, kann man ein fertiges Kabel zu einem meist annehmbaren Preis kaufen. Im Falle, daß der Fernseher eine BNC-Buchse (sieht so



AUTOREN GESUCHT

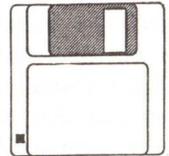
Buch

 **Amiga**



+

Programm



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

Schreiben Sie uns

Heim-Verlag

Kennwort: Autor
Heidelberger Landstr. 194
6100 Da.-Eberstadt
Tel.: 061 51/560 57

Jetzt hat Qualität einen **Preis**

Amiga 500 und
Amiga 2000
sofort lieferbar

Amiga 500- & 1000-Hardware

Kickstart EPROM-Modul V 1.2

Als Einbauversion in der Golem-Speicherbox

179,- DM

oder

externes Modul in separatem Gehäuse 199,- DM

Alle Versionen abschaltbar, so daß andere Kickstart-Versionen wieder gebootet werden können.
Betriebskontrollanzeige durch LED, Systembus ist durchgeführt.

3,5-Zoll-Amiga-Laufwerk

hochwertiges NEC 1036a im Metallgehäuse, 880-KB-Interface für 3. Laufwerk vorbereitet

3,5-Zoll-Doppellaufwerk

technische Daten wie Einzellaufwerk

5,25-Zoll-Amiga-Laufwerk

40/80-Track-Umschaltung, 880-KB-Speicherkapazität, in farblich passendem Metallgehäuse.
Nutzen Sie Ihre alten 64'er-Disketten mit dem 5,25"-Laufwerk als externes Amiga-Laufwerk.

ext. RAM-Board

Leistungsmerkmale

autokonfigurierend, abschaltbar, erweiterbar durch Sandwichtechnik, kombinierbar mit anderer Peripherie, Betriebskontrollanzeige durch LED, Leistungsaufnahme unter 300 mA, superschnell durch perfektes Timing, Zusatzslot für Clock- oder Kickstart-Modul, Busdurchführung

1 MB 898,- DM,

2 MB 1198,- DM

369,- DM

699,- DM

479,- DM

Kupke Computertechnik GmbH · Apelank 28 · 4600 Dortmund · Telefon 02 31 / 85 26 05

ähnlich aus wie Cinch-Steckverbinder, besitzt aber einen Bajonetting zur Arretierung) eingebaut hat, lötet man sich das Kabel am besten selbst, da solche „Sonderwünsche“ beim Fachhandel oft mit gewaltigen Preisaufschlägen belohnt werden. 2 – 3 Meter abgeschirmtes, einadriges Kabel und je ein Cinch- bzw. BNC-Stecker werden benötigt, um dieses Kabel herzustellen. Die Abschirmung wird jeweils mit dem äußeren Kontakt der Stecker verbunden und die Ader des Kabels mit den Stiften in der Mitte verlötet. Danach dürfte der Nutzung des Fernsehers als Datensichtgerät nichts mehr im Wege stehen. Aber auch bei dieser Anschlußart wird man noch Abstriche an der Bildqualität in Kauf nehmen müssen. Die 80-Zeichen-Schrift hat stark mit Farbbrändern zu kämpfen, aber bei 60 Zeichen pro Zeile und bei Grafikdarstellung ist das Bild ganz annehmbar. Sozusagen „Schwarzsehen erster Klas-

putermonitoren. Der Fernseher wird in diesem Fall über einen RGB-Eingang angesteuert. Wer jetzt sein Kabel vom Amiga-Monitor auf den Fernseher umsteckt, wird verwundert

seits die Composite-Sync-Spannung (Mischung aus den Spannungen für horizontale und vertikale Synchronisation) für das RGB-Signal bereitzustellen. Die Synchron-Information wird

SCART-BUCHSE

PIN	
1	Ton Ausgang rechts
2	Ton Eingang "
3	Ton Ausgang links
4	Ton Masse
5	Blau Masse
6	Ton Eingang links
7	Blau Eingang
8	Schaltsp Tuner/AV
9	Grün Masse
10	Frei
11	Grün Eingang
12	Frei
13	Rot Masse
14	Frei
15	Rot Eingang
16	RGB-Schaltspannung
17	Video Masse
18	RGB-Schaltsp. Masse
19	Video Ausgang(FBAS)
20	Composite Video/Comp. Sync Ein

2<----->	Cinch r.
4<----->	Cinch Masse
6<----->	Cinch l.
7<----->	5
11<----->	4
15<----->	3
16<----->	23
20<----->	10
5,9,13,17,18<----->	13,16..20

Amiga RGB-Monitoranschluß

PIN	
1	External Clock
2	" " Enable
3	Analog Rot
4	" Grün
5	" Blau
6	Digital Intensität
7	Digital Blau
8	" Grün
9	" Rot
10	Comp. Sync.
11	Horizontale Sync.
12	Vertikale "
13	Masse
14	Zero Detect
15	Takt
16	Masse
17	"
18	"
19	"
20	"
21	-12 Volt (max. 50mA)
22	12 Volt (" 100mA)
23	5 Volt (" 100mA)



Für ca. 30,- DM kann das Kabel selbst hergestellt werden.

se“ kann, wer im Besitz eines neueren Fernsehgerätes ist, das mit einer sogenannten SCART-Buchse (nach dem französischen Fachverband der Hersteller von Unterhaltungselektronik – von manchen Fernsehherstellern wird die Steckverbindung auch Euro-AV-Buchse genannt) ausgestattet wurde (Abb. 1). Mit dem passenden Kabel erreicht man eine Bildqualität, die der des Amiga 1081 ebenbürtig ist. Das wird dadurch möglich, daß beim Anschluß über die SCART-Buchse der gleiche Signalweg begangen wird wie bei den „echten“ Com-

putermonitoren. Der Fernseher wird vor einem weißen Bildschirm sitzen. Keine Angst, weder der Rechner noch die heißgeliebte Flimmerkiste ist defekt. Der Übeltäter ist immer das Kabel, denn das ist erbarmungslos auf den 1081 ausgelegt. Darum fehlen einfach alle Leitungen, die der Monitor nicht benötigt. Unter ihnen ist auch die RGB-Schaltspannung. Mancher Leser wird sich fragen, wozu diese Spannung dient. In der Norm für den SCART-Stecker wird der Pin 20 zweifach benutzt, um einerseits ein komplettes Video-Signal (FBAS) zu übertragen und anderer-

also immer an Pin 20 bereitgestellt. Welche Bildinformationen der Fernseher auswerten soll, wird ihm über die Schaltspannung mitgeteilt. Die besagte Schaltspannung wird an Pin 16 verlangt. Das entspricht zwar nicht der Norm, die hier das Austastsignal (das zur Strahldunkelsteuerung beim Zeilenrücklauf dient) anlegt, wird aber von allen mir bekannten TV-Geräten so verlangt. Trotzdem sollte der Nachbauwillige sich genauestens vergewissern, ob diese Anschlußbelegung auch bei ihm zutrifft, da eine falsche Spannung am falschen Anschluß fatale Folgen haben kann. Insgesamt benötigt man also mindestens 6 Adern, um den Fernseh Bildschirm zum Strahlen zu bringen: R, G, B, Composite Sync, RGB-Schaltspannung und natürlich Masse. Die RGB-Schaltspannung wird dabei von Pin 23 (+5 Volt) an der Sub-D-Buchse abgenommen. Wenn man die entsprechenden Pole am Amiga mit denen am SCART-Anschluß des Fernsehers verbindet, dürfte der Zusammenarbeit von Amiga und Fernseher nichts mehr im Wege stehen. Eine genaue Anschlussbelegung der Scartbuchse und des 23-poligen Sub-D-Steckers finden Sie in

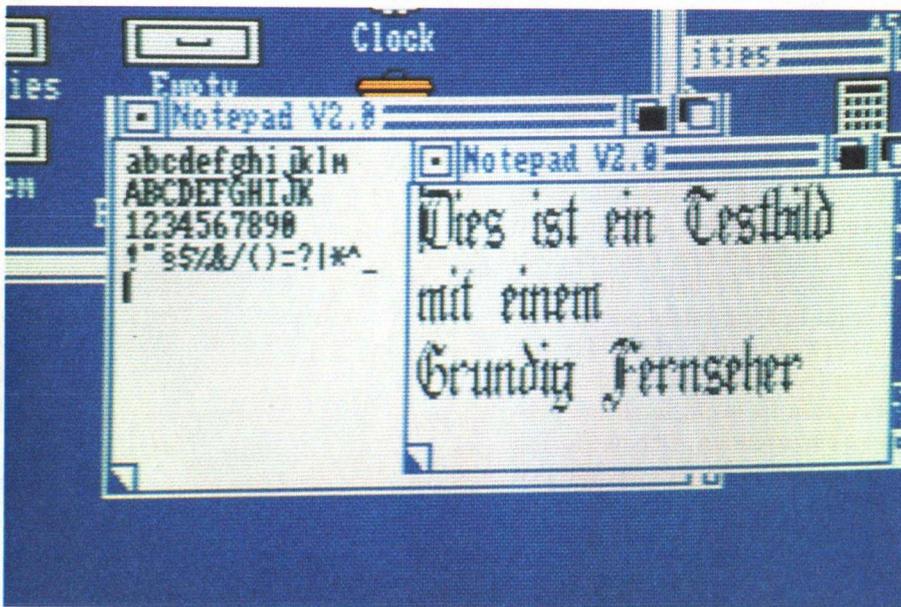
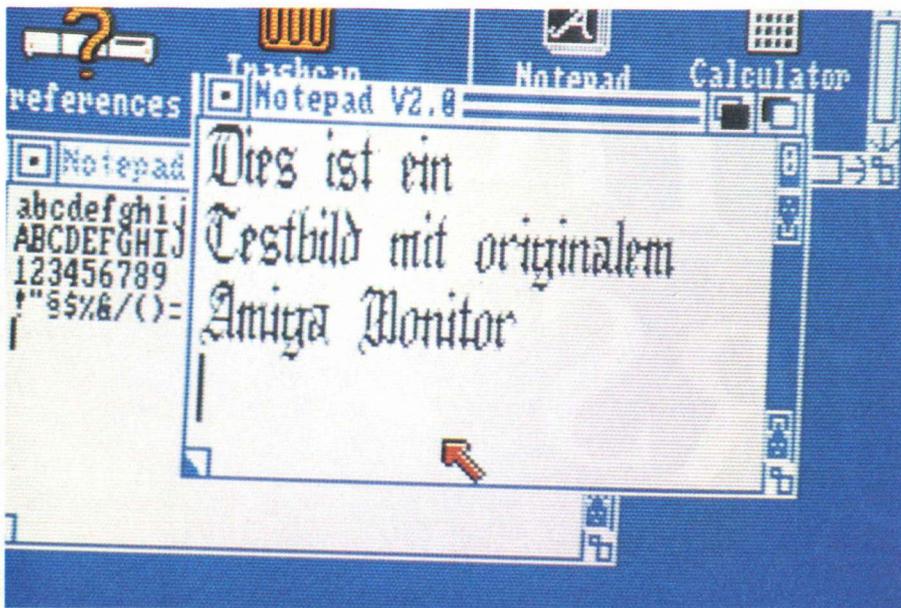


Abb 1. Dort sind auch alle notwendigen Verbindungen explizit aufgeführt. Qualitativ wird man von den Bildern, die ein Fernseher bei Benutzung des SCART-Einganges liefert, überrascht sein (vor allem die C64-Kenner, die sich noch an dessen Bilder auf der Glotze erinnern). Ein 68-cm-Sony lieferte ein Bild, das dem 1081er in keiner Weise nachstand. Auch ein 42-cm-Grundig lieferte ähnliche Ergebnisse. Diese Geräte stellen jedoch keine Vorauswahl dar, andere Fernseher dürften ebenso gute Ergebnisse liefern. Ein besonderes Plus bieten die großen TV-Geräte. Spiele und Grafikanwendungen wirken auf

dem großen Bildschirm wesentlich imposanter. So erhält das Wort „Cinemaware“ eine neue Bedeutung. Für einen Test ist die Startsequenz von „Halley-Projekt“ zu empfehlen. Wer einen Amiga kaufen will, sollte sich also überlegen, ob ein kleiner Fernseher mit SCART-Buchse nicht die bessere Wahl gegenüber einem Monitor ist. Und falls der Kauf eines neuen Fernsehers ansteht, sollte man grundsätzlich auf einen SCART-Anschluss achten.

chk

Digi-View V2.0	399,-	DM
Marble Madness	58,95	DM
The Pawn	55,-	DM
Deep Space	69,95	DM
Archon II	59,95	DM
Seven Cities of Gold	59,95	DM
Programmers Handbook	45,-	DM
Rom-Kernel Ref. Exec	57,95	DM
3,5"-Disk. DS/DD	3,09	DM
Public Domain	ab 5,50	DM
MS-DOS-SOFTWARE AUF ANFRAGE		
Cambridge Lisp	299,-	DM
MCC-Pascal	175,95	DM
MCC-Assembler	149,-	DM
Lattice C V3.1	350,-	DM
MCC-Toolkit	77,95	DM
3,5"-Amigos-Laufwerk DM mit Metallgehäuse	369,-	DM
2-MB-Speichererw.	1098,-	DM
5,25"-Laufwerk	479,-	DM
Future-Sound-Digi.	399,-	DM

Kostenlose Prospekte gibt's bei...

CWTC

Computerversand CWTG
 Joachim Tiede
 Bergstraße 13 · 7109 Roigheim
 ☎ 0 62 98 / 30 98 von 17-19 Uhr

Anzeigenschluß
 für

AMIGA-KickStart

9/87

ist der 24.07.87

Heim-VERLAG

Heidelberger
 Landstraße 194

6100 Darmstadt-Eberstadt
 Telefon 061 51/5 60 57

AB-COMPUTERSYSTEME

AMIGA® ATARI®

A. Büdenbender · 5 Köln 41 · Wildenburgstr. 21
 ☎ 02 21 / 430 14 42

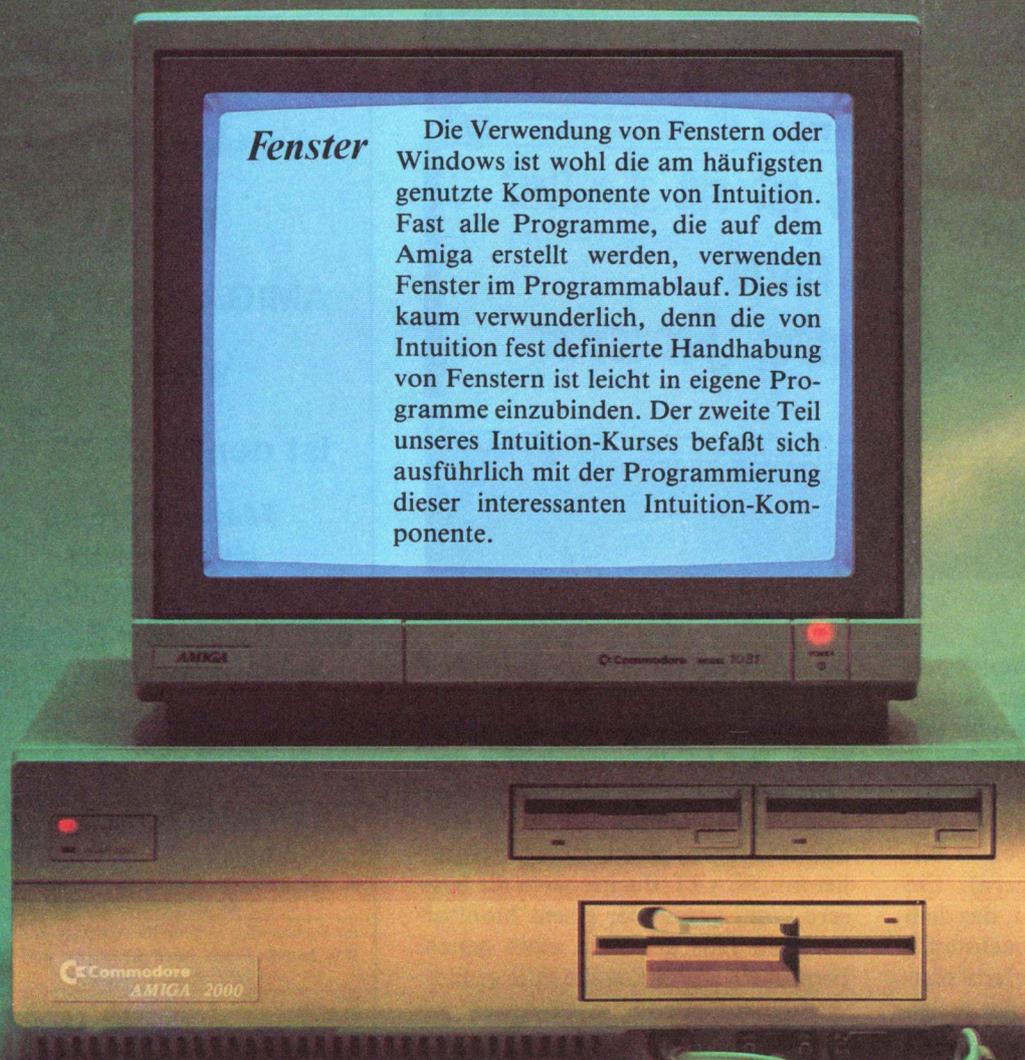
Ihr Fachhändler in Köln für AMIGA/ATARI/PC
 Wir bieten Ihnen noch Beratung und Service
 für Ihren Computer

NEC Laufwerke einzeln FD 1036a slim Line für Amiga/ST modif. +10 DM	249,-
AMIGA LAUFWERK 3,5 Zoll anschlussfertig Metallgeh. einzeln abschaltbar	349,-
AMIGA hardware Uhr für Amiga 1000 läuft auch ohne Strom weiter steckbar	99,-
AMIGA 500 COMPUTER NEU mit Laufwerk anschlussfertig o. Monitor	1148,-
AMIGA 2000 sofort lieferbar o. Monitor 1 Laufwerke Deutsche Vers.	2598,-
NEC P6 Drucker 24 Nadeln St/Amiga deutsche Version 12 Mon. Garantie	1198,-

INTUITION- KURS 2

Fenster

Die Verwendung von Fenstern oder Windows ist wohl die am häufigsten genutzte Komponente von Intuition. Fast alle Programme, die auf dem Amiga erstellt werden, verwenden Fenster im Programmablauf. Dies ist kaum verwunderlich, denn die von Intuition fest definierte Handhabung von Fenstern ist leicht in eigene Programme einzubinden. Der zweite Teil unseres Intuition-Kurses befaßt sich ausführlich mit der Programmierung dieser interessanten Intuition-Komponente.



Rückblick

Mit einem kleinem Rückblick auf den ersten Teil des Intuition-Kurses möchte ich beginnen. Viele Leser, die vielleicht die erste Ausgabe dieser Zeitschrift noch nicht erwerben konnten, wissen mit dem Begriff Intuition wenig anzufangen. Leser, die den ersten Teil des Kurses mitverfolgt haben, können diesen Abschnitt ohne weiteres überspringen.

Intuition ist eine Library oder Bibliothek (eine Library ist eine Sammlung von kleinen Programmroutinen) des Amiga-Betriebssystems. Sie befaßt sich mit der grafikorientierten Benutzeroberfläche des Amiga, der Workbench oder vielmehr mit deren Komponenten. Intuition-Komponenten sind unter anderem Bildschirme (Screens), Gadgets, Requester oder Fenster (Windows) – um nur einige zu nennen. Die verschiedenen Elemente werden im Laufe des Kurses behandelt und näher erläutert. Die Handhabung und Erstellung von Fenstern möchte ich im folgenden Kursteil von Intuition aufgreifen.

Intuition, eine Library des Amiga-Betriebssystems, beinhaltet einen großen Teil des 'Funktionsapparates' der grafikorientierten Benutzeroberfläche. Um nun die einzelnen Routinen verwenden zu können, muß die Bibliothek geöffnet werden. Daraus folgt, daß erst nach dem Öffnen einer Bibliothek die Routinen zu gebrauchen sind.

Fenster Know-How

Intuition stellt dem Computeranwender eine Vielzahl von Routinen, die sich mit der Verwaltung, Gestaltung und Handhabung von Fenstern beschäftigen und auf die der Programmierer beliebig zurückgreifen kann, zur Verfügung.

Zunächst möchte ich jedoch auf den Begriff „Fenster“ etwas genauer eingehen, denn viele Leser wissen noch nicht recht, was sie mit Fen-

stern anfangen können. Fenster kann man als selbständige Bildschirme verstehen, wobei ihre Anzahl auf einem Bildschirm nur vom Arbeitsspeicher des Rechners begrenzt ist. Sie können auf einem Bildschirm beliebig platziert werden, ebenso besteht die Möglichkeit, ihre Größe zu verändern und sie in den Hinter- oder Vordergrund zu setzen. Man sieht also, daß Fenster in der Lage sind, sich zu überlagern oder zu überschneiden, wobei jedes einzelne vollkommen unabhängig von den anderen verwaltet wird.

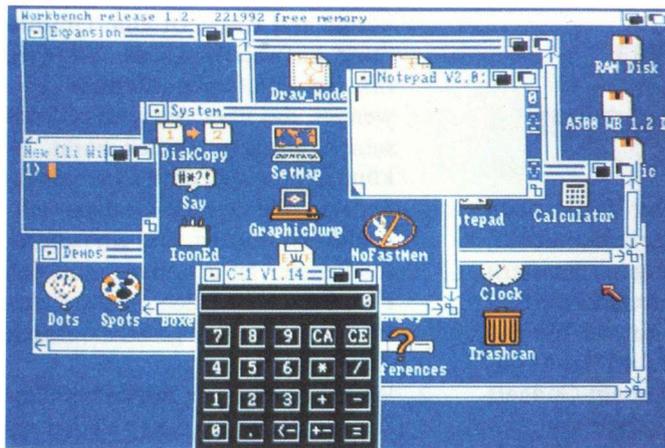


Abb. 1: Amiga Fenstervielfalt

Die Auflösung von einem Fenster wird von der des Bildschirms bestimmt. Man muß dazu erwähnen, daß Fenster immer mit einem Bildschirm verkettet werden müssen. Wobei das Anbinden nicht unbedingt mit der Standardanzeige von Intuition, dem Workbench-Bildschirm, vollzogen werden muß, genauso kann ein vom Programmierer erstellter Customscreen (also ein vom Programmierer erstellter Bildschirm) dazu verwendet werden (ausführliche Informationen über Custom- oder Workbenchscreens können Sie im ersten Kapitel des Intuition-Kurses nachlesen). Beispiel: Wird ein Fenster mit einem Workbenchscreen verkettet (d.h. darauf ausgegeben), der immer eine Auflösung von 640 * 200 Bildpunkten besitzt, würde ein Fenster, das über den ganzen Bildschirm gezogen ist, dieselbe Auflösung wie der Workbenchscreen besitzen, also 640 * 200 Bildpunkte. Andere Auflösungsstufen als beim Standardscreen von Intuition sind nur mit Hilfe von Customscreens zu

erzielen, da diese in allen Auflösungsstufen erstellt werden können.

Da im 'Innern' von Fenstern Programme unabhängig voneinander ablaufen können, nutzen Programme diese Eigenschaft häufig aus. Denn dadurch ist es möglich, daß mehrere Anwendungen gleichzeitig ablaufen können. Das Arbeiten von mehreren Programmen, scheinbar zur gleichen Zeit, nennt man Multitasking, wobei jedes Programm einen „Task“ repräsentiert. Andererseits sind Programme in der Lage, weitere Fenster aufzurufen, um beispielsweise Unter-

programme in ihnen ablaufen zu lassen.

In einem Fenster können andere wichtige Intuition-Komponenten eingebunden werden, beispielsweise die sogenannten Gadgets. Gadgets sind im allgemeinen Dialogboxen, die mit der Maus angewählt werden. Mit ihnen kann der Computer-Anwender der Software bestimmte Mitteilungen zukommen lassen. Dabei unterscheidet man drei Arten von Gadgets: Die Boolean-Gadgets sind Boxen, in denen der User eine Frage nur mit 'Ja' oder 'Nein' beantworten kann, die also nur eine 'negative' bzw. 'positive' Antwort ermöglichen. In String-Gadgets kann der Benutzer, nach dem Anwählen mit der Maus, einen Text oder eine Zahlenreihe mit der Tastatur eingeben. Die sogenannten proportionalen Gadgets sind Boxen, in denen sich ein Schieber befindet, der in alle Richtungen, mit Hilfe der Maus, bewegt werden kann. Doch darauf möchte ich im Moment nicht näher eingehen. Interessierte muß ich

daher auf ein folgendes Kapitel dieses Kurses verträsten.

Weiterhin können Pull-Down-Menüs (sie fallen nach Anwählen mit der Maus von der oberen Bildschirmzeile herab und geben dadurch eine Vielzahl von einzelnen Menüpunkten frei, die dann wiederum mit der Maus angewählt werden können) mit jedem Fenster verkettet werden; diese werden dann in der Titelzeile des Bildschirms, mit der das Fenster verbunden ist, angezeigt.

System-Gadgets in einem Fenster

Ich möchte an dieser Stelle noch einmal auf Gadgets zurückkommen, da man im Zusammenhang mit Fenstern zwei grundsätzliche Unterschiede machen sollte. Da sind einmal die System-Gadgets und daneben die Application-Gadgets. Als System-Gadgets bezeichnet man Gadgets, die vom Betriebssystem des Amiga bereits fest definiert sind. Application-Gadgets hingegen sind vom Programmierer selbst erstellt.

Bei einem Fenster besteht die Möglichkeit, verschiedene System-Gadgets einzubinden. Da ist das Gadget zum Schließen des Fensters in der oberen linken Ecke oder jenes, das das Fenster in den Vorder- oder Hintergrund setzt (rechts oben). Die verschiedenen System-Gadgets eines Fensters zeigt Abbildung 2.

Verschiedene Fenstertypen

Intuition gibt dem Programmierer die Möglichkeit, verschiedene Arten von Fenstern zu erstellen. Dabei behält jeder Typ die Grundeigenschaften von Fenstern.

Man unterscheidet vier Typen von Fenstern:

1. **Borderless Windows:** Borderless Fenster sind Fenster, die keinen sichtbaren Rand besitzen (randlos), die also nicht wie bei herkömmlichen Fenstern eine Umrandung aufweisen. Diese Art wird oft als Hintergrundfenster verwendet, das über den ganzen Bildschirm gezogen werden kann. Es können hingegen einem Backdrop-Fenster alle System-Gadgets eingebunden werden. Um ein randloses Fenster zu erstellen, muß das Flag `BORDERLESS` in der NewWindow-Struktur gesetzt werden, dazu später mehr.
2. **Backdrop Fenster:** Backdrop Fenster werden in manchen Fällen genauso genutzt wie die Borderless-Fenster, nämlich als Hintergrundfenster. Jedoch werden Backdropfenster immer im Hintergrund ausgegeben, es besteht nicht die Möglichkeit, diesen Fenstertyp vor ein herkömmliches zu setzen. Werden mehrere Backdrop-Fenster geöffnet, wird das momentan geöffnete Fenster immer vor das zuletzt geöffnete Fenster gesetzt. Dieser Art von Fenster kann nur ein System-Gadget zugewiesen werden, das

Closewindow Gadget, das nur das Schließen des Fensters erlaubt. Um ein Backdrop Fenster aufzurufen, muß in der Flag-Variablen der NewWindow-Struktur `BACKDROP` gesetzt werden.

3. **Gimmezerozero-Fenster:** Die Besonderheit eines solches Fensters besteht darin, daß es aus 'zwei' Flächen besteht: Der äußeren, größeren und der kleineren, inneren Fläche. Als äußere Fläche wird der Bereich bezeichnet, der die Titelzeile, die System-Gadgets und den Rand umgibt. Die innere Fläche hingegen ist nur der Bereich, der wirklich innerhalb des Fensters liegt, also ohne Titelzeile und Gadgets. Zur Verdeutlichung eines Gimmezerozero-Fensters dient Abbildung 3. Der entscheidende Vorteil eines solches Fensters gegenüber einem herkömmlichen besteht darin, daß die (0,0) Koordinate, nach der sich alle Berechnungen zur Ausgabe von Gadgets, Grafik und Text richten, nicht mehr der rechte, obere Punkt des Fensters ist, also vom äußeren Fenster. Die neue (0,0) Koordinate ist vielmehr der rechte, obere Punkt der inneren Fläche. Der Programmierer braucht durch diese Festlegung des neuen Koordinatenursprungs keine Zeit damit zu vergeuden, daß eine Ausgabe an das Fenster möglicherweise in der Titelzeile erscheint. Um ein solches Fenster aufzurufen, muß das Flag `GIMMEZEROZERO` in der NewWindow-Struktur gesetzt werden.
4. **SuperBitMap Window:** Ein SuperBitMap Fenster ist ein bestimmter Typ, der es einem Programmierer erlaubt, die Daten zur Neuerstellung eines Fensters in einem separaten Speicherbereich festzuhalten. Ein 'normales' Fenster hingegen legt die Daten für Größe, Gadgets, Menüs, Grafik oder Text in der BitMap (Speicher) des verketteten Bildschirms ab. Als BitMap wird demzufolge der Speicherbereich bezeichnet, in dem beispielsweise die Daten für Screens abgelegt werden. Ansonsten besitzt das SuperBitMap-Fenster keinerlei Unterschiede zum üblichen. Der entscheidende Vorteil bei der Zuweisung einer BitMap an ein Fenster, ist, daß auf diesen Spei-

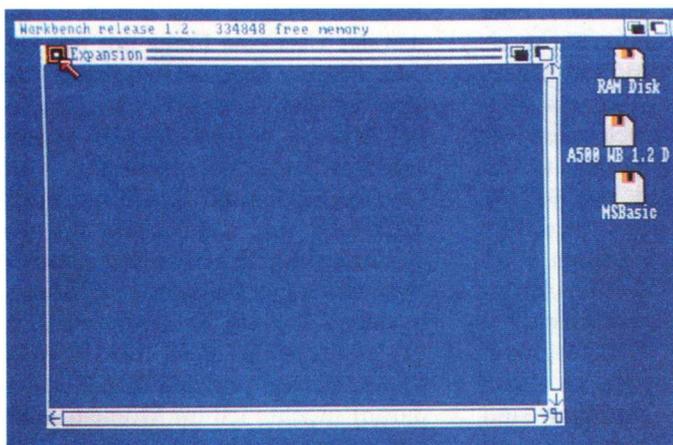


Abb. 2:
Ein Fenster mit allen System-Gadgets

cher beliebig zurückgegriffen werden kann. Dies ist nicht der Fall, wenn kein separater Speicher festgelegt wurde. Um ein solches Fenster zu erhalten, wird das Flag `SUPER_BITMAP` in der `NewWindow`-Struktur gesetzt, zudem muß eine `BitMap`-Struktur für das Fenster definiert werden.

Fenster Ein- und Ausgabe

Dem Computeranwender stehen zwei Arten zur Ein- bzw. Ausgabe im Zusammenhang mit Fenstern zur Verfügung: Der `Intuition Direct Communications Message Port`, kurz `IDCMP`, und der sogenannte `Console Device`. Der `IDCMP` kümmert sich um Eingaben über die Tastatur, um bestimmte Ereignisse, die mit der Maus oder anderen Eingabegeräten wie `Joystick` oder `Lightpen` zu tun haben, und ermöglicht die Kommunikation zwischen `Intuition` und einem Programm. Der `Console Device` gestattet hingegen fortlaufende Dateineingabe sowie `Intuition`-Nachrichten, umgewandelt in `ANSI`-Sequenzen (bestimmter Standard). Soll der `IDCMP` zur Eingabe genutzt werden, so müssen lediglich einige `IDCMP`-Flags in der `NewWindow`-Struktur festgelegt werden. Daraufhin erstellt `Intuition` zwei 'Message Ports', einen für In-

tuition, der andere steht dem Programmierer zur Verfügung. Auf die Bedeutungen der verschiedenen Flags möchte ich noch nicht näher eingehen, da dieses Kapitel ein umfangreiches Gebiet repräsentiert. Wenn der `Console Device` verwendet werden soll, muß dieser nach dem Öffnen des Fensters erstellt werden. Auf die verschiedenen Möglichkeiten der Nachrichtenübermittlung werde ich noch ausführlich in einem späteren Kapitel dieses Kurses zurückkommen. Interessierte Leser muß ich leider auf diesen Zeitpunkt vertrösten.

Fenster Refresh

Die Neuerstellung von Fenstern nennt man in der Fachsprache `Refresh`. Aber was bedeutet eigentlich `Refresh` und wozu ist es notwendig? Fenster sind in der Lage, sich zu überlappen; der Teil, der von dem überlagerten Fenster abgedeckt wird, muß irgendwo festgehalten oder gespeichert werden. Durch die verschiedenen Speichermethoden des verdeckten Bereiches und das Aufrufen der in diesem Speicher abgelegten Daten unterscheiden sich die einzelnen `Refresh`-Arten.

1. Simple-Refresh

`Intuition` stellt einem Computeranwender drei verschiedene `Refresh`-Methoden zur Verfügung. Die einfachste ist der `Simple-Refresh`: `Intuition` braucht sich keine Daten zu merken, für das Auffrischen von Fenstern ist jenes Programm verantwortlich, das das Fenster aufgerufen hat. Nur wenn das Fenster mit Hilfe der Maus an eine neue Position gesetzt wird, bewahrt `Intuition` den Inhalt des Fensters und baut es neu auf. Der Vorteil der `Simple-Refresh`-Methode besteht darin, daß das Bewahren der verborgenen Teile eines Fensters keinen wertvollen `RAM`-Speicher belegt, denn für diese Aufgabe wird der `Bildschirmspeicher` herangezogen. Ein Nachteil besteht bei dieser Methode allerdings darin, daß der `Refresh` langsamer ausgeführt wird als die beiden anderen Methoden. Um die `Simple-Refresh`-Methode verwenden zu können, muß in der `NewWindow`-Struktur das Flag `SIMPLE_REFRESH` gesetzt werden.

2. Smart-Refresh

Eine weitere Methode ist der `Smart-Refresh`. Dieser Typ benutzt nicht nur den `Bildschirmspeicher`, sondern behält die Informationen der Fenster in einen separaten `RAM`-Bereich. Die Neuerstellung übernimmt `Intuition`, d.h. das Programm braucht sich nicht um den Wiederaufbau des Fensters zu kümmern. `Smart-Refresh` besitzt gegenüber dem `Simple-Refresh` einen Nachteil: Er benötigt wesentlich mehr Speicherplatz, ist aber in der Ausführung wesentlich schneller. Wird die `Smart-Refresh` Methode gewünscht, muß das Flag `SMART_REFRESH` gesetzt werden.

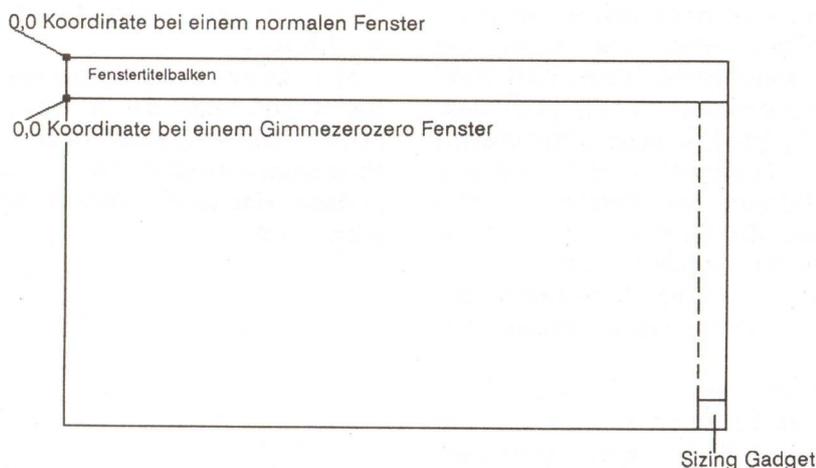


Abb. 3: Die Definition eines GIMMEZEROZERO Fensters

3. Super BitMap

Als SuperBitMap wird einmal ein spezieller Fenstertyp bezeichnet, zum anderen eine Refresh-Methode. Bei diesem Typus werden die Informationen der Anzeige in einem separaten RAM-Bereich abgelegt. Dieser Bereich muß jedoch vom Programmierer vorher definiert werden. Wie ein solcher Speicher definiert wird, können Sie aus Listing 3 ersehen. Man kann sich leicht vorstellen, daß die Methode natürlich den größten Speicherplatz benötigt. Ob sie angewandt wird, hängt vom Programmierer und seinen Vorstellungen ab. Weitere Informationen können Sie in dem Abschnitt über das SuperBit-Map Window, in diesem Kursteil, nachlesen.

Fensterprogrammierung gleich Zauberei?

Keineswegs, die Programmierung von Fenstern ist nüchtern betrachtet eine rein logische und leicht verständliche Angelegenheit, mit Zaubern oder Hexerei hat die Erstellung von Fenstern nichts zu tun! Auch der Computerlaie wird schnell in der Lage sein, Fenster in jeder möglichen Form zu erstellen.

Nachdem ich Ihnen eine großzügige Übersicht über Fenster vermittelt habe, möchte ich nun mit der eigentlichen Programmierung beginnen.

BASIC oder C?

Wenn Sie das mitgelieferte ABASIC Handbuch einmal aufmerksam durchgeblättert haben, ist Ihnen viel-

leicht der Befehl 'window' aufgefallen. Mit Hilfe dieses Befehls ist es ohne größeren Aufwand möglich, ein Standard-Fenster mit allen System-Gadgets und bereits integrierten IDCMP-Flags zu erstellen. Man sieht schnell, daß dem Programmierer, im wahrsten Sinne des Wortes, die Hände gebunden sind. Mancher Amiga-besitzer wird nicht zu unrecht meinen: „Was soll das denn, der BASIC-Befehl zum Aufrufen eines Fensters, die festeingebauten Gadgets bzw. festgelegten Flags reichen vollkommen aus für das effektive Programmieren von Fenstern!“ Unrecht hat der User mit dieser Äußerung sicher nicht. Die direkte Anpassung an eine bestimmte Aufgabe erfüllt der ABASIC Befehl aber kaum. Da Intuition in der Programmiersprache C erstellt ist, kann man sich vielleicht denken, daß das Ausnutzen der Intuition-Routinen wohl am einfachsten in der Sprache C ist. Aus diesem Grunde hielt ich die Sprache C für angebrachter. Laien sollten sich nicht durch diese Sprache verunsichern lassen: Zwar wird ihr eine gewisse Schwierigkeit nachgesagt, das Erlernen der Grundkenntnisse geht aber schnell vonstatten.

Fenster in BASIC

Beginnen möchte ich mit der Erstellung von Fenstern in BASIC. Der mitgelieferte ABASIC-Interpreter besitzt außer den üblichen Standardbefehlen noch einige andere, unter anderem auch einen Befehl, der es ermöglicht, Fenster aufzurufen. Die Syntax des Befehls lautet: WINDOW Fenster-Nummer [,Titel][, [Koordinaten] [,Typ] [,Bildschirm-Nummer]]] Die 'Fenster-Nummer' Variable identifiziert das Fenster. Hierbei können Zahlen von 1 - n (1 bis unendlich) eingegeben werden.

'Titel' ist eine Zeichenkette, die den Namen des neuen Fensters beinhaltet.

'Koordinaten' bestimmt die Koordinaten des Fensters, diese sollten in folgender Form eingegeben werden: (x1,y1)-(x2,y2), wobei (x1,y2) die obere, linke Ecke bestimmt und (x2, y2) die untere, rechte Ecke. Die Aus-

gabekoordinaten des Fensters beziehen sich immer auf die (0,0) Koordinate des Bildschirms, demnach sind alle Koordinaten relativ zu diesem absoluten Bildschirmpunkt zu berechnen.

Mit der 'Typ'-Variable wird die Art des Fensters bestimmt. Hier können Werte von 0 bis 31 eingesetzt werden, wobei jede Zahl einen bestimmten Typ von Fenster erzeugt. Fünf Grundfensterarten stehen dem BASIC-Programmierer zur Auswahl. Jede Fensterart besitzt einen bestimmten Zahlenwert, durch Addition der Zahlenwerte ist es dann möglich, daß 31 verschiedene Fenstertypen erzeugt werden können. Die Grundfenster mit den dazugehörigen Zahlenwerten sind folgende: '1' erzeugt das Sizing Gadget in der rechten, unteren Ecke des Fensters, welches mit Hilfe des Mauszeigers angewählt werden kann, um das Fenster zu vergrößern bzw. zu verkleinern. '2' gewährleistet, daß das Fenster mit der Titelzeile über den Bildschirm geschoben werden kann. '4' ruft die Gadgets in der rechten, oberen Ecke auf, mit der das Fenster in den Hinter- bzw. Vordergrund gesetzt werden kann. '8' erstellt das Close-Gadget in der linken, oberen Ecke. Dadurch besteht die Möglichkeit, das Fenster wieder zu schließen. '16' bestimmt, daß der Inhalt eines verdeckten Fensterteiles automatisch durch einen Refresh wieder dargestellt wird. Näherer Informationen über das Aufrufen in Fenstern können Sie auch aus Ihrem mitgelieferten BASIC-Handbuch entnehmen.

Die letzte Variable, 'Bildschirm-Nummer', bestimmt, auf welchen Bildschirm das Fenster ausgegeben werden soll.

Mein kleines Beispielprogramm in BASIC soll Ihnen das Arbeiten mit Fenstern ein wenig näherbringen. Das Programm erzeugt mehrere Fenster, in denen eine kleine Spielerei ausgegeben wird.

```
CLS
WINDOW 3, "Ausgabe 1", (10,10)-(200,130), 31
WINDOW 4, "Ausgabe 2", (240,10)-(430,130), 31

FOR i = 1 TO 120
  WINDOW OUTPUT 3
  a=RND * 90
  c=RND * 120
  LINE (a,a)-(c+a,c+a), 3, B
  WINDOW OUTPUT 4
  FOR k = 0 TO 3
    CIRCLE(a+c,c), 50,k
  NEXT k
NEXT i
```

Fenster in C

Wenn Sie das kleine Programm in BASIC abgetippt und gestartet haben, ist Ihnen vielleicht der recht langsame Aufbau und (oder) Neuaufbau der Fenster unangenehm aufgefallen. Durch das Programmieren von Fenstern in der Programmiersprache C können solche Geschwindigkeitsprobleme beseitigt werden. Die Programmiersprache C ist außerdem für das Erstellen von Fenstern hervorragend geeignet. Können doch alle nur erdenklichen Variationen, ohne größeren Aufwand, programmiert werden! Auch für Programmieranfänger ist der nun folgende Teil kein 'Kauderwelsch', sondern mit etwas Einarbeitung in die 'C-Materie' leicht verständlich. Freilich möchte ich hier keinen detaillierten C-Kurs anbieten, sondern vielmehr auf die verschiedenen Intuition-Routinen ausführlicher eingehen.

Programmierung

C-Programme besitzen immer ein bestimmtes Grundschema. Am Anfang eines C-Listings stehen immer allgemeine Festlegungen, die später im Hauptprogramm oder in Unterprogrammen benötigt werden. Nach dem Programmkopf kommt das sogenannte Hauptprogramm. Das Hauptprogramm beginnt immer an der Marke 'main()', gefolgt von einer offenen, geschweiften Klammer (), die dazugehörige schließende Klammer () bestimmt das Ausmaß und legt das Ende des Hauptprogrammes fest. Man stellt also fest, daß immer ein Klammerspaar zusammengehört. Nach dem Hauptprogramm können beliebig viele Unterprogramme erstellt werden, diese erkennt man an einer frei wählbaren Zeichenkette, gefolgt von einer öffnenden und schließenden Klammer, wobei innerhalb der Klammern einige Variablen übergeben werden können. Beispiel: Schluss (Variable1, Variable2). Ein geschweiftes Klammerspaar umschließt ebenfalls das Unterprogramm.

Um nun ein Fenster zu erstellen, müssen verschiedene Schritte vollzogen werden. 1. Zu Beginn muß eine NewWindow-Struktur erstellt werden. In ihr werden alle Parameter für das neue Fenster festgelegt. 2. Nach der Bestimmung der verschiedenen Parameter wird die Intuition-Routine OpenWindow(Zeiger), mit einem Zeiger auf die NewWindow Struktur, aufgerufen. Nach Aufrufen des neuen Fensters ist die NewWindow-Struktur nicht länger notwendig. Diese beiden Schritte müssen immer durchgeführt werden, um ein neues Fenster aufrufen zu können.

Die NewWindow Struktur hat folgendes Aussehen:

```
struct NewWindow
{
    SHORT LeftEdge, TopEdge;
    SHORT Width, Height;
    UBYTE DetailPen, BlockPen;
    USHORT IDCMPFlags;
    ULONG Flags;
    struct Gadget *FirstGadget;
    struct Image *CheckMark;
    UBYTE *Title;
    struct Screen *Screen;
    struct BitMap *BitMap;
    SHORT MinWidth, MinHeight;
    SHORT MaxWidth, MaxHeight;
    USHORT Type;
};
```

Die einzelnen Bedeutungen der Variablen und deren Möglichkeiten kann man in der Tabellarischen Übersicht entnehmen.

Die Routine CloseWindow() bewirkt, das sich das zuvor geöffnete Fenster schließt.

Eine NewWindow-Struktur muß also mit den verschiedenen Parametern ausgestattet werden. Solche Strukturen kann man in der Programmiersprache C leicht selbst erstellen. Alle verwendeten Strukturen und Definitionen der Intuition-Library sind auf der Compiler-Diskette im Direktory 'intuition' in der Datei 'intuition.h' abgelegt. Um eine vordefinierte Datei verwenden zu können, muß diese nur an das C-Programm gebunden werden. Das Einbinden geschieht mit dem '#include <Datei>' Ausdruck.

Auch die NewWindow-Struktur ist in der 'intuition.h' Datei zu finden, deren Definition in dem nebenstehenden Kasten ausgedruckt ist.

Die Verwendung der NewWindow Struktur und das darauf folgende Aufrufen eines Fensters habe ich in Listing 2 nachvollzogen.

Fensterzirkus

Als Fensterzirkus möchte ich weitere Funktionen bezeichnen, die in irgendeinem Zusammenhang mit Fen-

stern stehen. Davon stellt uns Intuition eine ganze Menge zur Verfügung. Zunächst die schon erwähnten OpenWindow(Zeiger)- und CloseWindow(Zeiger)-Funktionen. Sie öffnen bzw. schließen ein Fenster. Die erste Funktion beinhaltet einen Zeiger auf die NewWindow-Struktur, die – wie Sie wissen – die Parameter des neuen Fensters in sich definiert, die zweite hingegen einen Zeiger auf eine Window-Struktur.

Neben diesen Funktionen bietet uns Intuition eine weitere, die es uns ermöglicht, die Werte der minimalen sowie der maximalen Höhe und Breite des Fensters neu festzulegen. Sie wissen, daß durch das Sizing-Gadget die Größe verändert werden kann. Durch die nun folgende Funktion ist es möglich, die Grenzen neu festzulegen. Die Syntax der Funktion lautet: WindowLimits(Window, MinWidth, MinHeight, MaxWidth, MaxHeight) Die 'Min...' bzw. 'Max...'-Variablen beinhalten die minimale und maximale Größe, 'Window' ist wiederum ein Zeiger auf eine Window-Struktur.

Eine weitere interessante Funktion ist SetWindowTitles(...). Mit ihrer Hilfe ist das Ändern des Fenstertitels ein Kinderspiel. Mit gleicher Routine kann nebenbei auch noch der Bildschirmtitel neu erstellt werden. Der Syntax ist wie folgt definiert: SetWindowTitles(Window, Windowtitle, Screentitle) 'Window' ist mittlerweile schon ein Begriff, der Zeiger auf eine Window-Struktur. 'Windowtitle' beinhaltet den neuen Titel des Fensters, 'Screentitle' den des Bildschirms. Durch Ersetzen der Titels mit '-1' oder '0' wird der alte Titel behalten (-1) oder ganz gelöscht (0).

Ohne größeren Aufwand ist auch das Vergrößern oder Verkleinern eines Fensters zu bewerkstelligen. Hierbei muß lediglich die Funktion mit dem Syntax SizeWindow(Window, DeltaX, DeltaY) aufgerufen werden. 'Window' ist bekannt, 'DeltaX' und 'DeltaY' bezeichnen den Wert, um den die Größe entlang der X- bzw. Y-Achse verändert werden soll.

Die Funktion MoveWindow(Window, DeltaX, DeltaY) setzt das Fenster an eine neue Position, wobei 'Window' wohl nicht näher erläutert werden muß und 'DeltaX' und 'Del-

taY' die neuen Koordinaten des Fensters festlegen.

Zwei weitere Intuition-Routinen erlauben es dem Programmierer, Fenster in den Vorder- und Hintergrund zu setzen. Dies sind die Funktionen WindowToFront(Window) und WindowToBack(Window).

Zum Schluß meines kleinen Fensterzirkusses möchte ich noch auf zwei zusammengehörige Funktionen eingehen, die einen Refresh bewirken (siehe Abschnitt Refresh). Dadurch ist es einem Programmierer möglich, durch das eigene Programm einen Refresh zu erzeugen. BeginRefresh(Window) und EndRefresh(Window) lautet die jeweilige Syntax der Funktionen. 'Window' beinhaltet wie immer den Zeiger auf die Window-Struktur.

Um die hier aufgegriffenen Funktionen in der näheren Anwendung darzulegen, habe ich Listing 5 erstellt. Es beinhaltet zum Teil die hier angewandten Funktionen und muß lediglich an der markierten Stelle von Listing 2 eingesetzt werden.

Programm- erläuterung

Auf das BASIC Listing (Listing 1) möchte ich nicht näher eingehen, lassen Sie sich 'mal überraschen. Viele werden sicherlich auch sofort festgestellt haben, was das Programm in seiner Ausführung vollführt.

Listing 2 ist ein in C geschriebenes Programm und sollte meiner Ansicht nach noch etwas näher erläutert werden, damit auch jeder Amiga-User nähere Einblicke in die Programmierung von Intuition in der Programmiersprache C erhält. Den allgemeinen Aufbau eines C-Programmes habe ich schon an anderer Stelle erklärt und möchte deshalb mit den näheren Erklärungen des Listings fortfahren.

Zu Beginn des Listings, Zeile 15 und 16, werden zwei Dateien in das Programm eingebunden, diese enthalten bereits vorgefertigte Ausdrücke. Die erste Datei Namens 'ty-

pes.h' befindet sich hierbei im Ordner 'exec' und enthält Variablendefinitionen wie beispielweise 'void', 'int' oder 'LONG'. Es ist anzuraten, die Datei immer einzubinden, weil fast jedes C-Programm diese Definitionen benötigt. Die 'intuition.h'-Datei ist im Ordner 'intuition' zu finden und beinhaltet vorgefertigte Intuition Strukturen und dazugehörige Flag-Definitionen. Der nächste Programmschritt findet in Zeile 22 statt und versieht die IntuitionBase-Struktur mit einem Zeiger desselben Namens. In Zeile 26 und 27 werden zwei Variablen mit einem Wert versehen, genauso können die Werte direkt eingegeben werden. In Zeile 31 bis 51 findet die NewWindow-Struktur Verwendung, hier werden dem zukünftigen Fenster die Aufgaben, das Erscheinungsbild, die Größe usw. mitgeteilt. Das ACTIVATE-Flag ist momentan das einzige, das gesetzt ist, es bewirkt lediglich, daß das Fenster nach dem Aufruf in einen aktiven Zustand gesetzt wird. Jedoch sind keinerlei andere Flags gesetzt, zum Beispiel werden keine System-Gadgets im Fenster plaziert oder keine Refresh-Methode wird übermittelt. Geben Sie anstatt ACTIVATE nachstehendes ein, werden alle System-Gadgets eingebunden, sowie eine Refresh Art festgelegt:

WINDOWSIZING | WINDOW-
DEPTH | WINDOWCLOSE | WIN-
DOWDRAG | SMART_REFRESH,

Natürlich können Sie die Flags nach ihren Wünschen setzen. Mit der NewWindow-Struktur ist der Programmkopf beendet, es folgt das Hauptprogramm, das wie Sie wissen mit dem Label 'main()' beginnt. Die Zuweisung der Window Struktur mit einem Zeiger desselben Namens und die Festlegung der Variablen 'i' als Integerwert (nur ganzzahlige Werte) sind die ersten Programmschritte im Hauptprogramm, wobei beide ohne weiteres im Programmkopf erfolgen könnten. In Zeile 71 beginnt das Öffnen der Intuition Library, denn, man erinnere sich, ohne das Öffnen einer solchen können die Routinen, die die Library enthält, nicht verwendet werden. Das Öffnen der Library ist nach einem bestimmten Schema programmiert, wenn es nicht möglich war, die

Library zu öffnen, beendet das Programm seinen Ablauf, hierbei wird ein Wert an 'IntuitionBase' übergeben. Ist dieser gleich Null, konnte die Library nicht geöffnet werden und das Programm wird abgebrochen. Ab Zeile 73 findet der Aufruf des Fensters statt, wobei wiederum der Ablauf des Programms abgebrochen wird, wenn 'Window' der Wert Null zugewiesen wurde. Dies bedeutet, daß aus irgendeinem Grund das Fenster nicht geöffnet werden konnte. Der nächste Programmschritt beinhaltet eine Warteschleife, deren einziger Sinn nur darin besteht, den weiteren Programmablauf hinauszuzögern. Denn der letzte Programmteil, Zeile 88 - 90, schließt das zuvor geöffnete Fenster sowie die Intuition Library. Jetzt werden Sie auch sicherlich sehen, welcher nähere Zweck der ansich sinnlosen Schleife zukommt. Würde sie nämlich fehlen, würde das geöffnete Fenster sofort wieder geschlossen werden und man könnte das 'frisch produzierte' Fenster weder in Augenschein nehmen noch die eventuell eingebauten SystemGadgets testen, weil die Zeit dazu fehlt. Sicherlich ist der hier beschriebene Weg ohne jeden Zweifel nicht der komfortabelste, ist er doch für eigene Programme völlig ungeeignet. Eine weitaus bessere Art, ein Fenster zu schließen, vor allem wann immer es beliebt, ist der Weg, der über das 'Close Window'-Gadget führt. Sie wissen, durch Anwählen des Gadgets mit der Maus wird das Fenster geschlossen. Die Programmierung dieser Art ist jedoch etwas komplexer und vor allem reicht es in ein Gebiet hinein, dem man ein eigenes Kapitel widmen könnte. Dennoch gedenke ich wenigstens die Programmierung abzudrucken, obwohl ich an dieser Stelle auf nähere Erläuterungen verzichten möchte. Listing 4 zeigt lediglich den Weg der Programmierung. Wie das Programm eingebunden werden muß, habe ich in Kommentarzeilen dem Listing beigefügt.

GO AMIGA!

	sFr.	DM		sFr.	DM		sFr.	DM		sFr.	DM
ABSOFT			DIAMOND SOFTWARE			MERIDIAN SOFTWARE			PSYGNOSIS		
AC Fortran	590	649	*Investor	1499	1699	Zing CLI Deluxe	159	179	Bratuccas + Arena	79	89
AC Basic	390	429	DEMO Investor	15	15	DEMO Zing	15	15	*Barbarian	69	79
ACCESS SOFTWARE			C monitor V 200	179	199	*Games Gallery	59	69	*Terrorpods	69	79
*Tenth Frame	79	89	DIGITAL CREATIONS			*Zing Keys	99	119	RAINBIRD		
ACCESS ASSOCIATES			Gizmos 2.0	139	159	METACOMCO			Jewels of Darkness	39	49
Alegra 512K RAM	759	879	*D'Buddy	159	179	Shell CLI Deluxe	99	119	SIERRA ONLINE		
Alegra 2M RAM	1499	1649	Digital Link	139	159	Assembler	159	179	Kings Quest 1	89	99
Autopal Expander	68	70	DIGITAL SOLUTIONS			Toolkit	89	99	Kings Quest 2	89	99
ACCOLADE			*LPD Writer	259	289	MICHIGAN SOFTWARE			Kings Quest 3	89	99
Mean 18 Golf	59	69	*LPD Filer	259	289	Insider - 1 M RAM			SLIPPED DISK		
ACTIVISION			*LPD Planner	259	289	interne Expansion	699	769	BASIC		
Hacker I	49	59	DISCOVERY			MICRO ILLUSIONS			Tastaturschablone	19	25
Borrowed Time	49	59	Marauder II	79	89	Dynamic CAD	999	1149	DOS Tastaturschablone	19	25
Mindshadow	49	59	Grabbit	59	69	DEMO Dynamic CAD	15	15	SOFTWARELAND		
Shanghai	69	79	ECE RESEARCH			*Fairy Tale Adv.	99	119	jede Disk Public	15	15
Portal	79	89	ECE Midi Interface	119	139	MICROPOSE			10er Pack Public	135	135
Ch.ship Golf	69	79	Amiga DOS Ref Card	6	7	Silent Service	69	79	GO AMIGA! Datei	179	199
Ch.ship Basketball	69	79	*Speedy Amigos	129	139	*Gunship	79	89	DEMO Datei	15	15
Ch.ship Baseball	69	79	ELECTRONIC ARTS			*F 15 Strike Eagle	79	89	*GO AMIGA! Text	179	199
Ch.ship Football	79	89	Deluxe Paint 2	199	249	MICROSMITHS			SSI		
*Write Choice Elite	159	179	Paint Data 1	59	69	TxD	79	89	Computer Baseball	59	69
AGIS DEVELOPMENT			Paint Data 2	59	69	Fast Fonts	79	89	SUBLOGIC		
Aegis Draw	249	249	Paint Data 3	59	69	MICROSOFT PRESS			Flightsimulator 2	99	119
DEMO Draw	15	15	Deluxe Print	119	139	The Amiga Buch	59	69	*Jet	99	119
Aegis Draw plus	499	499	*Deluxe Video 2	199	249	MICRO SYSTEMS SOFTWARE			*Scenery Disk	49	59
Aegis Animator	279	279	Deluxe Music	199	249	Analyze V 2.0	299	349	SUNRISE INC.		
DEMO Animator	15	15	Music Data 1	59	69	BBS Bulletin Board	199	229	Perfect Sound	159	179
Aegis Images	79	89	Articfox	59	69	Online	139	159	DEMO Perfect Sound	15	15
Aegis Art Pak # 1	59	69	Adventure Construct.	66	79	Scribble plus	199	229	SWISS COMPUTER ARTS		
Aegis Impact	179	199	*Demo Adventure	15	15	Organize	199	229	*Deluxe		
Aegis Sonix	159	189	Marble Madness	59	69	MILES COMPUTING			Titelkonstruktionsset	79	89
*Aegis Diga!	159	189	Chessmaster	79	89	Quintette	89	99	DEMO Titelseit.	15	15
*Arozok's Tomb	99	119	Bards Tale	89	99	MIMETICS (SOUNSCAPE)			TDI SOFTWARE		
AMISOFT			*Demo Bards Tale	15	15	Pro Studio	299	349	Modula-2 Standard	199	219
Diskzine Nr. AA-AG	19	29	EPYX			Midi Interface	99	119	Modula-2 Developer	299	349
ANAKIN RESEARCH			Wintergames	59	69	Sound Sampler	199	229	Modula-2 Commercial	599	699
The Easel	899	999	FAUG			MINDSCAPE			DEMO Modula-2	15	15
DEMO Easel	15	15	RoboCity Zeitschrift	4	5	Deja Vue	79	89	Examples Disk	49	59
APPLIED VISIONS			Jahresabonnement	42	54	Univited	79	89	Amiga Editor	99	119
Future Sound	350	399	GOLD DISC INC.			Halley Project	59	69	The Grid	99	119
ARRAY'S INC.			Pagesetter	299	349	Balance of Power	79	89	Kermit Source	49	59
Master AM DOS Buch	39	49	Gold Spell	89	99	Defenders of Crown	79	89	TAURUS IMPEX		
ARTWORX			*Lazerscript	89	99	Sindbad	79	89	Acquisition	599	649
Strippoker	69	79	GOLDEN HAWK			SDI	79	89	DEMO Acquisition	30	30
ASSOCIATED COMPUTER			GoldMidi Interface	159	179	*King of Chicago	89	99	TRUE BASIC		
Brush Works	69	79	GRAFOX OF ENGLAND			Perfect Score	159	179	Interpreter	299	349
BANTAM			Logistic english	299	329	Masterlype	79	89	Compiler Runtime	299	349
AM Manual Buch	59	69	IDEAS CREATED			NEW HORIZONTS			Developers Toolkit	99	119
Express DOS Refdisk	59	69	Filter für HI-Res	29	39	Flow	199	229	Libraries:		
AM User Guide Buch	49	59	IMPULSE			DEMO Flow	15	15	Sorting + Searching	99	119
BAUVILLE			*Prism	139	159	Pro Write	259	289	Advanced Strings	99	119
Video Vegas	59	69	*Proclip	59	69	NEW-TEK			3 D Graphics	99	119
BETHESDA SOFTWARES			INFINITY SOFTWARE			Digi View	399	459	Programme:		
Gridirion	119	139	Grand Slam Tennis	69	79	DEMO Digi View HAM	15	15	Algebra 1	99	119
DEMO Gridirion	15	15	*Galileo	199	219	*Digi Paint	119	129	Algebra 2	99	119
BROWN WAGH			INNOVATIVE TECHNOLOGY			NORTHEAST SOFTWARE			Calculus	99	119
Zuma Fonts	59	69	Diskmappe 3 1/2	19	24	*Publisher	399	459	Discrete Maths	99	119
DEMO Zuma Fonts	15	15	Diskmappe 5 1/4	19	24	*Organizer	99	119	Probability Theorie	99	119
TV Text	199	249	INOVATRONICS			ORIGIN SYSTEMS			Trigonometry	99	119
DEMO TV Text	15	15	Power Windows	179	199	Ogre	69	79	Pre Calculus	99	119
C LTD.			INTERACTIVE SOFTWARES			Ultima 3	120	135	TrueStat	159	179
aMEGA 1 M RAM	999	1100	*The Calligrapher	199	219	Ultima 4	128	135	UNISON WORLD		
HARDDISK 20 M	2000	2200	ISM INC.			*Autoduel	99	119	Printmaster plus	99	119
TIMESAVER	159	179	The Surgeon	119	129	OTHER GUYS			Art Gallery 1	59	69
CENTRAL COAST SOFTWARE			JAGHWARE INC.			Omega File	159	179	Art Gallery 2	59	69
DOS 2 DOS	118	128	*Alien Fires	79	89	Great States	79	89	VERSASOFT		
*Disk 2 Disk	99	119	JUMPDISK			Promise	99	119	dBMan	299	349
CLASSIC IMAGE			Diskmagazin	19	29	PACIFIC CYPRESS			DEMO dBMan	15	15
Diablo	59	69	LATTICE			Expander 2M RAM	1399	1599	VIZA SOFTWARE		
COMMODORE			S Compiler V 3.1	450	450	PARCTECH			*Vizawrite english	299	359
AmigaTerm	99	119	Unicalc	179	179	Kabel Parallel	39	49	DEMO english	15	15
Enhancer	30	40	Mac Library	199	199	Kabel RS 232	29	39	*Vizawrite deutsch	398	498
1,2 KS/WB	30	40	dBC III	299	299	PIM PUBLICATIONS			VIP TECHNOLOGIES		
AM Jahrbuch 1987	10	10	MANX SOFTWARE			Amazing Zeitschrift	7	8	VIP Prof english	299	329
COMPUMED			C Commercial	999	1149	Jahresabonnement	72	84	WAVE PAD		
Mirror Kopierer	99	119	C Developers	599	699	PRECISION SOFTWARE			Mouse Pad	12	15
Mirror Hacker Pak	99	119	*C Personal	399	449	Amiga System Buch	39	49	WESTCOM INDUSTRIES		
COSMI			MARKT + TECHNIK			PROGRESSIVE PERIPHERALS			ADFO		
Super Huey	59	69	Superbase	199	249	CLI Mate	69	79	Disk File Organizer	99	119
CREATIVE MICRO								HardHat			
KICK ELIMINATOR	259	289						Harddisk Backup	139	159	

* = Lieferbar nach Verfügbarkeit.

Preisänderungen vorbehalten.

Bestellservice:

BRD: 0041-1-311 5959
CH: 01-311 5959

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)

Wie geht's weiter?

Im nächsten Intuition-Kurs möchte ich auf die Programmierung von Gadgets aller Art eingehen. Dieser sicherlich hochinteressante Teil meines Kurses wird Ihnen weitere Einblicke in die Programmierung von Intuition geben.

(AK)

Tabellarische Übersicht

Die Erklärungen der Variablen in der NewWindow-Struktur. Durch ihre Veränderung kann man einem Fenster ein bestimmtes Erscheinungsbild geben und natürlich auch den Verwendungszweck genau bestimmen. Die vielfältigen Möglichkeiten der Fenstermanipulation liegt den einzelnen Variablen zugrunde.

LeftEdge

Legt die X-Koordinate der linken, oberen Ecke des Fensters fest.

TopEdge

Die Y Koordinate der linken, oberen Ecke des Fensters.

Width

Die Breite des Fensters in Bildpunkten.

Height

Die Höhe des Fensters in Bildpunkten.

Die Dimensionierung des Fensters ist relativ zu der oberen, linken Ecke des Bildschirms. Die Koordinaten dieses Punktes sind immer (0,0).

DetailPen

Farb Register Nummer für die Titelleiste und Gadgets. Wird das Fenster auf dem Workbench-Bildschirm ausgegeben, stehen nur vier Farbwerte zur Verfügung (0-3), auf einem sogenannten Customscreen bestimmt die Depth-Variable der NewWindow-Struktur, wie viele Farbnummern verwendet werden können. Näheres ist in der letzten Kickstart-Ausgabe

nachzulesen, in der das Kapitel Screens ausführlich behandelt wurde. BlockPen Farb Register Nummer für den Titellisten und die Umrandung des Fensters. Hierbei gelten dieselben Bedingungen wie bei der DetailPen-Variablen.

IDCMPFlags
REQVERIFY
REQCLEAR
REQSET
MENUVERIFY
SIZEVERIFY
NEWSIZE
REFRESHWINDOW
RAWKEY
MOUSEBUTTONS
MOUSEMOVE
GADGETDOWN
GADGETUP
CLOSEWINDOW und
MENUPICK.

Flags

Durch die Flag-Variable können bestimmte Eigenschaften in ein Fenster implementiert werden.

a.

Die nachstehenden Flags geben System Gadgets in das Fenster aus. Es besteht die Möglichkeit, alle oder nur einzelne zu setzen.

WINDOWDRAG

erlaubt das Fenster durch den Bildschirm zu ziehen. WINDOWDEPTH gibt dem Benutzer die Möglichkeit, das Fensters in den Hinter- oder Vordergrund zu setzen und setzt das Gadget in der rechten, oberen Ecke des Fensters. WINDOWCLOSE erstellt nur das Standard-Close-Gadget in der linken, oberen Ecke. Durch alleiniges setzen dieses Flags ist das Schließen des Fensters nicht durchführbar.

WINDOWSIZING

erlaubt das Fenster in der Größe zu verändern und setzt das Sizing-Gadget in der rechten, unteren Ecke. Haben Sie das WINDOWSIZING Gadget erstellt, müssen Sie SIZEBOTTOM und (oder) SIZEBRIGHT setzen.

b.

GIMMEZEROZERO erzeugt einen bestimmten Fenstertyp. Nähere In-

formationen können im Abschnitt über unterschiedliche Fensterarten nachgelesen werden.

Drei verschiedene Refresh-Arten können für ein Fenster gewählt werden, wobei immer nur eine Art verwendet werden kann. Näheres ist im Abschnitt 'Neuerstellungen von Fenstern' nachzulesen.

SIMPLE_REFRESH
SMART_REFRESH
SUPER_BITMAP

BACKDROP

wird gesetzt, wenn Sie ein spezielles Intuition-Untergrundfenster wünschen. Genauere Angaben über die Auswirkungen des Flags kann in dem Abschnitt 'Verschiedene Fenstertypen' nachgelesen werden.

REPORTMOUSE

können Sie setzen, wenn Sie feststellen wollen, an welcher Stelle die Maus sich befindet. Dieses Flag arbeitet mit dem IDCMP Flag MOUSEMOVE zusammen.

BORDERLESS

erstellt ein Fenster ohne Rand. Nähere Erläuterungen finden Sie in dem Abschnitt 'Verschiedene Fenstertypen'.

ACTIVATE

setzt ein Fenster nach dem Aufruf gleich in den aktiven Zustand. Ein inaktives Fenster erkennt man daran, daß der Fenstertitel 'ghostet' dargestellt ist.

NOCAREREFRESH

wird gesetzt, wenn Sie keine Nachrichten für das 'Auffrischen' von Fenstern erhalten wollen.

ACTIVEWINDOW und INACTIVEWINDOW

können zusammen oder einzeln gesetzt werden. Diese Flags erkunden, ob Ihr Fenster sich in einem aktiven oder inaktiven Zustand befindet.

Gadget

Beinhaltet einen Zeiger auf das erste Gadget von einer verketteten Liste, welches für das Fenster bestimmt ist.

CheckMark

Ein Zeiger auf eine Image-Struktur.

Eine Image-Struktur verwaltet Daten eines Bildes. Sollte also ein Bild in einem Fenster ausgegeben werden, muß diese Variable einen Zeiger auf eine Image-Struktur aufweisen.

Screen

Ein Zeiger auf eine Screen-Struktur. Wird NULL gesetzt, wenn der Standard-Screen (Workbenchscreen) verwendet werden soll. Text Ein String, der als Titel des Fensters verwendet wird. Die Farben des Strings sind diejenigen, die mit BlockPen und DetailPen festgelegt werden.

BitMap

Beinhaltet einen Zeiger auf eine Bit-Map Struktur. Muß nur gesetzt werden, wenn in der Flag Variablen als Refresh Typ SUPER_BITMAP spezifiziert wurde.

Type

Legt fest, auf welchen Bildschirmtyp das Fenster ausgegeben werden soll. Einzusetzen ist entweder WBENCH-SCREEN für den Standardbildschirm oder CUSTOMSCREEN für einen selbst erstellten. MinWidth Die minimale Breite des Fensters.

MinHeight

Die minimale Höhe des Fensters.

MaxWidth

Die maximale Breite des Fensters.

MaxHeight

Die maximale Höhe des Fensters.

Die letzten vier Variablen legen fest, in wie weit das Fenster vergrößert bzw. verkleinert werden kann. Die Angaben müssen in Bildpunkten erfolgen.

```

1: /*****
2: * Zusatzprogramm zum Definieren einer BitMap *
3: * für einen Refresh. *
4: *****/
5: * Nachstehende Schritte müssen befolgt werden: *
6: * In Zeile 61, von Listing 2, fügen Sie *
7: * Speicher (); *
8: * ein und hängen dieses Listing zusätzlich an *
9: * Listing 2. *
10: *****/
11:
12: Speicher();
13: {
14:   struct BitMap Speicher;
15:   int k;
16:   InitBitMap(&Speicher,2,640,200);
17:
18:   for(k=0;k<2;k++)
19:   {
20:     Speicher.Planes[k]=AllocRaster(640,200);
21:     if(Speicher.Planes[k]==NULL)
22:     {
23:       printf("Kein freier Speicherplatz");
24:     }
25:   }
26: }

```

Listing 2

```

1: /*****
2: * *
3: * Das Erstellen eines Fensters *
4: * Autor : Andreas Krämer *
5: * Datum : 1.Mai.1987 *
6: * Erstellt mit dem Lattice C Compiler *
7: * *
8: *****/
9:
10: /*****
11: * Zwei 'include' Dateien werden in das Programm *
12: * eingebunden. *
13: *****/
14:

```

```

15: #include <exec/types.h>
16: #include <intuition/intuition.h>
17:
18: /* Zuweisung der IntuitionBase Struktur mit einem *
19: * Zeiger des elben Namens. Wird zum öffnen und *
20: * schließen der Intuition Library gebraucht. */
21:
22: struct IntuitionBase *IntuitionBase;
23:
24: /* Variablen Definition */
25:
26: #define INTUITION_REV 0
27: #define Million 1000000
28: /* Für Listing 4 void *Nachricht */
29: /* Die NewWindow Struktur */
30:
31: struct NewWindow NeuesFenster =
32: {
33:   20, /*X Bestimmung*/
34:   20, /*Y Bestimmung*/
35:   400, /*Breite*/
36:   100, /*Höhe*/
37:   0, /*Farbwert von DetailPen*/
38:   1, /*Farbwert BlockPen*/
39:   NULL, /*Keine IDCMPFlags*/
40:   ACTIVATE, /*Flags*/
41:   NULL, /*Keine Ausgabe eines Gadgets*/
42:   NULL, /*Kein Ausgabe eines Bildes*/
43:   "Mein Fenster", /*Fenster Title*/
44:   NULL, /*Kein Custom Screen*/
45:   NULL, /*Kein BitMap für das Auffrischen*/
46:   10, /*Minimale Breite*/
47:   10, /*Minimale Höhe*/
48:   640, /*Maximale Breite*/
49:   200, /*Maximale Höhe*/
50:   WBENCHSCREEN /*Bildschirm Typ*/
51: };
52:
53: /*****
54: * *
55: * Hauptprogramm *
56: * *
57: *****/
58:
59: main()
60: {
61:

```

```

62: int i;
63:
64: /* Zuweisung der Window Struktur mir einem Zeiger   *
65: * des selben Namens.                                *
66:
67: struct Window *Window;
68:
69: /* Öffnen der Intuition Library */
70:
71: IntuitionBase = (struct IntuitionBase *)
72:     OpenLibrary("intuition.library", INTUITION_REV);
73: if(IntuitionBase==NULL)exit(FALSE);
74:
75: /* Öffnen des Fensters */
76:
77: if((Window=(struct Window *)
78:     OpenWindow(&Neues Fenster))==NULL)
79: exit(FALSE);
80:
81: /* Listing 4 anfügen */
82: /* Einfache Warteschleife */
83: for(i=0;i<Million;i++);
84:
85:
86: /* Schließen des Fensters und der Library */
87:
88: CloseWindow(Window);
89: CloseLibrary(IntuitionBase);
90: exit(TRUE);
91:
92: /* Programmende */
93: }

```

Listing 3

```

1: /*****
2: * Zusatzprogramm zum Definieren einer BitMap *
3: * für einen Refresh.                        *
4: *****/
5: * Nachstehende Schritte müssen befolgt werden: *
6: * In Zeile 61, von Listing 2, fügen Sie *
7: * Speicher(); *
8: * ein und hängen dieses Listing zusätzlich an *
9: * Listing 2. *
10: *****/
11:
12: Speicher();
13: {
14:     struct BitMap Speicher;
15:     int k;
16:     InitBitMap(&Speicher,2,640,200);
17:
18:     for(k=0;k<2;k++)
19:     {
20:         Speicher.Planes[k]=AllocRaster(640,200);
21:         if(Speicher.Planes[k]==NULL)
22:         {
23:             printf("Kein freier Speicherplatz");
24:         }
25:     }
26: }

```

Listing 4

```

1: /*****
2: * Zusatzprogramm um das Fenster über das *
3: * 'Close Window' Gadget zu schließen. *
4: *****/
5: * Zu Beginn müssen folgende Schritte erfolgen: *
6: * In Listing 2 muß alles nach der Zeilennummer *
7: * 81 gelöscht werden. Danach setzen Sie dieses *
8: * Listing ein. Zum reibungslosen Ablauf sollte *
9: * Zeile 67, von Listing 2, in die Zeile 23 *
10: * verschoben und der Ausdruck *
11: * void *nachricht in Zeile 28 eingesetzt *
12: * werden. Wenn Sie die Anweisungen genau *
13: * befolgt haben, können Sie die das Fenster *
14: * nun über das 'Close Window' Gadget schließen.*
15: *****/
16:
17: for(;;)
18: {
19:     if(nachricht = GetMsg(WInow -> UserPort))
20:     {
21:         ReplyMsg(nachricht);
22:         Schluss();
23:     }
24: } /* for */
25: } /* Schließende Klammer des Hauptprogrammes */
26:
27: Schluss() /* Unterprogramm */
28: {
29:     CloseWindow(Window);
30:     CloseLibrary(IntuitionBase);
31:     exit(TRUE);
32: }

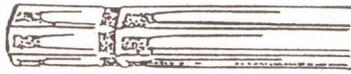
```

Listing 5

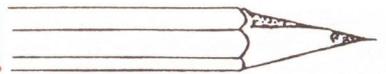
```

1: /*****
2: * Zusätzliche Fensterfunktionen *
3: *****/
4: * Das Listing wird an Listing 2 angehängt. *
5: * In Zeile 80, von Listing 2, muß jedoch *
6: * der Ausdruck *
7: * Funktionen(); *
8: * angefügt werden und Zeile 67 sollten Sie in *
9: * Zeile 23 setzen. *
10: *****/
11:
12: Funktionen()
13: {
14:     WindowLimits(Window,20,20,400,100);
15:     /* Die neuen minimal und maximalen Grenzen */
16:
17:     SetWindowTitles(Window,"Der neue Titel",-1);
18:     /* Setzt einen neuen Titel */
19:
20:     SizeWindow(Window,100,100);
21:     /* Verkleinert das Fenster */
22: }

```



Graph Zeichenmal

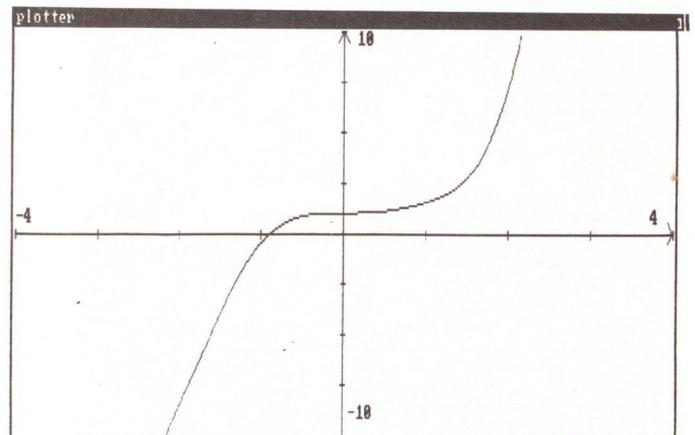
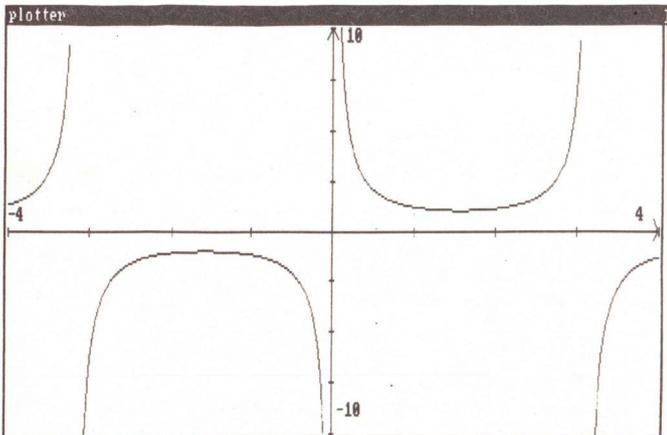


...hat zugeschlagen: Wer bisher den Programmtext von Hand änderte oder einen Parser verwendete, kann nun eine andere Methode zur Funktionseingabe in ein laufendes Programm kennenlernen.

Ursprung dieses Programmes ist der Ärger über lange und langsame Parser sowie die wenig elegante Methode, ein Programm von Hand zu ändern. Die Anwendung eines Parsers, die sich in Compilersprachen nicht umgehen läßt, ist bei einer interpretierenden Sprache wie Abasic

sehr zeitraubend. Dieser Weg wird bei Programmen, die oft parsen (to parse: engl. zerlegen, analysieren), schlicht unmöglich. Aus der Überlegung, daß man den Computer auch das machen lassen kann, was der Benutzer oft selbst macht (den Programmtext ändern und das Programm neu starten), entstand die Routine „modify“, das Kernstück des Programms. Darin wird das Programm zuerst um diejenigen Zeilen verkürzt, die geändert werden sollen. Nachdem der Programmtext als ASCII-Datei

auf die RAM-Disk abgespeichert ist, wird die Datei geöffnet und die zuvor eingegebenen Zeilen werden angehängt. Schließlich wird das nun mit den neuen Zeilen versehene Programm angestartet und „kickt“ das alte aus dem Speicher. Der schmückende Rest dient dann zur Erzeugung der Menüs, zur Fensterverwaltung und natürlich zum Zeichnen des Graphen – aber das sieht man ja. Das Programm ist gut dokumentiert (viele REM's) und bietet sich zum Abändern an. Viel Spaß bei $y = \sin(x/2)$, $y = \sqrt{x}$, ...!!!



```

REM          **** FUNKTIONSPLOTTER ****
REM          * MIT EINGABE DER FUNKTION *
REM          **** IM PROGRAMM-LAUF ****

ON ERROR GOTO fehler:
REM Variablendeklaration
wsy=179
wsx=619
SCREEN 2,640,200,2,2
WINDOW 1,"plotter",(0,0)-(wsx,wsy),16,2
PALETTE 0,0,0,0
PALETTE 1,1,1,1
PALETTE 2,.93,.2,0
PALETTE 3,0,1,0

REM Erzeugung der Menueleiste
MENU 1,0,1,"Funktion"
MENU 1,1,1,"..ändern"
MENU 1,2,1,"..zeigen"
MENU 2,0,1,"Koordinaten"
MENU 2,1,1,".. zeigen"
MENU 2,2,1,"X-Grenzen ändern"
MENU 2,3,1,"Y-Grenzen ändern"
MENU 2,4,1,"Maßstab fuer Winkelfunktionen"
MENU 3,0,1,"X-Step"
MENU 3,1,1,"16 "
MENU 3,2,1,"4 "
    
```

```

MENU 3,3,1,"1 "
MENU 3,4,1,"1/4 "
MENU 3,5,1,"1/16 "
MENU 4,0,1,"Programm"
MENU 4,1,1,"..beenden"

SUB wart STATIC
REM Warteschleife mit Abfrage des linken Mausknopfes
wartmarke: IF MOUSE(0) >=0 THEN wartmarke
weiter:
END SUB

SUB draw(xma,xmi,xst,ya,yi,yst,wsx,wsy) STATIC
REM Dieses Unterprogramm erstellt das Koordinatenkreuz
REM und erzeugt eine Skalierung
CLS
PRINT "Untere X-Grenze ist ";xmi
PRINT
PRINT "obere X-Grenze ist ";xma
PRINT
INPUT "Abstand zwischen Markierungen:";xmdif
IF xmdif=0 THEN
REM Berechnung eines Standardwertes bei
REM fehlender Eingabe
xmdif=(xma-xmi)/8
END IF
    
```

```

CLS
PRINT "Untere Y-Grenze ist ";ymi
PRINT
PRINT "obere Y-Grenze ist ";yma
PRINT
INPUT "Abstand zwischen Markierungen: ";ymdif
IF ymdif=0 THEN
  REM Standardwert
  ymdif=(yma-ymi)/8
END IF

CLS
REM Berechnung der Position der Y-Achse
yax=-xmi/xst
REM Berechnung der Position der X-Achse
xax=wsy+ymi/yst
IF yax>=0 AND yax<=wsx THEN
  REM Zeichnen der X-Achse
  LINE(0,xax)-(wsx-1,xax),2
  REM Skalenerzeugung
  FOR m= 0 TO xma STEP xmdif
    LINE (yax+(m/xst),xax-2)-(yax+(m/xst),xax+2),2
  REM Positionierung des Textcursors
  IF m>xmdif>xma THEN
    LOCATE((xax)/8),((yax+(m/xst)-20)/8)
    PRINT m ;
  END IF
  NEXT m
  LINE(wsx-5,xax+5)-(wsx-1,xax),2
  LINE(wsx-5,xax-5)-(wsx-1,xax),2

  FOR m= 0 TO xmi STEP -xmdif
    LINE (yax+(m/xst),xax-2)-(yax+(m/xst),xax+2),2
  REM Positionierung des Textcursors
  IF m<-xmdif<xmi THEN
    LOCATE((xax)/8),((yax+(m/xst)+10)/8)
    PRINT m ;
  END IF
  NEXT m
  REM Ende Skalenerzeugung
END IF

IF yax>=0 AND yax<=wsx THEN
  REM Zeichnen der Y-Achse
  LINE(yax,0)-(yax,wsy-1),2
  REM Skalenerzeugung
  FOR m= 0 TO yma STEP ymdif
    LINE (yax-2,xax-m/yst)-(yax+2,xax-m/yst),2
  REM Positionierung des Textcursors
  IF m+ymdif>yma THEN
    LOCATE((xax-m/yst+5)/8),((yax+10)/8)
    PRINT m SPC(0)
  END IF
  NEXT m
  LINE(yax,0)-(yax+5,5),2
  LINE(yax,0)-(yax-5,5),2

  FOR m= 0 TO ymi STEP -ymdif
    LINE (yax-2,xax-m/yst)-(yax+2,xax-m/yst),2
  REM Positionierung des Textcursors
  IF m-ymdif<ymi THEN
    LOCATE((xax-m/yst-10)/8),((yax+10)/8)
    PRINT m SPC(0)
  END IF
  NEXT m
  REM Ende Skalenerzeugung
END IF
END SUB

vorgabe: REM Einstellung der Grenzwerte nach Programmstart
ymin=-10
ymax=10
xmin=-4
xmax=4
ystep=(ymax-ymin)/wsy
xstep=(xmax-xmin)/wsx
xspr=1
GOTO eingabe

eingabe: REM Warten auf Menueanwahl
MENU ON
ON MENU GOSUB auswert
GOTO eingabe
STOP

```

```

auswert: REM Ermittlung des gewählten Menuetitels
w=MENU(0)
REM Ermittlung des gewählten Menupunktes
v=MENU(1)
REM Verzweigung nach Menuetitel
ON w GOSUB funk,koord,xsprung,ausgang

funk: REM Verzweigung nach Menuepunkt
ON v GOSUB modify,anzeig
GOTO eingabe
STOP

koord: REM Verzweigung nach Menuepunkt
ON v GOSUB see,xchange,ychange,wifu
GOTO eingabe
STOP

xsprung: xspr=4^(v-3)
GOTO eingabe

ausgang:
KILL "ram:funktion"
KILL "ram:funktion.info"
SYSTEM
END

see: REM Voreinstellungen anzeigen
WINDOW 3,"zeig",(300,0)-(540,60),0,2
PRINT "untere X-Grenze: ";xmin
PRINT "obere X-Grenze: ";xmax
PRINT "untere Y-Grenze: ";ymin
PRINT "obere Y-Grenze: ";ymax
CALL wart
WINDOW CLOSE 3
GOTO eingabe

xchange: REM X-Grenzen abändern
WINDOW 3,"frag",(0,0)-(300,100),0,2
INPUT "Untere X-Grenze eingeben: ",xmin
PRINT " "
INPUT "Obere X-Grenze eingeben: ",xmax
REM Gueltigkeitspruefung
IF xmax <= xmin THEN
  PRINT " "
  PRINT "Ungueltige Eingabe"
  CALL wart
  GOTO xchange
END IF
xdif=ABS(xmax-xmin)
REM xstep=Differenz zwischen 2 Bildpunkten in horiz. Richt.
xstep=xdif/wsx
WINDOW CLOSE 3
GOTO anzeig

ychange: REM Y-Grenzen abändern
WINDOW 3,"frag",(0,0)-(300,100),0,2
INPUT "Untere Y-Grenze eingeben: ",ymin
PRINT " "
INPUT "Obere Y-Grenze eingeben: ",ymax
REM Gueltigkeitspruefung
IF ymax <= ymin THEN
  PRINT " "
  PRINT "Ungueltige Eingabe"
  CALL wart
  GOTO ychange
END IF
ydif=ABS(ymax-ymin)
REM ystep=Diff. zwischen 2 Bildp. in vert. Richt.
ystep=ydif/wsy
WINDOW CLOSE 3
GOTO anzeig

wifu: REM Grenzwerte-Einstellung fuer trigonometrische Funktionen
ymin=-2
ymax=2
xmin=-6.283
xmax=6.283
ystep=(ymax-ymin)/wsy
xstep=(xmax-xmin)/wsx
xspr=1
GOTO anzeig

```

```

modify: REM Löschen der zu ersetzenden Programmzeilen
DELETE 1000
DELETE 1010
REM Speichern der Programmumpfes als ASCII-Datei
SAVE "ram:funktion",A
OPEN "ram:funktion" FOR APPEND AS #1
LOCATE 1,1
REM Funktionseingabe
INPUT " Eingabe y=" ,A$
A$="1000 y="+A$
REM Ab hier werden die Zeilen an das Programm angehängt
FOR i =1 TO LEN(A$)
PRINT #1,MID$(A$,i,1);
NEXT i
PRINT #1,CHR$(26)
b$="1010 return"
FOR i =1 TO LEN(b$)
PRINT #1,MID$(b$,i,1);
NEXT i
PRINT #1,CHR$(26)
REM Die Veränderung des Programmes ist abgeschlossen
CLOSE #1
REM Das neue Programm wird geladen und automatisch gestartet
SCREEN CLOSE 2
LOAD "ram:funktion" ,r
GOTO eingabe

IF ypos>wsy THEN PSET(xpos,wsy+1),1
IF 0<ypos AND ypos<wsy THEN LINE-(xpos,ypos),3
NEXT x
BEEP
REM Graph fertig
GOTO eingabe

fehler:
IF ERR=53 THEN RESUME NEXT
IF ERR=11 AND ERL=1000 THEN RESUME NEXT
IF ERR=5 AND ERL=1000 THEN RESUME impos
END

impos: LOCATE 2,2
PRINT "ungültiges Argument"
PRINT "nach neuer X-Begrenzung 'Funktion zeigen' anwählen"
CALL wart
CLS
WINDOW 1
WINDOW OUTPUT 1
GOTO xchange

REM Hier folgen die Zeilen, die vom Programm
REM verändert werden
funktion:
1000 y=1/SIN(x)
1010 RETURN

anzeigen: REM Routine zum Anzeigen der Funktion
CLS
REM Abrufen des Koordinatenkreuzes
CALL draw(xmax,xmin,xstep,ymax,ymin,ystep,wsx,wsy)
REM Graph zeichnen
x=xmin
GOSUB funktion
PSET(0,(ymax-y)/ystep),3
FOR x= xmin TO xmax STEP xstep/xspr
xpos=(x-xmin)/xstep
GOSUB funktion
ypos=(ymax-y)/ystep
REM Ueberpruefung ob Punkt innerhalb des Fensters
IF ypos<0 THEN PSET(xpos,-1),1

```



Bis zum Exitus.

Operieren mit dem SURGEON



Für Commodore AMIGA 500/1000/2000 · DM 149,- exklusiv nur bei:
 PDC GmbH · Luisenstraße 115 · 6380 Bad Homburg · Telefon 061 72-247 48



Sound Sampling für den AMIGA:

*Future Sound von Applied
Visions und Mimetics
Sound Scape im Test*



Um die Tonerzeugungsmöglichkeiten, die im Amiga eingebaut sind, nutzen zu können, benötigt man ein wenig zusätzliche Hardware. Im Amiga sind nur Digital/Analogwandler eingebaut, mit denen man digitalisierte Klänge ausgeben kann. Wie das funktioniert, können Sie in ei-

nem anderen Artikel dieses Heftes nachlesen. Ein Analog/Digitalwandler, der das Wandeln eigener Klänge erlaubt, ist jedoch leider nicht vorhanden. Glücklicherweise gibt es aber von verschiedenen Herstellern die nötigen Geräte. In diesem Test sollen der Analog/Digitalwandler

von Applied Visions, der zusammen mit der dazugehörige Software den Namen FutureSound trägt, sowie ein Konkurrenzprodukt der Firma Mimetics, daß sich SoundScape nennt, betrachtet werden.

Zuerst zu FutureSound:

Geliefert wird eine Diskette mit Software und ein kleines, graues Gehäuse, an dem ein Flachbandkabel für den Anschluß an den Amiga angebracht ist. Außerdem wird eine recht ausführliche englische Anleitung mitgeliefert. Natürlich ist auch der Programmtext in Englisch gehalten. Wohl der Vollständigkeit halber wird auch ein Mikrofon mitgeliefert; Man darf es aber wirklich nur als Dreingabe betrachten: Seine Qualität ist bestenfalls erbärmlich. Na ja, zum Ausprobieren des Samplers reicht es immerhin.

Der Sampler verlangt die Kickstart-Version 1.0. Deshalb kann das Gerät nur am Amiga 1000 betrieben werden. Außerdem wird das Gerät an der Druckerschnittstelle angeschlossen, von der aus es auch mit Strom versorgt wird. Ein Anschluß an Amiga 2000 und Amiga 500 kommt auch deshalb nicht in Frage, da ja an Pin 23 der Druckerschnittstelle keine 5V-Spannung mehr anliegt. Die parallele Schnittstelle ist übrigens durchgeschleift; mit einem kleinen, beleuchteten Druckknopf an der Frontplatte des Gerätes kann zwischen Drucker und Sampler umgeschaltet werden, so daß man nicht dauernd

Kabel umstecken muß, um mit dem Gerät zu arbeiten.

An der Frontplatte finden sich außer dem Umschalter noch ein Regler für die Eingangsempfindlichkeit sowie zwei Buchsen: Eine Mini-Klinkenbuchse, in die die meisten Billig-Mikrofone passen und eine Cinch-Buchse, über die man den Wandler mit so ziemlich allen anderen Geräten koppeln kann.

Jetzt aber zur mitgelieferten Software:

Das FutureSound-Programm dient dazu, bis zu vier verschiedene Klänge aufzunehmen und diese über die Tongeneratoren des Amiga abzuspielen. Die Länge der digitalisierten Klänge hängt nur vom Speicherausbau des Rechners und von der Samplingrate, die sich einstellen läßt, ab. Klänge lassen sich mit dem Programm im IFF-Format abspeichern, so daß man sie in andere Musikprogramme einladen und dort verwenden kann. Das ist sehr nützlich, da es so möglich ist, andere Sounds als die bei diesen Programmen mitgelieferten zu verwenden.

Wie gehen solche Aufnahmen vor sich?

FutureSound simuliert praktisch die Funktionen eines Tonbandgerä-

tes, nur daß man die Klänge nicht auf Band, sondern in den Speicher des Computers aufnimmt. Vier 'Bandspuren' stehen zur Verfügung, die gleichzeitig abgehört werden können. Während einer Aufnahme kann allerdings nicht abgehört werden, was sich auf den anderen Spuren befindet; das ist auch schon der größte Nachteil des FutureSound-Systems.

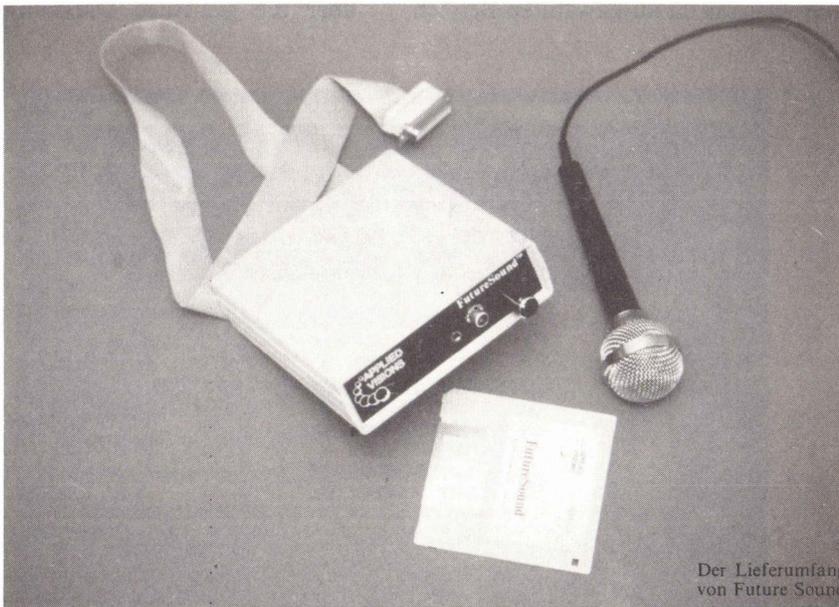
Nach dem Laden eröffnet das Programm einen eigenen virtuellen Bildschirm. Am oberen Rand finden sich die üblichen Gadgets für Vorder- und Hintergrund sowie für das Schließen des Bildschirms, was gleichzeitig das Verlassen von FutureSound bedeutet. Leider fehlt eine Sicherheitsabfrage, die davor warnt, daß bei Betätigen dieses Gadgets alle mühsam erarbeiteten Sounds verloren gehen, falls sie nicht vorher auf Disk abgespeichert wurden. In der Mitte des obersten Bildschirmbereiches zeigt das Programm den freien Speicher an.

Das mittlere Drittel des Schirms wird von einer großen Anzeige eingenommen, die zur grafischen Darstellung der Aufnahme auf jeweils einer Spur dient. Man kann damit sehr schön die unterschiedlichen Charakteristika von Geräuschen erkennen. Rechts von dieser Anzeige befindet

sich ein Schieber, mit dem man Teile des Samples vergrößern kann. Darüber kann man mit vier großen Knöpfen die Spur, die angezeigt und bearbeitet werden soll, auswählen. Unter der grafischen Anzeige sieht man Tasten, die sehr an einen Kassettenrekorder erinnern. Tatsächlich wird mit diesen Tasten die Arbeitsweise eines echten Bandgerätes genau simuliert. Ganz links ist die 'Play'-Taste angebracht, mit der man eine Aufnahme abspielen kann, daneben die 'Record'-Taste, die eine Aufnahme auf der gerade gewählten Spur startet. Die nächste Taste stoppt das Abspielen. Mit 'Rewind' und 'Fast Forward' kann man eine längere Aufnahme an eine beliebige Stelle spulen. Damit man auch etwas finden kann, gibt es ein 'Band'-Zählwerk, an dem sich die aktuelle Position ablesen läßt. Eine Taste mit der Aufschrift 'Step' schließlich schaltet die grafische Anzeige eine Seite vor oder zurück, je nachdem, in welche Richtung zuvor gespult wurde.

lungen, die das Abspielen betreffen, vornehmen kann. Für jede der Spuren ist ein eigenes Feld vorhanden. Spuren können zuallererst einen Namen bekommen. Das macht die Sache sehr Übersichtlich. Ein Schieber dient zum Einstellen der Abspiellautstärke. Für jede einzelne Spur kann die Geschwindigkeit der Wiedergabe eingestellt werden, wobei sich, wie bei einem Tonbandgerät, natürlich auch die Tonhöhe der Aufnahme ändert (Die Abtastwerte – siehe den Sampling-Artikel in diesem Heft – werden, wenn man die Geschwindigkeit erhöht, schneller ausgegeben, als sie aufgenommen wurden. Das bedeutet eine höhere Frequenz, also Tonhöhe. Beim langsameren Abspielen hört man genau den umgekehrten Effekt. Dabei verstärken sich allerdings aus verschiedenen Gründen diverse 'Schmutzeffekte' unangenehm.). Nimmt man beispielsweise eine Stimme auf und läßt sie erheblich schneller abspielen, so erhält man einen wunderhübschen Micky-Maus-Effekt.

Kassettenrekorder muß man auch den FutureSound-Analog/Digitalwandler möglichst genau aussteuern. Ist das Eingangssignal zu leise, ist der Rauschanteil sehr hoch, übersteuert man aber den Eingang, hört man starke Verzerrungen. Im Vergleich zu einem Kassettenrekorder ist ein digitaler Wandler wie das FutureSound-Gerät sogar erheblich empfindlicher gegen Übersteuerungen. Deswegen bietet die Software auch eine Aussteuerungsanzeige. Hier wird der Pegel des Eingangssignals angezeigt, eine besonders große 'Leuchte' zeigt Übersteuerungen an. Mit Hilfe dieser Anzeige und mit dem Regler auf der Vorderseite des FutureSound-Gehäuses kann man recht ordentlich aussteuern. Unter dieser Anzeige finden sich drei Felder, mit deren Hilfe man die Sampling-Rate (Siehe Grundlagenartikel in diesem Heft) bzw. die Aufnahmezeit für die Aufnahme einstellen kann. Daraus, daß diese Anzeigen nur einmal vorhanden sind, kann man schließen, das nicht mehrere Spuren gleichzeitig aufgenommen werden können. Das ist tatsächlich so. Die eingestellten Werte gelten immer für die Spur, die mit den Auswahlknöpfen am oberen Bildrand gerade ausgewählt ist. Je höher die Sampling-Rate, desto öfter wird das Eingangssignal pro Sekunde gemessen. Stellt man hier einen Wert von 10000 ein, werden pro Sekunde 10000 Messungen des Klanges gemacht und gespeichert. Für diese eine Sekunde benötigt man 10KByte Speicher, da jeder Meßwert eine Genauigkeit von 8 Bit besitzt. Die maximal zur Verfügung stehende Zeit hängt nur vom Hauptspeicher Ihres Rechners ab. Leider konnte man bei unserem Rechner, trotz Speichererweiterung auf 1,5 Megabyte, nur maximal 512 KByte Speicher nutzen. Offensichtlich ist das Programm nicht in der Lage, den darüber hinaus eventuell verfügbaren Speicher zu erkennen und zu gebrauchen. Schade! Wenn man die Sampling-Rate gewählt hat, kann man mit den beiden anderen Einstellern entweder die Zeit für die Aufnahme oder die Speichergröße, die für die Spur zur Verfügung stehen soll, einstellen. Dabei merkt das Programm, wenn man versucht, mehr Speicher zu verbrauchen als vorhanden ist und beschwert sich bei Ihnen.



Unter dem Zählwerk finden sich zwei weitere Anzeigen, mit denen man zwei beliebige Punkte innerhalb einer Aufnahme markieren kann. Der Bereich zwischen diesen beiden Punkten kann dann mit einigen interessanten Optionen, auf die wir später noch zu sprechen kommen wollen, bearbeitet werden.

Den größten Teil der unteren Bildschirmfläche nimmt ein Block ein, in dem man die eine Reihe von Einstel-

Aus dem 'Miau' einer Katze kann man schöne brüllende Tiger machen. Ansonsten kann man jede Spur an- und ausschalten. Außerdem besteht die Möglichkeit, jede Spur in einen sogenannten 'Loop'-Modus zu schalten: Diese Spur wird dann pausenlos wiederholt.

Die wichtigsten Anzeigen und Einsteller für Aufnahmen findet man auf der rechten Bildschirmseite unter den Zählwerken: Wie jeden guten

Aber jetzt ans Werk! Um etwas aufzunehmen, steuern Sie zuerst einmal aus. Das funktioniert, wie gesagt, wie bei einem Kassettenrekorder. Stellen Sie nun die gewünschte Aufnahmezeit ein und drücken Sie die 'Record'-Taste. Alles, was Sie während der gewählten Aufnahmezeit ins Mikrofon sprechen, wird digital aufgezeichnet. Nach der Aufnahme sehen Sie in der großen Anzeige, wie Ihre Aufnahme aussieht. Mit der Play-Taste können Sie Ihre Aufnahme wieder abhören. Sie können noch drei weitere Aufnahmen machen und diese dann zusammen abhören. Leider kann man aber während der Aufnahme die anderen Spuren nicht abhören. Sonst könnten Sie zum Beispiel auch mit sich selbst im Chor singen... Wirklich zu schade, daß man bei Applied Visions diese Möglichkeit nicht vorgesehen hat. Der Nutzen der vier Spuren wird dadurch sehr stark begrenzt, da es doch großer Mühe bedarf, selbst Geräusche auf zwei Spuren synchron aufzunehmen, von Musik (zum Beispiel Chorsingen) ganz zu schweigen.

FutureSound bietet Ihnen aber noch einige weitere Funktionen, die wirklich lustige Spiele mit Klängen erlauben. Wenn man die rechte Maustaste betätigt, erscheint eine Menüleiste. Dort findet man zuerst einmal ein 'Disk'-Menü, das das Speichern und Laden von Stücken auf Disk erlaubt. Dabei besteht die Möglichkeit, eine Spur entweder in einem FutureSound-eigenen Format abzulegen oder im Standard-IFF-Format, das auch von anderen Programmen verstanden werden kann. Leider ist es nicht möglich, alle Spuren zusammen abzuspeichern. Das 'Track'-Menü bietet einige Funktionen, mit denen man Tracks löschen und die Abspielparameter wieder auf normale Werte setzen kann. So richtig spannend wird es aber im dritten Menü, das den vielversprechenden Titel 'Bag of Tricks' trägt. Mit den Funktionen unter diesem Menütitel können Klänge auf verschiedene Weise bearbeitet werden. Dabei wirkt die Funktion nur in dem Bereich, der mit den beiden Zählwerken, wie oben erwähnt, eingestellt wurde. Man muß also nicht immer ganze Spuren bearbeiten, auch Teile können ausgewählt und verändert werden. Aber nun zu den einzelnen

Funktionen:

Reverse Sound: Der ausgewählte Bereich der Spur wird umgedreht, also von hinten nach vorne abgespielt. Viele Geräusche, oder auch Sprache, werden wirklich sehr mysteriös, wenn man sie auf diese Weise verändert.

Copy Sound: Der gewählte Abschnitt einer Spur wird ab der aktuellen Zählwerkposition in eine andere Spur hineinkopiert. Damit kann man einen Klang aus verschiedenen Teilen zusammensetzen.

Mix Sound: Hier werden die Klänge nicht hintereinanderkopiert, wie bei der 'Copy'-Funktion, sondern übereinandergelegt und gemischt. Die alte Spur kann dann wieder neu überspielt werden. Leider muß man bei dieser Funktion sehr darauf achten, daß sich keine Übersteuerungen ergeben. Man muß die beiden Spuren, die gemischt werden sollen, eigentlich immer vorher mit der 'Scale Sound'-Option (siehe unten) bearbeiten.

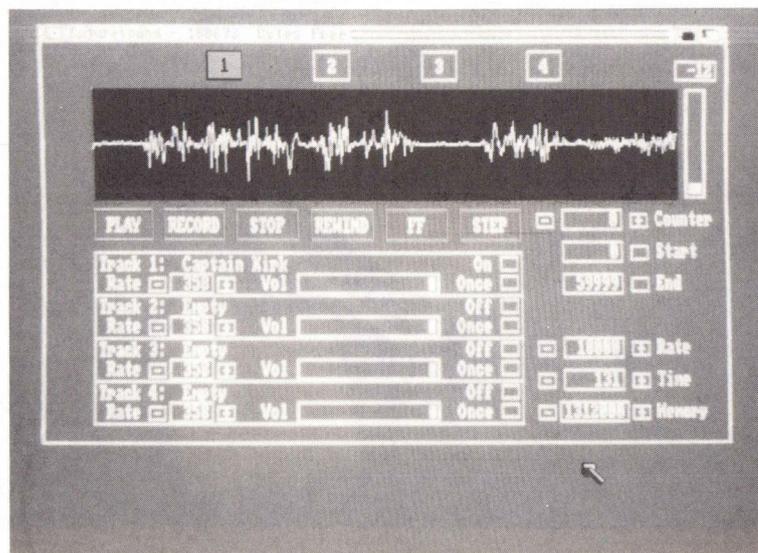
Zero Sound: Damit wird der gewählte Abschnitt eines Klanges gelöscht.

Scale Sound: Mit dieser Option kann die Lautstärke eines Klanges im ausgewählten Bereich

naueres erklärt, dafür sind jedoch gut kommentierte Beipielprogramme vorhanden, die bei einigen Vorkenntnissen keine Verständnisschwierigkeiten bieten dürften.

FutureSound erlaubt es, die Soundfähigkeiten des Amiga hervorragend auszunutzen. Man darf von dem System natürlich keinen professionellen Klang erwarten, die 8-Bit-Auflösung und auch die maximale Sampling Rate (28000 Hz) sind dazu zu gering; die Audio-Hardware des Amiga limitiert den Frequenzgang jeder Tonausgabe darüberhinaus sogar auf maximal 7 KHz.

Ich sehe die Hauptanwendung für FutureSound bei den Programmierern, denen es mit dem Gerät leicht wird, ihre Programme, vor allem Spiele, mit zeitgemäßem Digital-Sound zu versehen. Außerdem kann man mit dem Programm eine Menge lustiger Dinge mit Klängen tun und auch die Klangbibliothek für Musik-Programme, die das IFF-Klangformat verstehen (z.B. Aegis Sonix), erweitern. Darüber hinaus kann man über die grafische Darstellung ein



verändert werden. Damit kann man natürlich Übersteuerungen, die während der Aufnahme geschehen sind, nicht reparieren.

Auf der FutureSound-Diskette finden sich, außer dem Programm und einigen Beispiellängen, eine Reihe von Files, die das Einbinden von gesampelten Klängen in eigene C- oder AmigaBasic-Programme erlauben. Im Handbuch wird dazu kaum etwas ge-

wenig Verständnis für das Phänomen 'Klang' erwerben. Ob die Möglichkeiten des Gerätes den Preis von 399,- DM rechtfertigen, ist eine Frage, die sich der mögliche Kunde selbst stellen muß. Derjenige, der seine eigenen Programme mit einem richtig guten Sound versehen will, sollte sich FutureSound unbedingt näher anschauen. Auch die wildesten Klangeffekte können leicht produ-

ziert werden, Bearbeitungen von Sprache sind, bei Verwendung eines besseren Mikrofons und sorgfältiger Aussteuerung, sogar in recht guter Qualität möglich. Für alle anderen ist das Gerät ein schönes und unterhaltsames, darüber hinaus auch noch lehrreiches Spielzeug. Obwohl es dafür nicht gerade billig ist, halte ich es für empfehlenswert.

Doch jetzt zur Konkurrenz:

Der Analog/Digitalwandler von Mimetics ist erheblich kleiner als das Applied Visions-Gerät. Es ist ein kleines, unscheinbares Blechkästchen, das nicht an der Druckerschnittstelle, sondern am zweiten Joystick-Port angeschlossen wird. Mitgeliefert wird eine Diskette mit Software und eine Anleitung, die aus zwei Teilen besteht. Programm und Anleitung sind selbstverständlich auf Englisch. Der zweite Teil der Anleitung ist offensichtlich eine Erweiterung zum ersten Teil (der sich anscheinend mit einer älteren Programmversion befaßt), die erst nachträglich hinzugekommen ist; die Übersichtlichkeit leidet sehr darunter. Man muß immer nach der gewünschten Information suchen, zumal vieles im ersten Teil des Handbuches offensichtlich bei der vorliegenden Programmversion völlig unzutreffend ist. Dem völlig unerfahrenen Benutzer könnte das Handbuch den Einstieg in SoundScape sehr erschweren.

Die SoundScape-Software öffnet keinen eigenen virtuellen Bildschirm wie FutureSound, sondern benutzt bis zu drei Fenster, in denen sich alle Funktionen bedienen lassen. Eine Menüleiste gibt es nicht.

Das Konzept des Programms unterscheidet sich erheblich von dem des FutureSound-Samplers. SoundScape erlaubt grundsätzlich nur die Aufnahme eines einzelnen Sounds, der jedoch für jede von zehn möglichen Oktaven separat aufgenommen werden kann. Ein einzelnes Sample läßt sich also immer nur über den Bereich einer Oktave verwenden. Über die obersten beiden Tastenreihen, die knapp zwei Oktaven einer Klaviatur simulieren, kann eine Aufnahme gespielt werden. Das Prinzip des Samplers ist also nicht das eines Tonbandgerätes, wie bei FutureSound,

sondern mehr das eines Sampling-Keyboards. In unserem Grundlagenartikel in diesem Heft können Sie näheres über das Prinzip eines solchen Instrumentes nachlesen. Zusammenfassend sei an dieser Stelle nur erwähnt, daß ein digitalisierter Klang höher oder tiefer wiedergegeben werden kann, indem man die Sampling-Rate bei der Wiedergabe gegenüber der Aufnahme rate verändert. Da dies bei zu großen Entfernungen von der Originalrate zu starken Klangverfälschungen führt, muß der gesamte Tonumfang eines Instrumentes auf bis zu 10 Oktaven verteilt werden. Natürlich ließe sich die Klangqualität noch verbessern, wenn man statt der Oktaven auch Quinten oder gar Quartan als größtes Intervall verwenden könnte. Das ist in der SoundScape-Software aber leider nicht vorgesehen. Natürlich muß man nicht jede der zehn Oktaven mit Aufnahmen des gleichen Instrumentes in verschiedenen Tonhöhen füllen. Es ist genauso möglich, die ersten beiden Oktaven zum Beispiel mit einer Baßgitarre, die nächsten beiden mit einem E-Piano und die übrigen mit diversen Schlaginstrumenten zu füllen. In diesem Fall könnte man auf einer entsprechend großen Tastatur alle diese Instrumente auf einem Keyboard spielen. Hier zeigt sich bereits eine der Stärken des SoundScape-Konzeptes: Es ist möglich, eine Midi-fähige Tastatur an den Amiga anzuschließen und die gesampelten Sounds über diese zu spielen. Dabei reagiert der Computer, eine geeignete Klaviatur vorausgesetzt, sogar auf Anschlagsdynamik und Pitchbending (stufenloses Verstimmen von gespielten Tönen, das mit Saiten- und Blasinstrumenten möglich ist, den Spielern von Tasteninstrumenten aber bis zur Einführung elektronischer Instrumente unmöglich war). Tatsächlich wird Ihr Amiga im Zusammenspiel mit einer Midi-Tastatur und SoundScape fast zu einem richtigen Sampling-Keyboard, dessen Klangmöglichkeiten und -qualitäten zwar noch nicht mit 'richtigen' Musikinstrumenten zu vergleichen sind, wohl aber das Einarbeiten in Funktionen und Möglichkeiten solcher Geräte ermöglichen. Noch sind die PCs dieser Preisklasse offensichtlich nicht in der Lage, die, zugegebener-

weise auch etwas teureren, professionellen Musikinstrumente zu ersetzen.

Betrachten wir nun die Bedienung der SoundScape-Software:

Nach dem Laden erscheinen zunächst nur zwei der maximal drei Bedienungsfenster auf dem Workbench-Bildschirm. Das erste ist sehr klein und für die Bedienung auch völlig unwichtig, es enthält lediglich das Abbild einer Klaviatur und die Tasten der Amiga-Tastatur, mit denen sich diese Klaviatur simulieren läßt. Es ist also ein reines Hilfsfenster. Das größere Fenster füllt ungefähr zwei Drittel des Bildschirms und dient zur Bearbeitung von aufgenommenen Samples sowie zur Verwaltung der Diskettenlaufwerke. Am oberen Rand des Fensters befinden sich die vier Wörter 'Load', 'Save', 'Sample' und 'Clear'. Durch Anklicken des entsprechenden Wortes mit der Maus wird das Kommando, das sich hinter dem Wort verbirgt, angewählt. Die Funktion von 'Load' und 'Save' muß wohl nicht näher erläutert werden. Wichtig ist beim 'Save'-Kommando nur, daß ähnlich wie bei FutureSound zwei Formate zur Verfügung stehen. Ein aufgenommener Klang kann entweder im IFF-Format oder in einem Mimetics-Spezialformat auf Diskette abgelegt werden. Das Mimetics-Format hat den Vorteil, daß Klänge so abgelegt werden, wie sie sich im Speicher befinden, mit allen Daten, die die Einstellung von Pitchbending und Anschlagsdynamik betreffen. Das ist mit dem IFF-Format nicht möglich, weil diese Sonderfunktionen im IFF-Standard nicht vorgesehen sind. Man braucht das IFF-Format eigentlich nur, um mit SoundScape erstellte Sounds in anderen IFF-kompatiblen Programmen zu verwenden. Die 'Clear'-Funktion löscht den gesamten Speicher und besitzt deswegen eine Sicherheitsabfrage. Schließlich könnte man sich ja das Werk mehrerer Stunden zerstören... Die 'Sample'-Funktion schließlich öffnet das dritte Fenster, das SoundScape zur Verfügung stellt. In diesem Fenster befinden sich (fast) alle Funktionen, die für das Aufnehmen eigener Klänge erforderlich sind. Hier zeigt sich auch ein Nachteil der

SoundScape-Software gegenüber FutureSound: Eine Aussteuerungsanzeige existiert leider nicht, man muß die Einstellung der Empfindlichkeit, die übrigens per Software erfolgt, also auf Verdacht vornehmen. Um die dabei häufig vorkommenden Übersteuerungen zu vermeiden, haben die Programmautoren eine automatische Aussteuerung eingebaut. Sie funktioniert auch recht gut, ist aber bei Klängen, deren Lautstärke sich stark ändert, kaum zu gebrauchen. Die Empfindlichkeit wird mit zwei Reglern eingestellt; einer dient zur Einstellung der Empfindlichkeit, der andere stellt eine Mindestlautstärke ein, die erreicht werden muß, bevor die Aufnahme beginnt. Um etwas aufzunehmen, muß man zuerst im Hauptfenster das unter dem Wort 'Sample' befindliche Icon anklicken und auf den gewünschten Wert stellen. Dieser Wert kann entweder 'High' oder 'Low' sein und bedeutet die Samplingrate, die im Gegensatz zu FutureSound nicht kontinuierlich, sondern nur in zwei Stufen (eben 'High' und 'Low') eingestellt werden kann. In Stellung 'High' beträgt die Samplingrate ungefähr 14 KHz, in Stellung 'Low' ist es genau die Hälfte, also um 7 KHz. Danach öffnet man das Sampling-Fenster durch Betätigen der 'Sample'-Taste. Dort muß nun, wie beschrieben, der gewünschte Aufnahmemodus (automatische oder manuelle Aussteuerung) eingestellt werden. Dazu dient ein Icon, dessen Aussehen durch Anklicken verändert werden kann. Natürlich muß man auch auswählen, in welcher der zehn möglichen Oktaven die Aufnahme erfolgt. Dazu dient eine Reihe von Zahlen am linken unteren Fenstertrand. Die angewählte Oktave wird weiß dargestellt. Aber das ist noch nicht alles: Der Computer muß auch wissen, wie lang das Sample werden soll. Dazu gibt es einen mit 'Length' betitelten Schieberegler, der die Größe des für das Sample zu verwendenden Speichers angibt. Unter diesem Schieberegler wird die Länge in Sekunden angezeigt. Maximal 65KByte Speicher können für eine Oktave verwendet werden. Das entspricht, je nach Sampling Rate, zwischen 4,5 und 9 Sekunden zeitlicher Länge. Diese Länge können Sie natürlich nur dann nutzen, wenn Ihr Amiga



Die Hardware von SoundScape

über ausreichenden Speicher verfügt. Zu guter letzt müssen Sie noch die Empfindlichkeit und die Mindestlautstärke einstellen, dann kann die Aufnahme beginnen.

Dazu muß man das mikrofonförmige Icon am linken Fensterrand anklicken. Der Bildschirm wird völlig rot, aber nur für zwei Sekunden. Dann wird er gelb. Das bleibt er solange, bis sie den linken Mausknopf zweimal betätigen oder bis Sie den Knopf einmal betätigen und das aufzunehmende Signal die Mindestlautstärke überschreitet (Sie können die eingestellte Mindestlautstärke also durch den Doppelklick umgehen und SoundScape zu einer Aufnahme zwingen). In diesem Moment wird der Schirm grün, und das bleibt er, bis die eingestellte Aufnahmedauer erreicht ist. Danach schaltet der Computer die Ampelfarben wieder ab und der normale Workbenchescreen erscheint. Im 'Sample'-Fenster können Sie nun eine grafische Darstellung des aufgenommenen Signals bewundern, die Sie wie bei FutureSound vergrößern und verkleinern können. Mit einem weiteren Schieberegler können Sie den Teil des Samples wählen, den Sie gerade betrachten wollen.

In die grafische Darstellung Ihrer Aufnahme können Sie mit der Maus beliebige Veränderungen hineinmalen. Sie könnten also auch Stille aufnehmen und hinterher einfach Ihre künstlerischen Gelüste durch Anhören gemalter Klänge befriedigen. Es ist allerdings sehr schwierig, den

Klang solcher Malereien vorherzuahnen. Meistens wird man diese Option dazu verwenden, um eventuell in die Aufnahme hineingeratene Störgeräusche, die man meist recht gut optisch von dem eigentlichen Klang unterscheiden kann, herauszumalen. Im Sample-Fenster stehen aber auch noch andere Editierkommandos zur Verfügung. Mit einem Icon kann man ein Sample unverändert in eine andere Oktave hineinkopieren. Bevor man zum Beispiel eine Störung per Maus wegmalen will, sollte man sicherheitshalber eine Kopie des Samples in einer anderen Oktave machen. Ein anderes Icon kopiert den Sound in eine andere Oktave, rechnet ihn aber dabei, je nach Richtung, auf doppelt so hohen oder tiefen Klang um. Natürlich führt diese Umrechnung nicht zu natürlichen Klängen. Schließlich ist es noch möglich, einen bestimmten Bereich aus einem Sample auszuschneiden. Oft nimmt man sicherheitshalber einen zu großen Speicherbereich für ein Sample, von dem man nur einen kleinen Teil benötigt. Diesen gewünschten Teil kann man sich nun ausschneiden. Der dabei freiwerdende Speicher wird an das Betriebssystem zurückgegeben und kann anderweitig verwendet werden. Die letzte Edit-Funktion im 'Sample'-Fenster ist die wichtigste und nützlichste. Deswegen ist sie wahrscheinlich auch doppelt vertreten, sowohl im 'Sample'- wie im Hauptfenster. Viele Klänge, zum Beispiel von Streichern oder Orgel, sollen länger spielbar sein, als man sie aufnehmen kann. Wenn nur 4,5 Sekunden Sampling-Zeit zur Verfügung stehen, kann kein Ton länger klingen als 4,5 Sekunden. Das ist natürlich sehr ärgerlich. Was tun? Ein Blick auf die grafische Darstellung eines solchen Klanges zeigt die Lösung: An Anfang und Ende der meisten Klänge passiert schrecklich viel, was für diesen Klang wichtig und charakteristisch ist. In der Mitte finden sich aber häufig lange Teile, in denen sich immer die gleichen Wellenformen nahezu unverändert wiederholen. Was liegt also näher, als einen solchen Ausschnitt eines Klanges solange zu wiederholen, wie der entsprechende Klang gespielt werden soll? Diese Technik wird von allen Musikcomputern und Sampling-keyboards verwendet. Das Problem

dabei ist eigentlich nur, geeignete 'Loop-Points', also Anfangs- und Endpunkte für diese Abspielschleife zu finden. Mit SoundScape ist es möglich, für jeden Klang solche Loop-Points manuell einzustellen. Man muß zwar meistens ziemlich lange probieren, bis man einen halbwegs unhörbaren Loop zustandebringt, aber dieses Feature ist wirklich sehr nützlich.

Im Hauptfenster von SoundScape finden sich noch einige weitere nützliche Möglichkeiten: Es ist möglich, die digitalisierten Klänge in einem Bereich von plus/minus einem Halbton zu stimmen. Das ist notwendig, um mit anderen Instrumenten zusammenspielen zu können. Darüber hinaus können die Informationen, die von einem angeschlossenen Midi-Keyboard kommen, transponiert werden. Das heißt, man kann auf dem Keyboard in C-Dur spielen, obwohl die klingende Tonart Es-Dur oder noch etwas unangenehmeres ist. Um aufgenommene natürliche Klänge verfremden zu können, bietet SoundScape die Möglichkeit, sie mit einer sogenannten Hüllkurve zu versehen. Eine Hüllkurve ist nichts anderes als ein Verlauf, der einem beliebigem Parameter eines Signals 'aufgedrängt' wird. Man kann mit einer Hüllkurve die Frequenz eines Signals beeinflussen oder auch den Lautstärkeverlauf. Im Falle von SoundScape ist es möglich, den Lautstärkeverlauf des Signals zu verändern. Die Hüllkurven, die SoundScape bietet, bestehen aus vier Phasen wie die 'klassischen' Hüllkurven älterer Analogsynthesizer. Man bezeichnet diese Art Hüllkurven mit 'ADSR' für 'AttackDecay-Sustain-Release', oder auf Deutsch: 'Anstiegs-, Abfall-, Halte- und Ausklingphase'. In der ersten Phase steigt die Lautstärke, bei Null beginnend, bis zu einem einstellbaren Wert an. Die Länge der Phase ist einstellbar. In der zweiten Phase fällt oder steigt die Lautstärke des Signals ebenfalls bis auf einen bestimmten Level, wobei auch hier die Zeit einstellbar ist. Nach Ablauf der zweiten Phase wird in einstellbarer Zeit die Haltephase, in der die Lautstärke des Tons konstant bleibt, bis die Taste losgelassen wird, erreicht. Nach Loslassen der Taste klingt der Ton in einstellbarer Zeit aus. Diese Art von Hüllkurven

hat sich als einfache Näherung an das Verhalten natürlicher Instrumente bewährt. Natürlich sind auch sehr ausgefallene Effekte bei entsprechender Einstellung möglich. Im Hauptfenster wird der eingestellte Kurvenverlauf mit einer großen Grafik dargestellt, die Werte können mit Schiebereglern verändert werden. Dabei sind die Phasen von 1 bis 4 nummeriert, wobei Lautstärkereglern mit 'L1' oder 'L2', Zeiteinstellungen mit 'R1' usw. bezeichnet sind. Mit Hilfe dieser komfortablen Hüllkurven ist es problemlos möglich, aus natürlichen Klängen typische Synthesizerklänge zu erzeugen. Dadurch wird die Palette der Klangmöglichkeiten natürlich erheblich erweitert.

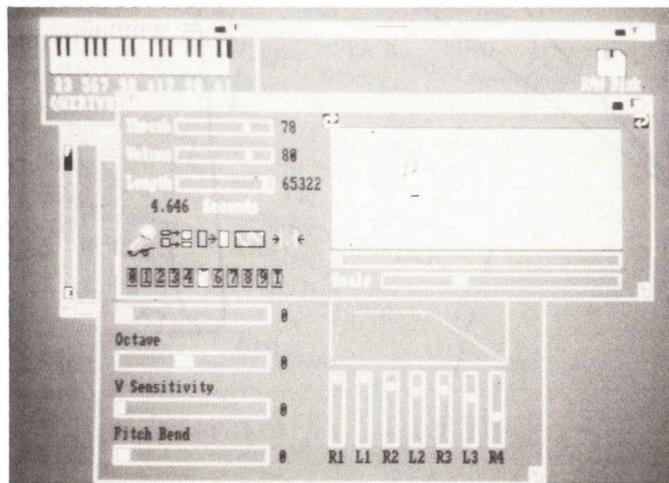
Im SoundScape-Paket ist keine Software zur Einbindung von gesampelten Klängen in eigene Programme enthalten. Wer allerdings FutureSound besitzt, dürfte keine Schwierigkeiten haben, die dort angegebenen Routinen auch für SoundScape-Klänge zu benutzen, obwohl dabei möglicherweise nicht alle Features von SoundScape nutzbar sind.

Für 229,- DM leistet SoundScape eine ganze Menge. Es bietet andere Möglichkeiten zur Klangbeeinflussung als FutureSound, so daß man in diesem Punkt, auch des völlig anderen Konzept wegen, kein Programm dem anderen uneingeschränkt vorziehen kann. SoundScape ist in seinen Funktionen mehr an einen Synthesizer angelehnt, über ein Midi-Keyboard kann es sogar wirklich wie ein Synthesizer-Modul verwendet werden, während FutureSound im wesentlichen ein elektronisches Tonbandge-

rät simuliert. Deshalb ist SoundScape für einen anderen Benutzerkreis interessant als FutureSound. Nur in einem Punkt fallen die Anwendungsbereiche zusammen: Beide Sampler erlauben es, Klänge für andere Musikprogramme, die IFF-Files lesen können, zu erstellen. Für diese Aufgabe sind beide Systeme gleich gut geeignet, FutureSound bietet hier sogar etwas mehr, weil viele Spezialeffekte von SoundScape über das IFF-Format nicht mitübertragen werden können. Dafür ist SoundScape um einiges billiger. Ansonsten ist FutureSound besonders gut dafür geeignet, Geräuscheffekte für eigene Programme zu produzieren. Die eingebauten Effekte sind für diese Aufgabe besonders brauchbar. Dafür bietet SoundScape zusammen mit einem Midi-Keyboard die Möglichkeit, erste Erfahrungen mit einem Sampling-Keyboard zu sammeln, auch wenn der Amiga das Keyboard natürlich nicht ersetzen kann. Wer aber nur einmal in das Gebiet des Samplings und seinen Einsatz im Bereich Synthesizer hineinriechen will, für den ist SoundScape ein geeignetes Programm.



So hat jedes der beiden Systeme seine besonderen Leistungen. Wer sich einen Sampler für den Amiga zulegen will, sollte sich gut überlegen, was er damit tun will – und sich dann SoundScape und FutureSound, die beide in ihrem Konzept empfehlenswert sind, genauer anschauen.



Verschiedene Fenster für verschiedene Aufgaben

A I N B

Autoren gesucht

Die KICKSTART-Redaktion sucht Programmierer, die sich mit dem AMIGA auskennen und darüber zu berichten wissen, sei es über Software oder Hardware.

Wenn Sie eigene Programme geschrieben haben, die Sie gerne veröffentlichten würden, so lassen Sie sie nicht in der Schublade verschwinden, sondern schicken sie uns!

Neben Programmlistings suchen wir auch Tips & Tricks über den AMIGA. Angefangen bei Assembler bis hin zu Basic, C, Pascal oder anderen Sprachen. Auch die Bastler sind aufgerufen, über ihre Fachgebiete (z.B. Floppies, Drucker, Harddisk, Monitor) zu berichten. Das Honorar erfolgt nach Vereinbarung.

Über Ihre Beteiligung würden wir uns sehr freuen.

KICKSTART-Redaktion
'Programmeinsendung'
Schwalbacherstr. 64
6236 Eschborn

Comiczeichner/ innen gesucht

Mögen auch Sie etwas Auflockern-des und Heiteres in Computerzeit-schriften?

Wir möchten in Zukunft regelmä-ßig Cartoons und kleine Comicstrips veröffentlichen und suchen dazu noch begabte Zeichner. Wenn Sie eigene Ideen haben, um eine Serie zu gestalten, diese Ideen in Bilder umsetzen können und andere Leute damit auch zum Lachen bringen, dann setzen Sie sich bitte augenblicklich mit uns in Verbindung und schicken Sie einige Probezeichnungen!

Nach Möglichkeit sollten die Comics eine oder mehrere immer wiederkehrende Personen zum Inhalt haben und computerbezogen (nicht unbedingt AMIGA) sein.

Nicht nur Comics sind denkbar, auch sonstiges in Wort und Bild – seien es Karikaturen, Kuriositäten, Gedichte, lustige Erfahrungen aus dem AMIGA-Leben, humorvolle Berichte bis hin zu geballtem Unfug. Auch hier soll natürlich der Bezug zu Computern bestehen.

Bei Interesse wenden Sie sich bitte an die Kickstart-Redaktion, Telefonnummer nicht vergessen!

KICKSTART-Redaktion
'Comic'
Industriestr. 26
6236 Eschborn
Tel. 061 96/41245

ORD

Public Domain gesucht

Für viele ist Public-Domain bereits ein Begriff. Es sind Programme, die von den Autoren freigegeben wurden und somit frei kopiert werden dürfen. Damit stehen sie jedem Anwender zur Verfügung.

KICKSTART hat sich entschlossen, eine eigene Public-Domain-Sammlung zu schaffen. Darin werden neben einer Auswahl an ausgewählter, schon existierender Software auch eigene Programme enthalten sein. Wir möchten mit dieser Sammlung gerade für Neulinge der AMIGA-Szene die Möglichkeit schaffen, gute Software für ihren Rechner zu bekommen, ohne tief in den Geldbeutel greifen zu müssen.

Schon jetzt hat sich gezeigt, daß die Qualität der privaten Programmierkunst an manches professionelle Produkt heranreicht und sich davor nicht verstecken muß. Zwar ist meist nicht die Leistung einer Softwarefirma zu erreichen, doch bestätigen Ausnahmen bekanntlich die Regel.

Nicht nur große Programme sind dabei gefragt; auch Utilities, die im Einzelfall von unschätzbarem Wert sein können, sind wichtige Bestandteile der Public-Domain. Auch ausgefeilte Routinen und Funktionsbibliotheken können für andere Programmierer sehr wichtig sein. Insgesamt sind dem Einfallsreichtum keine Grenzen gesetzt.

Was bei anderen Rechnersystemen funktioniert, sollte beim AMIGA schon lange funktionieren – was sind schon 500 MS-DOS Disketten oder die ST-Sammlung! Deshalb der Aufruf an alle Programmierer, die sich des Rechners mächtig fühlen: Tun Sie Ihren Teil dazu, um die AMIGA-Public-Domain Szene anzukurbeln und sie in der Computerwelt zu einem Begriff zu machen.

Wenn Sie sich an der KICKSTART-Sammlung beteiligen und Ihre Programme gerne der Allgemeinheit zur Verfügung stellen möchten, so schicken Sie uns das/die Programm/e einfach auf Diskette zu. Bitte bestätigen Sie, daß das Programm von Ihnen geschrieben wurde und daran keinerlei kommerzielle Rechte bestehen.

Wir halten die Public-Domain Software für eine sehr nützliche Kreation und möchten unseren Teil zu ihrer Verbreitung tun. Die Sammlung wird ab sofort zum Selbstkostenpreis über die Redaktion zu erhalten sein.

KICKSTART Redaktion
'PD-Einsendung'
Schwalbacherstr. 64
6236 Eschborn
Tel. 0 61 96 / 4 124 5

AMIGA-Clubs bitte melden

Ein Aufruf an alle AMIGA Clubs und die USER-Vereinigungen. Wir möchten allen Amiga-Usern die Möglichkeit geben, sich an Clubs anzuschließen. Dort finden Sie aller Voraussicht nach auch andere Anhänger des AMIGA, die Sie verstehen und gleiche Interessen haben. Doch dazu benötigt man die Adressen, die wohl äußerst selten im Telefonbuch stehen. Deshalb rufen wir alle AMIGA-Clubs auf, uns ihre Kontaktadresse zu nennen. Wir werden sie und hoffen, damit den Clubs die Chance zur Repräsentation geben zu können. Bitte geben Sie eine kurze Information über die Tätigkeiten der Clubs und ihre Besonderheiten (z.B. Größe, Interessen, Aktivitäten). Bitte schicken Sie die Info an:

KICKSTART-Redaktion
'Clubinfo'
Industriestr. 26
6236 Eschborn

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 KICK-START 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 verspiegelten 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

verschiedene mit dem AEGIS-ANIMATOR erstellte Filme incl. PLAYER zum Abspielen der Filme (einige Filme benötigen auf einem AMIGA 1000 mehr als 512k Speicher!).

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

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 C-Forth

Diskette 23: Grafik-Show

Neue Bilder

Diskette 24: Grafik-Show

Neue Bilder

Diskette 25: UTILITIES

Versandbedingungen:

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

- Legen Sie pro bestellter Diskette DM 10,- als Scheck bei.
- Fügen Sie dem Betrag folgende Versandkosten (für Porto und Verpackung) bei: Inland DM 5,-, Ausland DM 10,-.
- Zu dem Kostenbeitrag müssen wir leider pro Scheck eine Scheckgebühr von DM 0,50 berechnen. Verwenden Sie deshalb nur einen Scheck, auf dem die Gesamtsumme steht (z. B. für zwei Disketten DM 25,50).
- Für ausländische Besteller sei erwähnt, daß Sie mit Euroschecks auch in DM bezahlen können.
- Legen Sie unbedingt einen Aufkleber mit Ihrer vollständigen Adresse bei.



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.

Der Nachbrenner: CLI-Shell

Wer sich schon einmal mit Amiga-Dos beschäftigt hat, wird den Ärger mit den mangelhaften Editiermöglichkeiten kennen. Eine Möglichkeit zur Abhilfe kommt jetzt von Metacomco: SHELL.

Jeder kennt das Problem: Man hat gerade ein Programm geschrieben und möchte es nun kompilieren bzw. assemblieren. Der Befehl wird, mit allen Zusätzen, eingetippt – und dann stellt man fest, daß der Tippfehlerteufel sich eingeschlichen hat (nach Murphy ist es natürlich das erste von 40 Zeichen). Im CLI bleibt dann nichts anderes übrig, als alles zu löschen und neu einzutippen – und wieder einen Fehler zu machen. Auf die Dauer ist das sehr nervenaufreibend und man wünscht sich eine Hilfe für diese Misere. Hier setzten die Entwickler von Metacomco-Shell an. Außerdem wollte man dem Anwender das wiederholte Eintippen von immer wiederkehrenden Kommandos ersparen oder erleichtern. Der erste Kontakt mit Shell findet noch in der alten CLI-Umgebung statt, wo das Programm installiert wird. Hierbei wurde großer Wert auf bequeme Bedienung gelegt. Auf der Diskette findet man Batchdateien, die selbständig eine Workbench-Diskette in eine Systemdiskette umwandeln, die das Shell in lauffähiger Ausführung enthält. Positiv zu bemerken ist, daß die Shell-Entwickler sich an die Workbench V1.1 erinnert haben und dafür sorgten, daß diese auch in Verbindung mit Shell benutzt werden kann. Dazu findet man auf der Diskette ein Programm namens „Shellpatch“, das die Workbench V1.1 patcht. Anschließend kann das Shell sorglos in die Version 1.1 integriert werden. Das ist ein nicht zu unterschätzender Pluspunkt, da es manche ältere Programmversionen gibt, die nur unter der Workbench V1.1 laufen. Auch bei solchen Programmen muß man also nicht auf die Annehmlichkeiten von Shell verzichten. Insgesamt erwies sich die Inbetriebnahme des Shell als sehr einfach und dürfte auch dem unerfahrenen Programmierer keine Schwierigkeiten

bereiten. Was ebenfalls positiv auffällt, ist der geringe Platzbedarf auf Diskette. Wer einen Sidecar oder ein Bridgeboard besitzt, kennt die Misere mit dem Speicherplatz. Wenn man einige Druckertreiber (man braucht ja ohnehin nur einen) auf der Arbeitsdiskette löscht, reicht der Platz auf der Workbench-Diskette für die Installation von Shell.

Der Aufruf von Shell erfolgt mit dem gleichnamigen Befehl oder von der Workbench aus durch Anklicken des kunstvoll gestalteten Icons. Das Shell zeigt sich daraufhin wie ein normales CLI-Fenster. Die Vorteile, die es bietet, zeigen sich erst, wenn man Befehle eingibt. So kann man nun die aktuelle Eingabezeile richtig editieren. Im CLI-Fenster erstreckt sich das Editieren auf das Löschen des letzten Zeichens und einfaches Anfügen. Shell ermöglicht es nun, frei mit den Pfeiltasten in der aktuellen Kommandozeile hin- und herzuwandern, man kann wortweise vor- und zurückgehen sowie an Zeilenanfang oder -ende. Außerdem kann man Zeichen löschen oder einfügen. Allein diese Möglichkeiten erleichtern das Arbeiten mit dem AmigaDOS ungemein. Das Shell bietet aber noch weitere Arbeitserleichterungen. Auffällig ist dabei besonders der „history“-Befehl und die mit ihm zusammenhängenden Kommandos. Im Shell wird eine Liste der Eingaben geführt. Jeder Befehl, der mit RETURN abgeschlossen wurde, wird in die Liste aufgenommen und erhält in aufsteigender Reihenfolge eine Nummer. Die Länge der „Ahnentafel“ kann mit – history <n> – auf einen Maximalwert gesetzt werden. Dabei hat sich ein Wert von n=10 als sinnvoller Kompromiss erwiesen. So hat man fast alle Kommandos, die man ständig benutzt, gespeichert und benötigt doch keinen übermäßigen Speicherplatz für die Liste. Mit – history – kann man sich die Liste betrachten. Ein Ausrufezeichen, gefolgt von einer Zahl, bewirkt das Abarbeiten des entsprechend nummerierten Befehls. Dabei wird das Kommando noch einmal angezeigt, es kann aber nicht

mehr editiert werden. Mit den Cursor-Up- und Down-Tasten kann man beliebig in der Liste auf- und abwandern und dann auch neu editieren. Um den Befehl abzuschicken, genügt es, einfach die Eingabetaste zu drücken, ohne daß es notwendig ist, an das Zeilenende zu gehen.

Für die Verkürzung der Eingaben stehen zwei weitere Befehle zur Verfügung. Mit dem ersten, „alias“ (frei mit Deckname übersetzt), kann man allen CLI-Befehlen zusätzliche Namen geben, die man geschickterweise recht kurz wählt. Denselben Zweck erfüllt bei einfachen Eingaben der „set“-Befehl. Hierzu ein kleines Beispiel auf Bild 1.

```
Aus folgender Eingabezeile :
assem dfl:arc/window.asm -o t/t/e -i dfl:hdrs
wird mit folgenden Zuweisungen von Namen und Variablen
alias a assem
set i -i dfl:hdrs
set t t/t/e
set a dfl:arc/window.asm
```

Eine Beschleunigung beim Abarbeiten von Befehlen wurde mit „resident“ implementiert. Im Normalfall, wenn kein Shell installiert ist, wird jeder Befehl von der Workbenchdiskette geladen. Dies ist auch der Grund dafür, daß der Amiga bei einer Kommandoingabe immer die Workbench verlangt, falls man sie aus dem Laufwerk entfernt hat. Mit obengenanntem Befehl ist es nun möglich, einzelne Befehle in den Arbeitsspeicher zu übertragen und den Zugriff darauf umzuleiten. Das hat gegenüber der Verwendung der RAM-Disk für diesen Zweck noch den Vorteil, daß ein residentes Kommando nur einmal im Speicher ist, auch wenn der Befehl in mehreren Tasks benutzt wird. Lädt man z.B. „ed“ auf die Ramdisk und startet ihn von einem oder zwei CLIs aus, so hat man „ed“ 2- (bzw.3-) mal im Hauptspeicher. Bei Benutzung von „resident“ wird der Speicherplatz nur einmal benötigt. Die Invertierung da-

von ist „nonres“ und expediert das Programm aus dem Hauptspeicher.

Beim Wechsel von Directories fehlt meist der Überblick, deshalb wankt man in den Verzeichnissen planlos herum. Mit dem Befehlen „push“ und „pop“ soll das ein Ende haben: Wer mit „push <pfad>“ in ein Verzeichnis wechselt, kann mit „pop“ in das zuvor aktuelle Directory zurückspringen.

Letztes Feature, das Shell zu bieten hat, ist die Verwendung der Funktionstasten, was im CLI ja gänzlich unmöglich ist. Die Belegung erfolgt sehr einfach mit `-key <Nummer> text-`. Man sollte dabei beachten, dass manche Funktionstasten von Shell bereits belegt sind.

Was man schwarz auf weiß besitzt...

...kann man bei Metacomco leider nur in Englisch nach Hause tragen. Mit geringen Kenntnissen dieser Sprache kommt man noch gut über die Runden. Inhaltlich macht das knapp 100 Seiten starke DIN A5-Büchlein, in drei Kapitel aufgeteilt und optisch

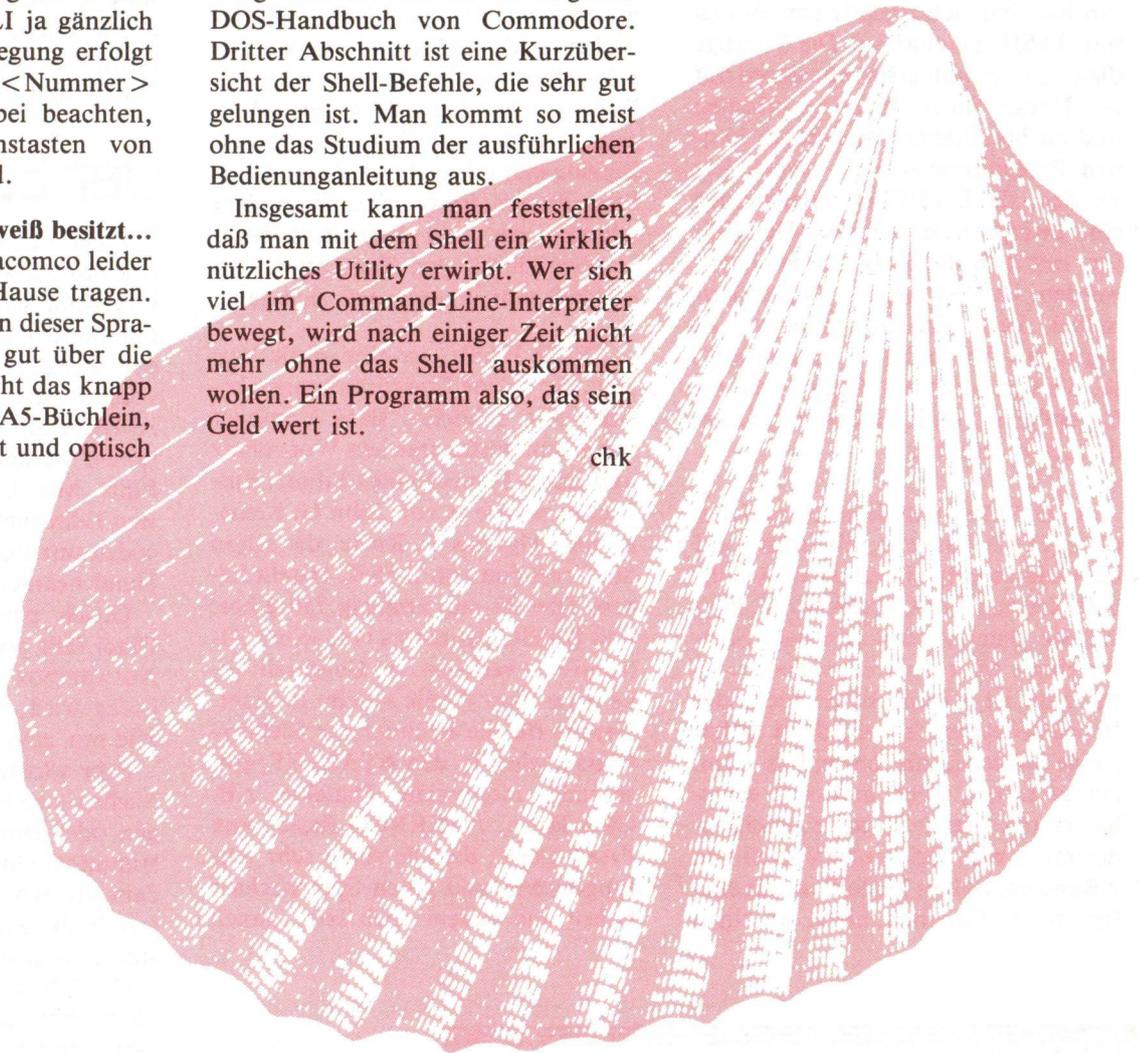
gut gegliedert, einen guten Eindruck. Es ist es eine Freude, darin nachzuschlagen. Man hat immer den Überblick und findet rasch und sicher die gesuchten Befehle. Der erste Teil des Handbuchs befasst sich eingehend mit dem Shell. Die Installation wird genau beschrieben; Sinn und Zweck der Shell-Kommandos werden erläutert. Im zweiten Kapitel werden die CLI-Kommandos der Workbench-Version 1.2 beschrieben. Die Beschreibung erinnert stark an das originale DOS-Handbuch von Commodore. Dritter Abschnitt ist eine Kurzübersicht der Shell-Befehle, die sehr gut gelungen ist. Man kommt so meist ohne das Studium der ausführlichen Bedienungsanleitung aus.

Insgesamt kann man feststellen, daß man mit dem Shell ein wirklich nützliches Utility erwirbt. Wer sich viel im Command-Line-Interpreter bewegt, wird nach einiger Zeit nicht mehr ohne das Shell auskommen wollen. Ein Programm also, das sein Geld wert ist.

chk

Befehlsübersicht des Shells:

alias	key
cd	nonres
edu	path
esc	pop
help	push
history	resident
	set



True BASIC™

Fast jeder, der mit Computern zu tun hat, hat sich irgendwann einmal mit BASIC beschäftigt. Die Vorzüge dieser Programmiersprache liegen auf der Hand: Sie ist leicht zu erlernen und zu handhaben. Außerdem können Programme schnell ausprobiert werden, da BASIC im allgemeinen eine Interpretersprache ist. Der Editor, mit dem die Programme erstellt werden, ist ebenfalls im System enthalten. Diese Zusammenfassung ist vor allem für Anfänger von Interesse, die sich nicht mit Tätigkeiten wie Compilieren und Linken auseinandersetzen müssen.

Jeder AMIGA-Besitzer kennt sicher die EXTRA-Diskette, die beim Kauf des Gerätes mitgeliefert wird. Auf ihr befindet sich das AMIGA-Basic, kurz ABasic genannt, das von dem Softwarehaus Microsoft entwickelt wurde. Nun ist ABasic sicher nicht der Weisheit letzter Schluß, denn es hat einige Mängel, die besonders dann zutage treten, wenn man sich ernsthafter mit dieser Sprache auseinandersetzen will. Der große Vorteil von ABasic ist, daß es kostenlos mitgeliefert wird. Daneben bietet es einen

großen Befehlsvorrat, der das Programmieren der spezifischen Besonderheiten des AMIGA unterstützen soll. Allerdings ist die Bedienung des Editors nicht besonders gut gelungen: Er bietet nur die notwendigsten Befehle und keinen besonderen Komfort. Zudem ist er nicht besonders schnell, was sich besonders beim Scrollen durch den Text zeigt, das wirklich sehr langsam vor sich geht.

TRUE BASIC ist eine Basic-Version, die solche Einschränkungen behebt: Ein schneller Basic-Compiler mit einem umfangreichen Wortschatz und einem komfortablen Editor. Entwickelt wurde es von John G. Kemeny und Thomas E. Kurtz, die schon die Urversion von BASIC erstellt haben. Dies ist nun zwanzig Jahre her, und BASIC hat nichts von seiner Attraktivität verloren. Im Gegenteil: Eine neue Basic-Welle scheint in Sicht. Gerade für die Rechner der 68000er-Klasse, wie z.B. den ATARI ST, gibt es einige interessante Dialekte, z.B. GFA-Basic, OMIKRON-Basic oder FAST-Basic, die alle einen sehr umfangreichen und guten Sprachschatz haben und zudem noch sehr schnell

sind. TRUE BASIC ist das erste professionelle 'Compiler'-Basic, das es für den AMIGA gibt. TRUE BASIC ist ein Basic-Dialekt, der auf verschiedenen Betriebssystemen und Computern lauffähig ist, z.B. auf IBM-Computern und Kompatiblen, Apple Macintosh und auch auf dem ATARI ST. Dadurch können Programme, die auf bestimmten Rechnern mit TRUE BASIC erstellt wurden, leicht auf andere übertragen werden. Die Anpassung bezieht sich dabei auch, sofern der jeweilige Computer grafikfähig ist, auf die Grafik und die Fenstertechnik.

Der Editor

Einer der wichtigsten Bestandteile des Basic-Systems ist sicher der Editor, denn schließlich verbringt man mit diesem Teil die meiste Zeit bei der Programmentwicklung. Er ist völlig in die Benutzeroberfläche des AMIGA eingebunden, fast alle Funktionen werden mit der Maus ausgeführt. Man kann jedoch die wichtigsten Editorbefehle auch über Tastencodes aufrufen, ohne die Maus in die Hand nehmen zu müssen.

Gleich beim ersten Laden einer Datei fällt positiv auf, daß hier eine Auswahlbox (Fileselectorbox) angeboten wird, bei der das gewünschte File nur angeklickt werden muß.

Sehr wichtig sind die Befehle zum Suchen und Ersetzen von Textpassagen oder zum Anspringen eines bestimmten Unterprogramms. Außerdem gibt es Möglichkeiten, um schnell ans Ende oder den Anfang des Listings zu gelangen.

Da TRUE BASIC sowohl mit als auch ohne Zeilennummern arbeitet, gibt es eine Funktion, die Zeilennummern löscht bzw. einfügt. Nun soll es ja Programmierer geben, die sich nur ungern die Mühe machen, ihre Programme so zu strukturieren, daß Schleifen und Abfragen eingerückt sind. Diesen Menschen wird jetzt mit dem Befehl DO FORMAT geholfen, denn dieser nimmt das Einrücken der betreffenden Zeilen nachträglich vor, damit das Listing besser zu überblicken ist.

Der Editor bietet noch eine Vielzahl anderer Befehle, die hier nicht alle erwähnt werden können. Auf je-

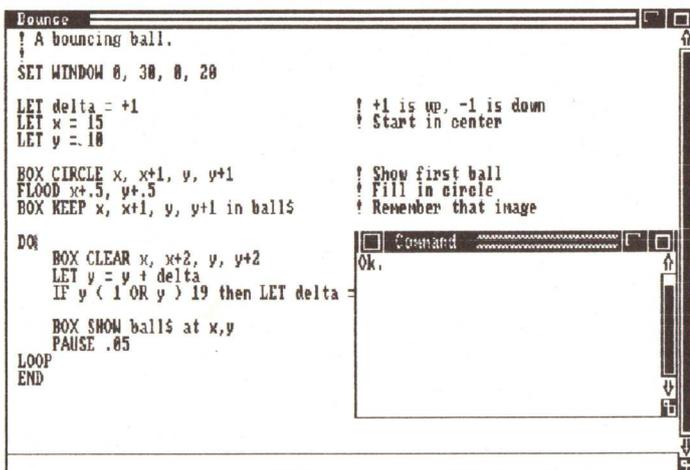


Bild 1:
Bildschirm von
TRUE BASIC

den Fall ist der Bedienungskomfort des Editors erheblich größer als bei ABasic. Dies sollte man beachten, wenn man ein BASIC zum Lernen und Arbeiten wählt.

Strukturbefehle

TRUE BASIC bietet viele Befehle an, die es ermöglichen, Programme klar gegliedert und übersichtlich zu schreiben, wie dies auch in Sprachen wie PASCAL, C und MODULA vorgesehen ist. Deshalb stehen neben der gewöhnlichen FOR..NEXT-Schleife noch Strukturen wie z.B.:

```
DO WHILE Bedingung
...
LOOP
oder
DO UNTIL Bedingung
...
LOOP
```

und verschiedene Variationen zur Auswahl. Alle Schleifen können mit einem EXIT-Befehl verlassen werden, was jedoch nicht immer der beste Weg ist. Die IF/THEN-Struktur wurde um ELSE und weitere ELSE-IF-Zweige erweitert:

```
IF Bedingung THEN
...
ELSEIF Bedingung
...
ELSE
...
END IF
```

In Anlehnung an Sprachen wie PASCAL u.a. wurde auch der Entscheidungsbefehl CASE aufgenommen:

```
SELECT CASE Ausdruck
CASE Konstante1 ...
CASE Konstante2 ...
CASE ELSE ...
END SELECT
```

Ausgerüstet mit diesen komfortablen Befehlen, sollte es jedem Programmierer möglich sein, Listings zu schreiben, die eine klare, leicht zu überblickende Struktur aufweisen. Wenn dies gewährleistet ist, ist ein Programm auch zu einem späteren Zeitpunkt noch lesbar, ohne daß man lange darin herumwühlen muß.

Befehle wie GOTO und GOSUB, die meistens zu dieser schlechten Struktur führen, können nun völlig wegfallen. Auch Zeilennummern sind nicht mehr notwendig, weil Sprünge nur noch zu Unterprogrammen oder zu Labels erfolgen.

Unterprogramme Funktionen und Libraries

Eine weitere, sehr leistungsfähige Möglichkeit, Programme zu strukturieren, ist die Verwendung von Unterprogrammen oder Modulen. Dabei wird das Problem in Teilbereiche (Module) zerlegt, die in Form eines separaten Unterprogramms programmiert werden. Im Extremfall der Modularisierung enthält das Hauptprogramm dann nur noch die Aufrufe der Unterprogramme:

```
DEF FUNC(x,y,...) = Ausdruck
...
END DEF
SUB Sub(x,y,...)
...
END SUB
Hauptprogramm:
  FUNC(x,y,...)
  CALL Sub(x,y,...)
END
```

Programmiersprachen übernommen wurden. DEF FUNC und SUB sind Funktions- bzw. Prozeduraufrufe mit Parameterübergabe. Das heißt, daß der aufrufende Befehl (z.B. CALL Sub(5,3)) Werte (hier: 5,3) an das Unterprogramm übergibt, die dann von diesem weiterverarbeitet werden, ohne daß die Variablen des aufrufenden Programmes verändert werden. Es ist allerdings auch möglich, Funktionsaufrufe mit Rückgabeparametern zu versehen.

Bekanntlich sind gerade lange Programme sehr unübersichtlich und fehleranfällig. Deshalb können sie in Teilprogrammen abgespeichert wer-

```
! REKURSION mit TRUE BASIC
!
def fac(n)
  if n<=1 then let fac=1 else let fac=n*fac(n-1)
end def
for i=1 to 10
  print i,fac(i)
next i
end
```

Listing 1: Rekursion in Basic
Die Funktion ruft sich selbst auf

den, die dann entweder nacheinander oder von einem Hauptprogramm aus aufgerufen werden. Somit ist wieder eine Art der Modularisierung geschaffen worden, bei der jedes Modul ein eigenes Programm ist, das separat entwickelt und ausgetestet

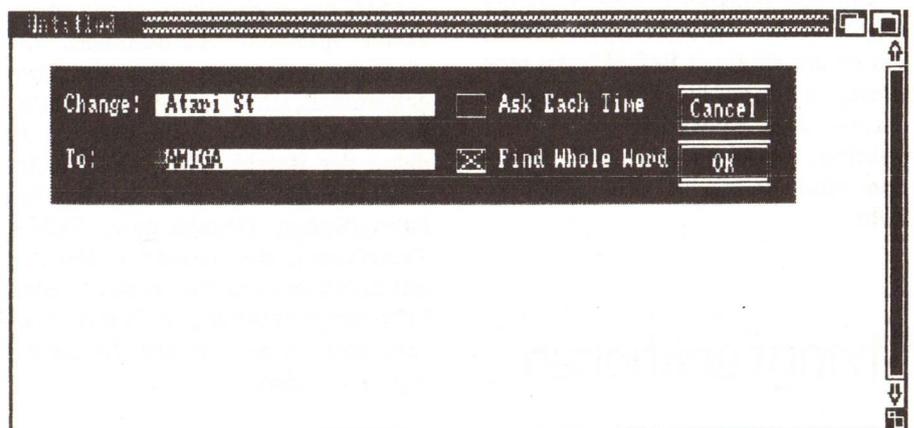


Bild 2: Bildschirm von TRUE BASIC

Wie diesem Beispiel zu entnehmen ist, gibt es mehrere Unterprogrammstrukturen, die ebenfalls von anderen

wird. Erst wenn die einzelnen Module einwandfrei funktionieren, werden sie mit dem Hauptprogramm verbun-

den. Dies ist eine sehr elegante Methode, die sich natürlich besonders bei langen Programmen anbietet; ihr Vorteil ist auf jeden Fall eine einfache Wartung des Programms und der Module sowie eine einfachere Fehlerlokalisierung.

Mathematische und trigonometrische Funktionen

Die mathematischen Funktionen enthalten neben ABS, EXP, INT, LOG, LOG2, LOG10, MAX, MIN, RND, SGN, SQR noch Befehle zum Runden (ROUND) und Abschneiden (TRUNCATE) von Nachkommastellen. Außerdem noch MODULO, DIVIDE und REMAINDER, die beim Teilen von Zahlen den ganzzahligen Anteil und/oder den Rest angeben.

Die trigonometrischen Funktionen bestehen neben SIN, COS, TAN noch aus ATN, PI, DEG, RAD und ANGLE. Weitere Funktionen (z.B. die Areafunktionen usw.) sind in den Libraries definiert, die bei Bedarf eingebunden werden können.

Sehr praktisch ist es, daß Winkelangaben im gebräuchlichen Bogenmaß und auch im Grad eingegeben werden können. Dies erspart bei manchen Anwendungen die recht lästige Umrechnung.

Sehr leistungsfähig sind auch die eingebauten Matrix-Operationen. Mit einem einzigen Befehl kann eine ganze, mehrspaltige Matrix eingelesen werden. Matrixaddition, -Subtraktion, -Multiplikation und -Inversion sind ebenfalls kein Problem mehr.

Stringfunktionen

Einige spezielle Stringoperationen erweitern das 'Standard-BASIC': LCASE und UCASE wandeln Buchstaben in Klein- bzw. Großbuchstaben; TRIM, RTRIM und LTRIM entfernen Leerzeichen, die an verschiedenen Stellen des Strings stehen; REPEAT kopiert einen String belie-

big oft in einen anderen; USING erlaubt die formatierte Ausgabe von Zeichen, Strings und Zahlen; POS gibt den Ort an, an dem ein Teilstring in einem anderen String steht.

Dateien

Drei Typen von Dateien werden unterstützt: TEXT, RECORD und BYTE. TEXT-Dateien sind dabei die 'normalen' Dateien, die mit INPUT und PRINT beschrieben und gelesen werden. RECORD und BYTE sind platzsparend gepackte Dateien, die mit READ und WRITE behandelt werden. Dateien können mit einer Schleife eingelesen werden, die gleich die Endbedingung enthält:

```
DO WHILE MORE #1
...
LOOP
bzw.
DO WHILE MORE DATA
...
LOOP
```

Einige weitere Befehle erleichtern die Behandlung von Dateien, die mit Basic bisher nicht gerade einfach war. TRUE BASIC kommt somit den Programmierern entgegen, die bisher immer auf die schlechten Möglichkeiten und die dadurch notwendigen Mühen oder Einschränkungen gewettet haben. TRUE BASIC hat auch hier einiges von anderen Sprachen übernommen. Zusätzlich sind weitere Befehle dazugekommen, damit das Arbeiten möglichst einfach ist. Sehr hilfreich ist dabei der Befehl ASK, der viele Informationen über eine Datei geben kann (Status, Organisation, Größe, Zeigerstand, Recordgröße). Mit diesen Befehlen kann das immer wiederkehrende Problem der Dateiverwaltung endlich einfach und elegant bewältigt werden.

Grafik & Sound

Die Standard-Grafikbefehle wie POINT, LINE, ELLIPSE, AREA und TEXT sind erwartungsgemäß vorhanden. Dazu kommen jedoch

noch sehr leistungsfähige Befehle, die helfen, Problemstellungen eleganter zu lösen. Auch die Verwaltung mehrerer Bildschirmfenster (WINDOWS) ist einfach gelöst. Der wichtigste Befehl dabei ist SET WINDOW, denn damit kann der Wertebereich für die Grafikausgabe bestimmt werden. Dann ist bei Grafikausgaben kein Umrechnen der Werte auf die Grafikpunkte des Bildschirms mehr notwendig, denn diese Arbeit übernimmt das Basic.

Wenn ein grafisches Element (z.B. ein Dreieck) mehrmals auf dem Bildschirm erscheinen soll, wobei Drehungen, Größenänderungen und ähnliches möglich ist, dann ist der Befehl PICTURE der richtige. Die Figur wird dazu in einer Procedure PICTURE definiert, in der die verschiedenen Anweisungen stehen, die das Objekt beschreiben. Der Übergabeparameter und ein Attribut bestimmen die Form.

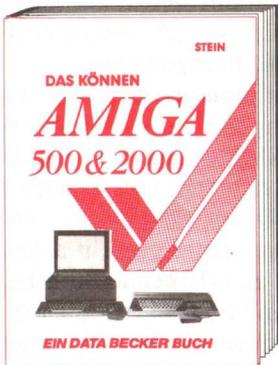
Recht interessant ist der BOX-Befehl, der einen bestimmten Bildschirmabschnitt als BOX definiert und in einem String abspeichert. Diese BOX kann beliebig oft an jede Stelle des Bildschirms kopiert werden. Benutzt man dazu noch einen speziellen Befehl zum Löschen der Box, kann hiermit ein Objekt bewegt werden. Ein Nachteil dieser Animationsart ist allerdings, daß sie auf dem normalen Monitor sehr stark flimmert.

Wer sich gerne mit Musik beschäftigt, für den sind die Befehle PLAY und SOUND von besonderem Interesse. Mit PLAY werden die Melodien in der Standard-Western-Notation eingegeben. SOUND bietet dagegen eine bessere Kontrolle über die Töne und dient zur Erzeugung spezieller Geräusche. Eine Besonderheit des SOUND-Befehls ist, daß eine spezielle Option erlaubt, die Melodie im Hintergrund abzuspielen, während das Programm weiterläuft. Damit ist es zum Beispiel leicht möglich, Spiele mit Musik zu unterlegen.

Die Struktur

Neben den Vorzügen von TRUE BASIC gibt es jedoch auch einige

NEUE AMIGA- BUCHHITS



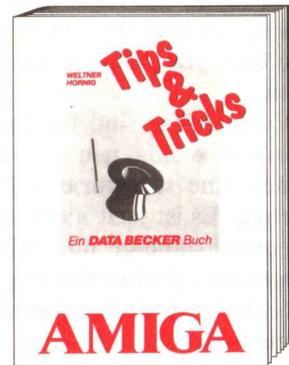
Was leisten die neuen Amigas? Hier finden Sie die Antwort. Unabhängig davon, ob Sie den Amiga schon haben oder den Kauf planen: Dieses Buch bietet Ihnen Entscheidungshilfen, technische Details und jede Menge von dem, was man mit Amiga 500 & 2000 so alles anstellen kann. Eben Informationen, die man braucht, 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,-



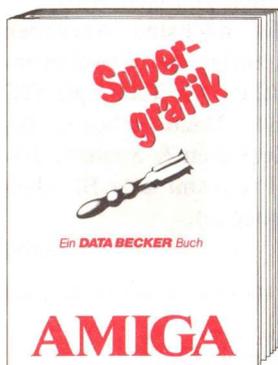
Wählen Sie gleich den richtigen Einstieg zu Ihrem Amiga 500. Denn das Handbuch läßt Sie dabei völlig allein. Versuchen Sie es lieber gleich mit Amiga 500 für Einsteiger. Hier heißt es: Anschließen und loslegen. Verständlich für jedermann zeigt Ihnen dieses Buch: Workbench, Amiga Basic, CLI und AmigaDOS.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



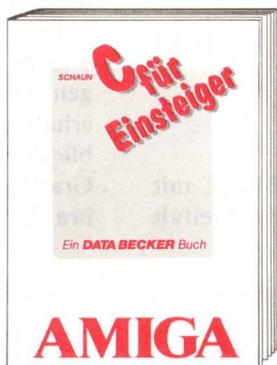
Das erfolgreiche Buch zu Amiga-BASIC – jetzt in der Neuauflage! Erweitert um Kickstart 1.2, neuer Workbench und Amiga 500 & 2000. Mit allem, was BASIC-Programmiern Spaß macht: Grafik und Sound, Laden und Speichern von Graficraft-Bildern in BASIC-Programme, sequentielle und relative Dateien, Business-Grafik,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



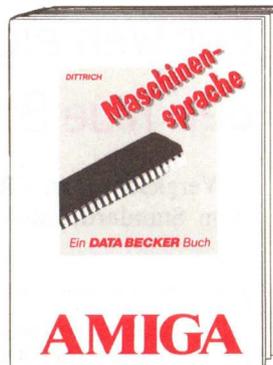
Amiga Tips & Tricks. Ein Buch, das voller Überraschungen steckt: 64 Farben gleichzeitig auf dem Amiga. Von BASIC aus Zugriff auf die Libraries. Benutzung verschiedener Zeichensätze in BASIC. Sinnvoller Einsatz von Windows, Screens und Menüs. Tips zu einzelnen Grafikbefehlen, Programm- und Amiga-DOS-Routinen!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



Wer die enorme Grafikfähigkeit seines Amiga ausschöpfen will, braucht entsprechendes Know-how: Grafikprogrammierung mit den vorhandenen BASIC-Befehlen, Nutzung der Libraries, die Register der Grafik-Chips, CAD, Aufbau und Programmierung von Screens, Windows, HAM, Halfbrite und Interlace aus BASIC und C. Informationen, die im Supergrafikbuch zum Amiga mit vielen Programmbeispielen anschaulich vermittelt werden.
Amiga Supergrafik Hardcover
ca. 700 Seiten, DM 59,-
erscheint ca. 6/87



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 Hardcover
ca. 250 Seiten
DM 39,-
erscheint ca. 6/87



Schreiben Sie Ihre Programme in Maschinsprache – 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 Maschinsprache Hardcover
ca. 300 Seiten
DM 49,-
erscheint ca. 6/87

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 Versandkosten zzgl. DM 5,- Verrechnungsscheck liegt bei
Name _____
Straße _____
Ort _____

Nachteile, die in Kauf genommen werden müssen. So wird vom Compiler eine bestimmte Struktur des Programms verlangt, die unbedingt eingehalten werden muß. Dies bedeutet, daß ein Programm keine Syntax-Fehler enthalten darf, denn diese werden noch vor dem Starten bemängelt: Alle Zuweisungen müssen mit dem Befehl LET eingeleitet werden. Dies war bisher zwar auch möglich, aber nicht Pflicht. Am Ende des Programms muß der Befehl END stehen, außerdem darf pro Zeile nur ein Befehl stehen.

Diese Vorschriften sind für Basic-Programmierer völlig neu, sie fördern jedoch eine strukturierte Programmierung. Es ist jetzt nicht mehr möglich, unvollständige und fehlerhafte Programme probeweise zu starten, wodurch der Basic-Programmierer zu ungewohnter Disziplin gebracht wird.

Compiler und Benchmarks

Bei jedem Starten des Programms mit RUN wird der Source-Code auf Syntax-Fehler untersucht. Außerdem wird ein sogenannter Zwischencode erzeugt, der kompakt und schnell ist. Zur Demonstration der Geschwindigkeit wurden die bekannten Benchmark-Tests durchgeführt (siehe Listing 2). Die Werte wurden mit denen von ABasic verglichen.

Benchmark Nr.:	AmigaBasic	TRUE BASIC
1	0.42	0.06
2	1.64	0.24
3	3.66	0.66
4	4.32	1.01
5	4.87	1.3
6	8.79	1.86
7	12.66	2.74
8	17.38	11.5

Was AMIGA-Anwender vermissen werden

Wer sich schon einmal näher mit dem ABASIC beschäftigt hat und

diesen Bericht bis hierher sorgfältig gelesen hat, der wird trotz der vielen Befehle, die TRUE BASIC besitzt, feststellen, daß einige Bereiche fehlen. Dies sind vor allem die AMIGA-spezifischen SPRITE-Befehle, aber auch PULL-DOWN-Menüs, die Fenstertechnik und die komfortablen Interruptbefehle (ON TIMER/MOUSE /MENU GOSUB), die bei dem entsprechenden Ereignis in ein Unterprogramm verzweigen.

Allerdings werden die meisten Funktionen nachgebildet. Es gibt die Möglichkeit, Fenster zu öffnen; allerdings sind diese Fenster lange nicht so komfortabel, wie der AMIGA-Anwender das gewohnt ist. Sie können nicht überlagert werden, außerdem haben sie kein Gadget zum Verschieben, Verkleinern usw.

Für Menüs gibt es eine spezielle Bibliothek, die diese Technik nachahmt; die Sprites werden durch Befehle wie BOX und PICTURE ersetzt.

Diese Befehle haben zwar nicht die gewohnte Form, aber sie bieten zum Teil sehr umfangreiche Möglichkeiten. Deshalb lassen sich die damit verbundenen Einschränkungen auch in Kauf nehmen, denn dadurch ist das Programm kompatibel auf den verschiedensten Rechnern.

Für wen eignet sich der True Basic?

Vergleicht man TRUE BASIC mit dem Standard, dann ist die Vielfalt

der Befehle überwältigend. Viele Bereiche, vor allem jedoch die Grafik, wurden gründlich erweitert. Besonders interessant sind jedoch die Strukturbefehle (WHILE, UNTIL, CASE) und die Unterprogrammtechnik (DEF FUNC, SUB), die eine gute Strukturierung der Programme erlauben.

Um einen neuen portablen 'Basic-Standard' verwirklichen zu können, mußten notgedrungen einige rechner-spezifische Besonderheiten entfallen, bzw. sie wurden so weit wie möglich durch ähnliche nachgebildet.

Fazit: Sieht man von den wenigen Einschränkungen ab, die man als Besitzer eines AMIGA mit TRUE BASIC in Kauf nehmen muß, dann ist dieser Basic-Dialekt genau das, wonach viele schon lange gesucht haben – eine leicht zu erlernende Sprache, mit vielfältigen Möglichkeiten zur strukturierten Programmierung und einer großen Anzahl von Befehlen, die das Programmieren erleichtern. Wenn man dazu bedenkt, daß die Programme auf so verschiedenen Rechnern, wie IBM, APPLE, ATARI ST und AMIGA laufen, dann ist auch klar, warum manche Einschränkungen notwendig waren. Trotzdem ist mit TRUE BASIC eine wirkliche Verbesserung des Basic-Standards erreicht worden, die für die meisten Anwendungen sicherlich genügen wird.

In einer der nächsten Ausgaben werden noch einige der Erweiterungen vorgestellt, die für TRUE BASIC erhältlich sind. Dazu gehören Bibliotheken (Sortieren & Suchen, 3D-Grafik) und Programme (z.B. Algebra, Trigonometrie).

(MN)

- + komfortabler Editor
- + Strukturbefehle (DO WHILE, DO UNTIL, IF THEN ELSE)
- + Modultechnik (Libraries)
- + gute Grafikbefehle und Animation
- + erzeugt Zwischencode (schnelle Programme)
- + portabler Code (PC u. a.)
- + viele Erweiterungen erhältlich (Libraries)

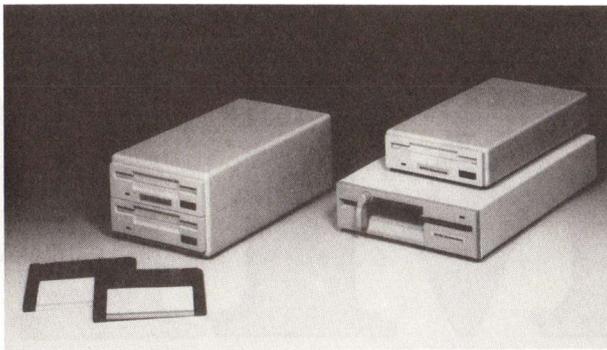
- relativ hoher Preis
- kein Run-Only-Compiler
- Einschränkungen bei WINDOWS, MENUS, SPRITES

Hardwarevoraussetzung: min. 512K RAM

Bezugsquelle: Softwareland
Franklinstrasse 27
CH-8050 Zürich
Tel. 00 41-1-311 59 59
Preis: DM 349.-

Original

Made in Germany



- F₁ 3,5"-Einzelfloppy, anschl. 389,- DM
- F₂ 3,5"-Doppelfloppy, anschl. 689,- DM
- F₅ 5,25"-Einzelfloppy, anschl. 559,- DM

NEC 1036 A + Interface + Kabel + Stecker
+ Anleitung zum Selbstbau einer 3,5"

Amiga-Floppy 299,- DM

- 3,5"-Gehäuse, Kunststoff, beige mit Befestigungsmaterial 34,90 DM
- NEC 1036A/NEC 1035LP, 1 MB, 3,5" 259,- DM

Amiga-Bücher

SOFTWARE	
Spielesoftware	Anwendersoftware
Alien Fires	99.-
Arena	79.-
Bard's Tale	99.-
Chessmaster 2000	89.-
Cruncher Factory*	a.A.
Defender of the Crown	89.-
Demolition*	a.A.
Faery Tale	99.-
Flightsimulator II	109.-
Flip Flop	39.-
Grand Prix*	89.-
Gunship*	89.-
Karate King	39.-
King of Chicago*	109.-
Neutralizer*	a.A.
Phalanx*	59.-
Return to Atlantis	99.-
S.D.I	109.-
Shanghai	79.-
Silent Service	89.-
Sinbad	99.-
Starglider	79.-
Strip Poker	79.-
Uninvited	99.-
Quiwi	59.-
Wintergames	59.-
Worldgames	59.-
Acquisition	599.-
Aegis Sonix V2.0	169.-
Aztec C Dev. V3.4a	599.-
CLI-Mate V1.2	69.-
Deluxe Music Con. Set	199.-
Deluxe Paint IIB	249.-
Dynamic-CAD	1149.-
Grabbit	69.-
Instant Music	99.-
Lattice C V3.1	299.-
Lisp	419.-
Macro Assembler	199.-
Marauder II	109.-
Meta Pascal	199.-
Modula-2 Standart	189.-
Modula-2 Developers	299.-
Pagesetter europ.Ver.	298.-
Printmaster	109.-
Scribble	199.-
K-Seka Assembler	129.-
Shell	139.-
Superbase (deutsch)	229.-
Toolkit	99.-
UBM Text V2.2	249.-
UCSD Pascal	169.-
Vip Professional	449.-

* = in Kürze lieferbar

Amiga Reference Manuals :	Public Domain Software
Hardware	62.50
Intuition	62.50
Exec	62.50
Libraries and Devices	88.-
	10 Disks 89.-
	30 Disks 249.-
	incl. Disketten

Komplette Softwareliste mit ca. 300 Prg. anfordern !

Harald Soyka * Hattinger Straße 685 * 4630 Bochum 5

PADERCOMP – Walter Ladz

Erzbergerstr. 27 · 4790 Paderborn · Tel. 0 52 51 - 3 63 96

FLOPPYSTATIONEN

- PADERCOMP FL 1** 448,-
3,5", 1 Mb, eingeb. Netzteil, NEC-Laufwerk, Abm. 240x105x40 mm, anschlussfertig, graues Metallgeh. Testbericht ATARI-Magazin 2/87, Seite 70
- PADERCOMP FL 2** 748,-
Doppellaufwerk übereinander, sonst wie FL 1
- PADERCOMP FL 3** 398,-
Zweitlaufwerk für Amiga
- NEC FD 1036 A 3.5"**, 1 MB, 32 mm Bauhöhe ... 245,-
neuestes Modell, keine Modifizierung für ATARI ST erforderlich.
- Industrie Floppystecker 9,90
- ST Kabel an Shugart-Bus 3.5" 29,90
- Monitorstecker 7,90

ZUBEHÖR

- 3.5" Disketten, ab 50 Stück** **Superpreise!**
- Disk Box SS-50, f. 50 3.5" Disketten** 19,90
- Media Box 1, f. 150 St. 3.5" Disketten** 39,90
- Druckerkabel ST** 34,90
- Dataphon S21/23, 300 bzw. 1200/75 Baud, BTX** 329,-
- Orion Farbmonitor CCM 1280 m. Kabel an Atari 260/520** 848,-
- NEC Multisync, alle drei Auflösungen** 1798,-
- Monitor-Ständer dreh-, schwenk- und kippbar** 29,-
- Joy-Star, Super Joystick mit 6 Microschaltern** 22,90
- Preisliste** **kostenlos**

Die aktuellen Preise erfragen Sie unter der PCB-Mailbox Tel. 052 51-3 17 80 (300 Bd., 8N1)

DRUCKER

- STAR NL 10 incl. Interface, Dt. Handbuch** 748,-
- Citizen 120D 120 Z/s, NLQ** 528,-
- OKIDATA ML 192 incl. vollaut. Einzelblatteinzug** ... 1198,-
- OKI-Laserline 6+, Centronics** 4995,-
- Mitsubishi DX-180W, 180 Z/s, NLQ, DIN A3** 998,-

Ein Schriftbild, fast wie gesetzt!
24-Nadeldrucker

- NEC P6, 24 Nadeln, 216 Z/s, DIN A 4** 1198,-
- NEC P7, 24 Nadeln, 216 Z/s, DIN A 3** 1598,-
- STAR NB 24/15, 24 Nadeln, 216 Z/s, DIN A 3** 1998,-

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.

DATA BECKER

AMIGO

TEXTOMAT Amiga

Die ideale Textverarbeitung

TEXTOMAT Amiga zeigt, wie einfach Textverarbeitung sein kann. Ohne langes Anlernen oder Lesen im Handbuch können Sie sofort Ihre ersten Texte erstellen. Ein typisches Programm für jeden Einsteiger also? Nicht nur. Schnelle Direktformatierung am Bildschirm, Funktionstastenbelegung, automatische Silbentrennung – das bietet TEXTOMAT Amiga dem professionellen Anwender. Daß diese Textverarbeitung zudem so komfortabel und schnell erlernbar ist, dürfte ihn wohl kaum stören.

TEXTOMAT Amiga in Stichworten:

Hohe Geschwindigkeit bei der Eingabe und Bearbeitung von Texten – sämtliche Funktionen über Menüleisten oder Kurzbefehle anwählbar – Funktionstastenbelegung – schnelle Direktformatierung am Bildschirm – automatische Silbentrennung – beliebig viele Tabulatoren – umfangreiche, sehr komfortable Druckeranpassungen – mit ausführlichem, deutschem Handbuch.

TEXTOMAT Amiga DM 99,-
erscheint ca. 6/87
geringfügige Änderungen vorbehalten

BECKERtext Amiga

Mehr als nur eine Textverarbeitung

Wer eine Textverarbeitung sucht, hat die Qual der Wahl. Soll sie schnell sein oder Direktformatierung am Bildschirm bieten? Soll sie souverän mit Worten, Sätzen und Textblöcken jonglieren können oder auch Zahlen verarbeiten? Verlangen Sie ein Höchstmaß an Leistung oder vor allem Bedienerfreundlichkeit? Bei BECKERtext Amiga stellen sich diese Fragen erst gar nicht. Hier müssen Sie weder auf Komfort noch auf Leistung verzichten.

BECKERtext Amiga in Stichworten:

Komfortables Rechnen im Text nicht nur spalten- sondern auch zeilenweise – Dezimaltabulatoren – ONLINE-Lexikon, das wahlweise während des Schreibens oder nachträglich den Text überprüft – es kann mit beliebigen, individuellen Lexika (auch Fremdsprachen) gearbeitet werden – schnelle Direktformatierung am Bildschirm – mehrspaltige Druckausgabe bis zu 5 Spalten – bis zu 999 Zeichen pro Zeile mit horizontalem Scrolling darstellbar, mit allen EPSON FX und kompatiblen Druckern lassen sich auch bis zu 999 Zeichen pro Zeile im Querdruckmodus ausdrucken – automatische Silbentrennung – vielseitige Funktionstastenbelegung – automatisches Stichwort- und Inhaltsverzeichnis – Multitasking fähig – Grafiken, die im IFF-Format vorliegen, können im Text ausgedruckt werden – nicht kopiergeschützt – mit ausführlichem Handbuch.

BECKERtext Amiga DM 199,-
erscheint ca. 7/87
geringfügige Änderungen vorbehalten

DATA BECKER

Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 31 0010

PRÄSENTIERT:

GA SOFTWARE

DATAMAT Amiga

Der Datenspezialist

DATAMAT Amiga ist die konsequente Umsetzung des erfolgreichen DATAMAT ST. Dabei werden natürlich alle speziellen Amiga-Eigenschaften, wie z. B. Multitasking, unterstützt. So lassen sich jetzt alle Daten auch mit dem Amiga professionell pflegen und verwalten – in der Geschwindigkeit, die man von diesem Rechner gewohnt ist, und mit einem Komfort, den man bei einer Programmsteuerung über Maus und Tastatur erwarten darf.

DATAMAT Amiga in Stichworten:

Maximal 8 offene Dateien gleichzeitig – Dateigröße, max. 2 Milliarden Zeichen – Datensatzgröße, max. 64.000 Zeichen – maximal 2 Milliarden Datensätze – Anzahl der Datenfelder, unbegrenzt – maximale Feldgröße, 32.000 Zeichen – Passwort-Schutz – Dateien werden auf Massenspeicher (hohe Datensicherheit) bearbeitet – maximal 20 Indexfelder mit wählbarer Genauigkeit (1-99 Zeichen) – komfortable Such- und Selektierkriterien (Bereiche, Und-, Oderverknüpfung ...) – Test-, Datums-, Zeit-, Zahlen- und Auswahl-Felder, Einlesen von IFF-Dateien – Datenaustausch mit anderen Programmen möglich, wichtig für Serienbrieferstellung – Programmsteuerung über Maus und Tastatur – frei gestaltbare Bildschirmmaske, bis zu maximal 5000 x 5000 Punkte groß – Bildschirmmaske unterstützt Grafikelemente wie Rechteck, Kreis, Linie, Muster usw. – Bildschirmmaske unterstützt verschiedene Textarten und -größen – mehrzeilige Textfelder mit Wortumbruch und Formatierungsmöglichkeiten – integrierter Druckermasken- und Listeneditor.

DATAMAT Amiga DM 99,-
erscheint ca. 7/87
geringfügige Änderungen vorbehalten

PROFIMAT Amiga

Das komplette Entwicklungspaket

Viele wissen um die fantastischen Möglichkeiten eines echten Assembler-Programms, wagen sich aber an eigene Programme selbst nicht heran. Dabei ist die Maschinensprache-Programmierung durchaus nicht nur ambitionierten 68000-Experten vorbehalten. Schnell erlernbar und überaus bedienerfreundlich steht jetzt jedem Anwender ein komplettes Entwicklungspaket zur Verfügung, mit dem er seinen Amiga voll ausreizen kann: PROFIMAT Amiga.

PROFIMAT Amiga in Stichworten:

Komplett in Assembler geschrieben, daher extrem schnell – integrierter Editor, Debugger, Disassembler und Reassembler – umfangreiche Betriebssystem-Bibliothek – läuft unter CLI und Workbench – erzeugt optional PC-relativen oder absoluten Code – Makros mit nahezu beliebig vielen Parametern unterschiedlicher Typen möglich – Fehlersuchfunktion – Cross-Referenz-Liste – bedingte und wiederholte Assemblierung menügesteuert – volle 32-Bit-Arithmetik – Debugger mit 68020-Single-Step-Emulation – lauffähig auf jedem Amiga mit 512 KByte und Kickstart 1.2.

PROFIMAT Amiga DM 99,-
erscheint ca. 7/87
geringfügige Änderungen
vorbehalten

BESTELL-COUPON

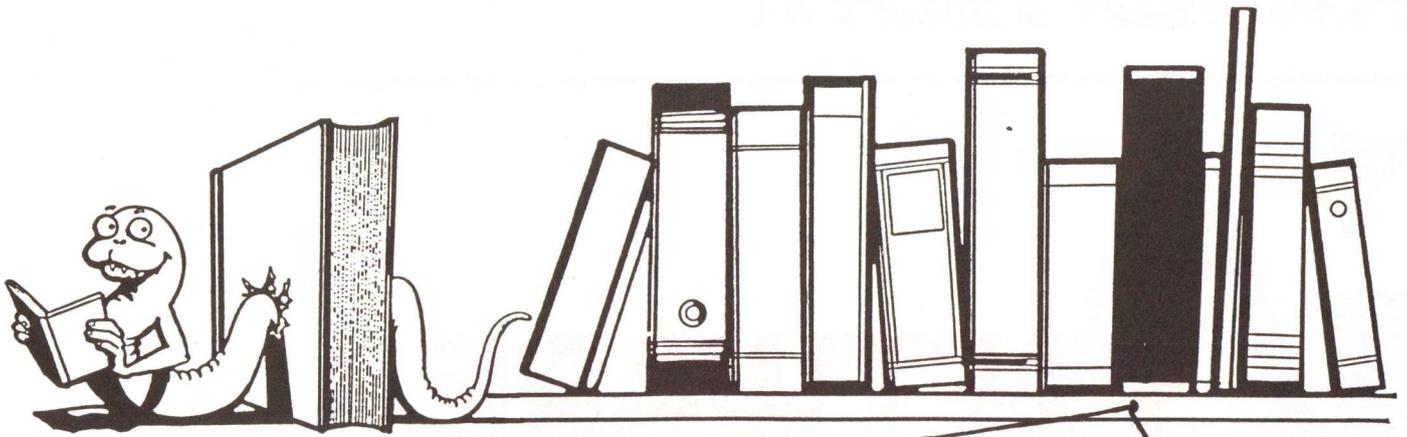
Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:

per Nachnahme zzgl. DM 5,- Versandkosten Verrechnungsscheck liegt bei

Name

Strasse

Ort



Das ‚A‘ und ‚O‘ des AMIGA-Betriebssystems



Das Amiga-Betriebssystem hat einen eigenwilligen Aufbau: Es ist in Libraries (Bibliotheken) unterteilt. Welchen Vorteil und Nutzen dieser bisher unbekannte Weg der Betriebssystem-Organisation bringt, möchte ich Ihnen hier verdeutlichen.

Alle Libraries werden mit dem Booten der Kickstart-Diskette in den Rechner geladen. Selbstverständlich ist dies nur beim Amiga 1000 der Fall, weil bei den Modellen 500 und 2000 die Software der Kickstart-Diskette sich bereits in fest eingebauten ROMs des Rechners befindet.

Doch was sind eigentlich Libraries? Die Beantwortung dieser Frage ist im Grunde recht einfach: Libraries sind Sammlungen von kleinen und effizienten Programmen, von denen vielfältige Aufgaben übernommen werden. Jede Library beinhaltet dabei die Routinen eines bestimmten Gebietes; die Library 'Intuition' z.B. besitzt Aufrufe über Aufbau und Handhabung der Workbench, also der grafikorientierten Benutzeroberfläche. Für das Löschen einer Datei ist dagegen die 'Dos'-Library zuständig. Durch die Verteilung der Aufgaben wird Ordnung im System geschaffen.

Das Aufrufen von Routinen

Libraries bestehen aus einer Anzahl von Routinen. Um eine Routine in einer Library verwenden zu können, muß sie zunächst geöffnet werden. Dies kann man mit einer Funk-

tion erreichen, die sich in der 'Exec'-Library befindet. Danach können alle Routinen, die sich in der zuvor geöffneten Library befinden, aufgerufen werden. Dabei eignet sich die Programmiersprache C am besten, denn die Funktionen fast aller Libraries (mit Ausnahme der 'Exec' Sammlung) wurden in dieser Sprache erstellt. Der logische Schluß daraus läßt sich natürlich leicht ziehen: C ist angesagt. Natürlich können auch andere Programmiersprachen verwendet werden, nur ist das Aufrufen der Library-Funktionen nicht ganz so einfach wie in C.

Library-Vielfalt

Das Amiga-System besitzt eine ganze Reihe von Libraries, die vom System selber oder durch eigene Programme angesprochen werden können. Insgesamt besitzt das Amiga-Betriebssystem 16 solcher Libraries, die unterschiedlichste Aufgaben haben: Von 'Clist' über 'Console', 'Diskfont', 'Dos', 'Exec', 'Graphics', 'Icon', 'Intuition', 'Layers', 'Mathffp', 'Mathieeedoubbas', 'Mathieeesingbas', 'Mathtrans', 'Potgo', 'Timer' bis hin zur 'Translator'-Library. Welche Aufgaben die einzelnen Sammlungen von Programm-routinen erfüllen, ist nicht einfach zu erklären, denn manche Routinen reichen so sehr in den Hardwarebereich hinein, daß sie nur von einem Amiga-Fachmann verstanden werden kön-

nen. Einen Laien könnten hier enorme Verständnisprobleme erwarten.

Welche Library wofür?

Dennoch möchte ich etwas näher auf einzelne Libraries eingehen. Jede Sammlung beinhaltet also eine Reihe von Routinen, die ein bestimmtes Aufgabengebiet abdecken sollen. Entsprechend ist auch die Größe der einzelnen Libraries unterschiedlich; manche besitzen nur eine einzige Routine (Translator), andere dagegen mehr als 80 (Exec).

Die 'Exec' Library beinhaltet wohl die wichtigsten Routinen; sie stellt auch den Kern des gesamten Betriebssystems dar. Unter anderem verwaltet sie die Handhabung der anderen Libraries. Dies soll keineswegs die Routinen anderer Programmsammlungen in irgendeiner Weise schmälern, sondern erklärt lediglich die größere Priorität gegenüber anderen Libraries. 'Exec' beinhaltet beispielsweise eine Routine, die es dem Programmierer erst ermöglicht, eine andere Library zu verwenden. Denn wie Sie wissen, muß eine Library geöffnet werden, bevor man die einzelnen Funktionen in einem eigenen Programm verwenden kann. Neben dieser Funktion beinhaltet 'Exec' natürlich noch eine ganze Reihe weiterer

Funktionen für die Speicherverwaltung, Multitasking-Verwaltung oder für die Datenein- und ausgabe.

Eine weitere sehr komplexe Sammlung von Routinen ist die Library mit der Bezeichnung 'Intuition'. Es ist wohl die bekannteste Bibliothek im Amiga-Betriebssystem, die unter anderem alle Routinen der Workbench, also die Verwaltung von Fenstern, Bildschirmen, Gadgets usw. beinhaltet.

Die 'Graphics'-Library gehört ebenfalls zu den etwas größeren; sie enthält Routinen, die sich mit der Animation von Objekten, bestimmten Speicherplatz-Reservierungen oder Farbzuzuweisungen beschäftigen. Generell kann man sagen, daß sich 'Graphics' mit der globalen Grafikerstellung auf dem Amiga beschäftigt.

Eine weitere sehr nützliche Library nennt sich 'Dos' (Disk Operation System). Sie enthält Routinen, die von den Befehlen der Benutzeroberflä-

chen genutzt werden. So beinhaltet diese Library diejenige Funktion, die den Disketteninhalt (also das Directory) anzeigt, oder die eine Datei aufruft und löscht.

'Icon' beschäftigt sich ausschließlich mit diesen Bildobjekten. Sie wissen: Als Icon bezeichnet man beispielweise das Symbol, welches das Laufwerk auf dem Workbench-Bildschirm darstellt.

Die 'Layers'-Library beinhaltet Routinen, von denen die Überlagerung von Bildelementen verwaltet wird. Sie ist also eng mit der 'Intuition'-Programmsammlung verknüpft. Um Ihnen die Aufgabe von 'Layers' zu erläutern, möchte ich ein wenig weiter ausholen. Wenn Sie beispielweise auf dem Workbench-Bildschirm mehrere Fenster öffnen, besteht die Möglichkeit, daß sich einige Fenster überlagern. Die Aufgabe von 'Layers' besteht nun darin, den Inhalt unter dem verdeckten Fenster zu bewahren und bei Bedarf sichtbar zu

machen. Nähere Hinweise können im Intuition-Kurs in diesem Heft unter dem Abschnitt 'Refresh' nachgelesen werden.

Die Libraries 'Mathffp', 'Mathieedoubbas', 'Mathieeesingbas' und 'Mathtrans' beinhalten mathematische Grundelemente wie die Behandlung von Addition oder Division, aber auch die trigonometrischen Funktionen Sinus, Cosinus und Tangens. Auf die Unterschiede der 'mathematischen Libraries' möchte ich zu diesem Zeitpunkt noch nicht eingehen, da dies zu weit in das binäre Zahlensystem des Computers hineinführen würde.

Geniale Organisation

Die Betriebssystem-Programmierer des Amiga haben mit der Organisation des Systems über Libraries einen völlig neuen Weg beschritten. Die verschiedenen Programmsammlungen arbeiten problemlos zusammen.

(AK)

ASH Computerware

AMIGA™ Soft- und Hardware

Hardware: (fordern Sie unser Info an)

Amiga 500/1000/2000	auf Anfrage
Speichererweiterung 500/1000/2000	auf Anfrage
3,5" ext. Laufwerk (NEC) 500/1000/2000	475,—
5,25" ext. Laufwerk (umschaltbar)	559,—
Festplatten 20-80 MB	ab 2.495,—
Druckerkabel 2 mtr.	37,—

Software: (200 Titel vorrätig — fordern Sie unsere Liste an)

Flightsimulator II	95,—
De Luxe Paint II	269,—
Surgent (Doktorspiel)	149,—
Sinbad	105,—

und viele mehr zu userfreundlichen Preisen!
(unverb. empf. Verk. Preise)

Händleranfragen erwünscht!

ASH Computerware Wendlinger Robert

von-Brandl-Straße 15 · 8229 Laufen · Tel. (08682) 1327
Vertrieb Österreich: EMC Handelsagentur,
Margarethenstr. 21 · A-1040 Wien · Tel. 222-587 5400



DOLPHIN® SOFTWARE

bekannt durch DolphinDOS, bietet für Amiga 1000 an:

NEC 1036A-Drive 3 1/2", Metallgehäuse (18x10x3 cm), Kabel mit Stecker (60 cm) **nur 369,—**

NEC 1036A-Dual-Drive 3 1/2" (2x), Kunststoffgehäuse (20x11x8 cm), Kabel mit Stecker (60 cm) **nur 649,—**

TEAC-FD 55 F Drive 5 1/4", 2x80 Tr. AmigaDOS, 2x40 Tr. MS-DOS, Metallgehäuse, Kabel mit Stecker **nur 499,—**

256-KB-Memory erweitert Ihren Amiga auf 512 KB **nur 129,—**

Alle vier Produkte sind sofort anschlussfertig für Amiga 1000 und 100 % kompatibel (Garantie). Dolphin, Amiga und MS-DOS sind eingetragene Warenzeichen.

NATHAN, der Weise

Das intelligente Kurvendiskussionsprogramm, das sehr viel bietet:

Nullstellen finden, Minimaxrechnungen ableiten, integrieren, Rotationskörper berechnen, verschiedene Regressionen, Lösen von Gleichungssystemen mit bis zu 10 (!) Unbekannten.

Funktionseingabe sowohl Standard als auch UPN, welche dann sogar programmierbar ist.

Und selbstverständlich kann alles auch auf den Drucker ausgegeben werden.



Preis **nur 99,—**

★★★★ WIR SUCHEN

Amiga-Programmierer, die Interesse an einem guten Verdienst durch Softwareerstellung haben. Interessenten bitte schriftlich melden.

Bestellungen bitte schriftlich an:
DOLPHIN SOFTWARE, Jan Bubela, Eigenoffstr. 19, 6000 Frankfurt 1
Tel. 0 69 / 44 65 73. Versand (innerhalb von 48 Stunden) per NN + 8 DM
Porto, bei Scheck + 5 DM Porto. Laden mittwochs von 10-12 und 14-18
Uhr. Gratis-Info gegen adressierten Freiumschlag (80 Pf).

THE FAERY TALE ADVENTURE:



Ein Abenteuer aus dem Land der Feen und Zauberer.

Es war einmal vor langer Zeit im Lande Holm. Dort lebten drei Brüder in einer kleinen Stadt namens Tambrü.

Julian, der älteste, war bekannt für seinen Wagemut. Dem zweiten Bruder Phillip hingegen hatten die guten Feen des Landes neben der Klugheit auch das Glück in die Wiege gelegt. Zuvorkommend und gütig war Kevin, der Jüngste.

In den Tagen der großen Finsternis begab es sich, daß ein böser Zauberer die Töchter des Königs entführte. Eingesperrt in ein finsternes Verlies in den unzugänglichen Bergen des Landes schmolz die Hoffnung auf Rettung dahin. Unsagbar traurig war der König ob des Verlustes seiner Kinder. Die Mächte der Finsternis regierten das Land und das Leben der Menschen wurde von Tag zu Tag unerträglicher.

Nur eine Hoffnung gab es: Ein Talisman und fünf goldene Statuen konnten zusammen die Kräfte der bösen Magier brechen. Aber wohin waren die Kleinode verschwunden? In wessen seelenlosen Händen ruhten die einzigen Waffen, mit denen die Welt noch gerettet werden konnte?

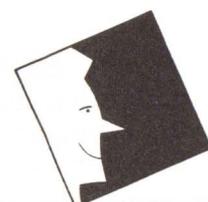
Also verließ Julian tapfer und entschlossen seine Heimatstadt, um die Welt – und natürlich die Königstochter – zu retten. Eh(r)e und Ruhm winkten ihm als Preis, ein schrecklicher Tod in den Klauen von Hexen, Knochenmännern oder Geistern war die Alternative.

Der Ausgang dieser Sage ist offen. Ob Julian und seine Brüder tatsächlich die bösen Geister überwinden und schließlich die Töchter des Königs erringen, hängt allein von Ihnen ab.

'Faery Tale', die Geschichte aus dem Lande der Feen und Märchen,

ist eine Mischung aus Adventure, Rollenspiel und Action – die eindeutig beste, die es bisher gibt. Gesteuert wird das Spiel mit der Maus und/oder der Tastatur. Die Graphik, Animation und Sound nutzen den Vorsprung, den der Amiga anderen Rechnern gegenüber auf diesen Gebieten hat. In diesen Punkten kann 'Faery Tale Adventure' mit jedem Arcadegame mithalten. Besser als alles bisher Dagewesene ist die Vielzahl der Landschaften, Gebäude und Laby-

rinthe. Mehr als 120 x 150 Bildschirme umfaßt die Märchenwelt mit ihren zwei Ebenen. Unser Held betritt diese Welt mit nichts als einem kurzen Schwert in den Händen. Bis er durch Glück und Kampf genügend Gold und Waffen hat, um auf der 'Spirit Plane' dem Magier gegenüberzutreten, der den Talisman gestohlen hat, ist es ein langer Weg.



SURGEON

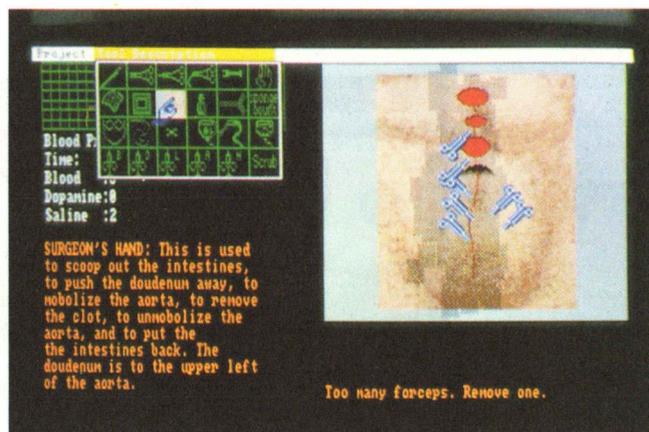
Schwester, Skalpell bitte!

Seine Hauptszenerie ist der Operationssaal. Auf dem Bildschirm sieht man einen Körper vor sich liegen, an dem man sein Glück als Mediziner versuchen kann. Dafür wird eine stattliche Anzahl von Instrumenten und Werkzeugen zur Verfügung gestellt. Neben diversen Spritzen, Infusionen, Transfusionen und anderen Operations-Utensilien hat man natürlich auch ein Skalpell zur Verfügung, mit dem der Patient aufgeschlitzt wird. Wer jetzt sorglos den Kranken masakriert, wird mit einem markerschütternden Todesschrei des dahinscheidenden Patienten „bestraft“. Überhaupt muß man bei „The Surgeon“ sehr sorgfältig vorgehen, um nicht am vorzeitigen Ableben von Mr. Jones (so heißt der Patient) schuld zu sein. Hat man Glück, kann man Mr. Jones auch ohne Operation retten, wenn man sich nach der Sichtung der Röntgenbilder dazu entschließt, den Patienten vorerst nur zu beobachten.

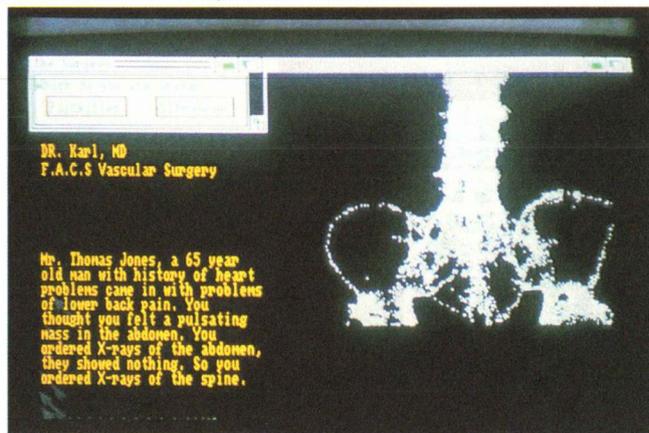
Das Programm ist insgesamt gut gemacht und recht wirklichkeitsgetreu gehalten. Auf einem anderen Blatt steht der Preis: Mit 149,- DM muß man schon sehr tief in die Tasche greifen, um sich als Hobby-Chirurg und Retter der Kranken zu beweisen. Man orientierte sich dabei wohl am Einkommen eines Chirurgen.

Für alle, die schon immer in die Fußstapfen von Doktor Brinkmann treten wollten, ist das richtige Spiel auf den Markt gekommen: „The Surgeon“, zu deutsch „Der Chirurg“.

Und genau der ist auch die Hauptfigur in diesem Spiel. Das Wort „Spiel“ ist in diesem Zusammenhang allerdings vorsichtig zu gebrauchen; darauf weist auch die Anleitung zu „The Surgeon“ hin. Denn das ganze soll nicht nur „just for fun“ sein,



Die Blutung gezielt stoppen.



Ein Röntgenbild gibt nähere Hinweise.

sondern auch zum besseren Kennenlernen der menschlichen Innereien dienen. Für das reine Spaß-Spiel ist die Thematik freilich auch recht makaber. Wer aber mit der Materie beruflich oder in der Schule/Universität zu tun hat, könnte sogar eine nützliche Hilfe zum Lernen in dem Programm finden.

Kategorie: Lernspiel

Besonderheiten:
Sehr realitätsnah, ganz neue Spielidee.

Hersteller: ISM



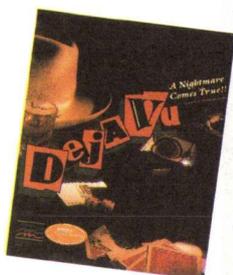
Vertrieb: PDC
Louisenstraße 115
6280 Bad Homburg
Tel. 0 61 72 / 2 47 48

Preis: DM 149,-

DEJA VU

Ein Alptraum wird wahr!

Ein wahrhaft schrecklicher Gedanke: Sie wachen mit einem schweren Kopf auf und wissen nicht, wo Sie sind und, was noch schlimmer ist, wer sie sind. Wird es Ihnen gelingen, Ihre Identität zurückzuerlangen?



Sie erwachen mit einem Gefühl der Benommenheit. Ihr Kopf pocht wild und auf Ihrer Handfläche bemerken Sie Blut. Trotzdem können Sie keine Wunde feststellen, nur ein Nadelein-

Mantel und die Pistole und gehen aus dem Bad. Sie kommen in einen leeren Barraum, gehen eine Treppe hinauf und erreichen einen langen Gang. An der Wand hängen Fotografien von Boxern. Ihre Erinnerung scheint

beeilen Sie sich, denn man ist Ihnen auf den Fersen...

DEJA VU ist ein Grafikadventure mit besonders hohem Bedienungs-komfort. Nichts muß eingetippt wer-



Bild 1: Unverhofft stolpert man über eine Leiche.

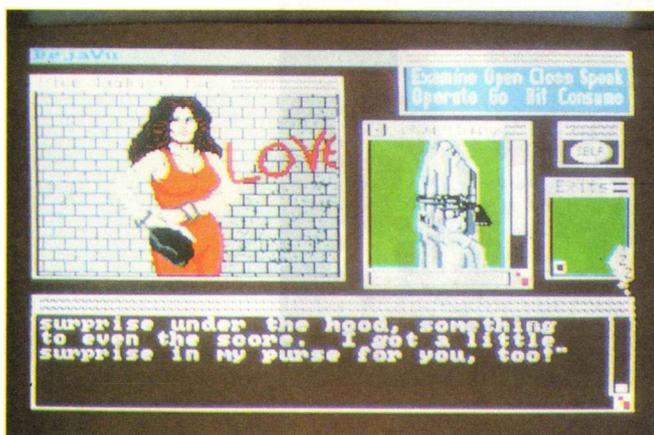


Bild 2: Vor verruchten Frauen sollte man sich in acht nehmen.

stich am linken Unterarm macht sich unangenehm bemerkbar – an Ihrem Unterarm, aber wer sind Sie? Sie können sich an nichts erinnern. Langsam lichtet sich der Sie umgebende Nebel, und Sie sind wenigstens fähig, die Umwelt wahrzunehmen. Vor Ihnen ist eine Tür, an der Wand hängt ein Trenchcoat, darunter ein Schulterhalter mit einer 38er Spezial. In den Manteltaschen finden Sie etwas Geld, Zigaretten, eine coole Sonnenbrille und eine Brieftasche mit den Initialen 'J.S.'. Sie nehmen den

durchzubrechen – doch dann wieder das Nichts. Der Weg führt durch ein Büro vor eine verschlossene Tür mit der Aufschrift 'Privat'. Neugierig, wie Sie nun einmal sind, öffnen Sie sie mit dem Schlüssel, den Sie in Ihrer Manteltasche gefunden haben... Schrecklich: Ein offensichtlich toter Mann hängt über dem Schreibtisch, den Telefonhörer noch in der Hand. Im Hintergrund ist ein geschlossener Tresor. Was geht hier vor? Auf Ihrem weiteren Weg können Sie noch manche Überraschung erleben. Doch

den, denn alle Aktionen werden mit der Maus ausgeführt: Gehen, Untersuchen, Öffnen, Reden usw. geschieht mit einem Doppelklick. Will man einen Gegenstand mitnehmen, dann zieht man ihn mit dem Mauszeiger in ein 'Inventory'-Fenster und schon ist er jederzeit verfügbar. Genauso einfach ist das Auffinden der möglichen Bewegungsrichtungen, denn ein spezieller Plan zeigt alle Ausgänge.

Die Handlung von DEJA VU ist ebenso interessant wie spannend, die Texte sind spritzig und humorvoll. Zusammen mit der starken Bedienung: Ein wirklich ausgezeichnetes Adventure.

(mn)



Mind-Scape /
99, – / PDC/Softline

UNINVITED

Der ungebetene Gast

Völlig unbeabsichtigt geraten Sie einen Strudel von Ereignissen. Ein Unfall mit Ihrem Wagen veranlaßt Sie dazu, ein altertümliches Herrenhaus zu betreten, in dem sich allerlei seltsame Dinge ereignen.



Das letzte, woran Sie sich noch erinnern können, ist der Schrei Ihres kleineren Bruders, als sich Ihr Wagen dem Baum näherte. Sie wollten einer schemenhaften Gestalt ausweichen, als Sie die Kontrolle über den Wagen verloren und über die Fahrbahnbegrenzung hinausfuhren. Nun sind Sie alleine, Ihr Bruder ist sicher auf der Suche nach Hilfe. Der Geruch von ausgelaufenem Benzin sticht Ihnen in die Nase. Jetzt ist es wohl an der Zeit, den Wagen zu verlassen. Gedacht, getan – die nun folgende Explosion bestätigt Ihre Vermutung.

Nun stehen Sie vor einem großen Herrenhaus, das weit und breit das einzige Gebäude ist. Der Wind bläst scharf und Blitze zucken vom Himmel. Die Tür läßt sich leicht öffnen und fällt danach wieder knarrend ins Schloß: Sie sind gefangen. Die Tür wird von einer unsichtbaren Macht geschlossen gehalten, und Ihnen bleibt nur die Möglichkeit, den Raum, in dem Sie sich gerade befinden, zu untersuchen. Dabei stoßen Sie auf rätselhaft Gegenstände wie Pentagramme, Amulette und Bücher, die von mysteriösen Dingen handeln.

Nach DEJA VU ist UNINVITED das zweite Adventure von MINDSCAPE, das ausschließlich mit der Maus bedient wird. Der Bedienungs-komfort ist deshalb ebenso vorzüglich. Die Handlung ist wieder sehr spannend, in diesem Fall sogar reich-

lich mysteriös und unheimlich. Neu ist, daß einige Figuren aus dem Adventure animiert sind und deshalb von Zeit zu Zeit über den Bildschirm huschen.

UNINVITED ist also wiederum ein Spiel zum richtigen Entspannen. Zurückgelehnt in einen bequemen Sessel, hat man die volle Kontrolle über alle Aktionen. Wenn ein Gegenstand sich beim Anklicken verfärbt,

man kann sie einfach mitnehmen – man kann ja nicht wissen, wozu sie zu gebrauchen sind. Danach geht die Entdeckungsreise durch die vielen Räume des Hauses weiter, wobei man sich jedoch vor deren Bewohnern in Acht nehmen sollte, denn ein Zusammen-treffen hat meistens kein gutes Ende...

(mn)

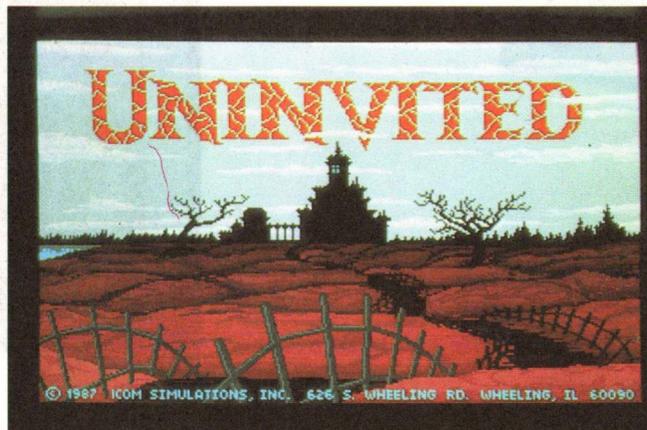


Bild 1: Düster und geheimnisvoll liegt das Haus in der Landschaft.

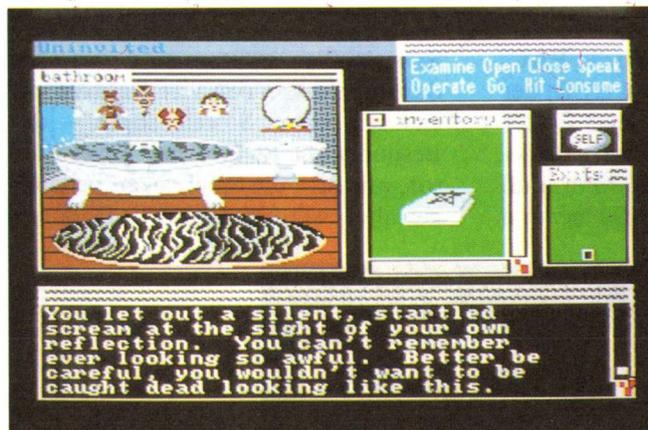


Bild 2: Ein Bad gefällig?

dann sollte man ihn näher betrachten, denn er könnte eine wichtige Information enthalten. Einige Gegenstände sollte man im Zweifelsfall ein-

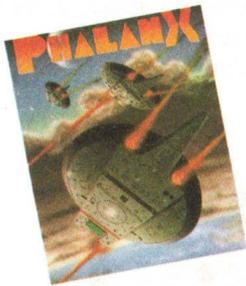


PHALANX

Ein starkes Ballerspiel für schnelle Joystick-Spezialisten ist das Weltraumspiel PHALANX. Dabei müssen sich ständig ändernde Formationen von angreifenden Raumschiffen bewältigt werden. Dies erfordert eine hohe Reaktionsfähigkeit, denn man weiß nie, von wo sie jeweils kommen. Besonders heimtückisch ist

es nämlich, wenn sie dem eigenen Raumschiff in den Rücken fallen. Allerdings hilft hier ein Scanner, der einen größeren Bildausschnitt darstellt. Als besondere Hilfe gibt es einen Schutzschirm, der für etwa zwanzig Sekunden aktiviert werden kann und das Raumschiff unverwundbar macht.

Wer sich ordentlich anstrengt, dem gelingt es sicher, sich in der HiScore-Liste zu platzieren. PHALANX ist ein reines Ballerspiel und sehr gut gemacht. Es bietet eine Menge Action und gute Grafik. Der Preis von 29,95 DM ist deshalb sehr günstig.



Angriffe von allen Seiten

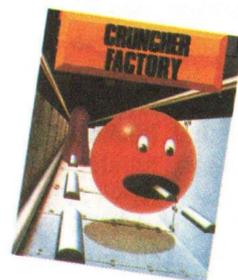
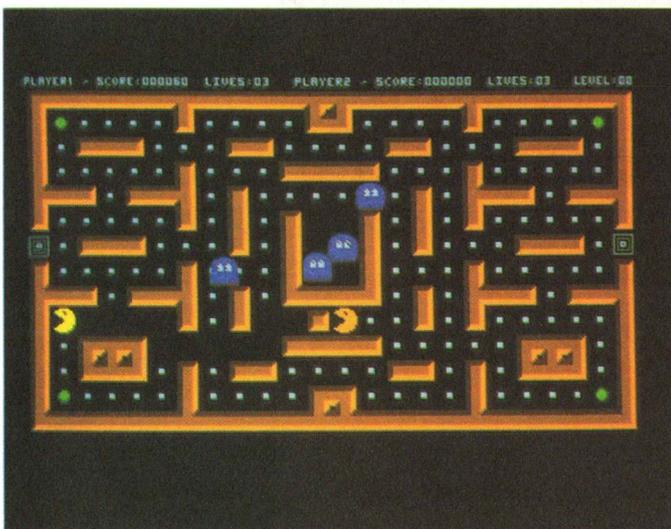
CRUNCHER FACTORY

Eine amüsante und schnelle PACMAN-Variante ist CRUNCHER, der zudem noch mit einem Level-Editor versehen ist. PACMAN dürfte für jedermann ein Begriff sein: Man muß in einem Labyrinth alle Steine und

Energiepillen aufessen und darf sich dabei nicht von den vier Geistern erwischen lassen. Soweit, sogut – die Besonderheit ist hier der Level-Editor, mit dem die Spielszenen beliebig gestaltet werden können. Dadurch

kann sich jeder ein Erfolgserlebnis verschaffen, indem er viele Energiepillen in den Level einbaut.

Die Grafik ist ausreichend gut, die digitalisierte Titelmelodie wieder einmal hörensenswert. Wer diesen Klassiker gerne für den AMIGA haben möchte, sollte die Gelegenheit wahrnehmen.



King-Soft / 29,90 / Cruncher Factory

Das 'Gecrunche' kann losgehen.

SPACE BATTLE

Wer kennt ihn nicht, den Spielhallenklassiker ASTEROIDS, der besonders durch seine Steuerung aufgefallen ist. Mit dem Joystick wird das Raumschiff rechts- bzw. links herum gedreht, ein Druck nach vorne beschleunigt es dann. Das Abbremsen ist nur möglich, indem kräftig gegengesteuert wird. Am Anfang führt diese gewöhnungsbedürftige Steuerung dazu, daß das Raumschiff in wild trudelnden Bewegungen über den Bildschirm zischt.

Die Spielidee besticht durch ihre Schlichtheit. Ein winziges Raumschiff schwebt durch den Weltraum und versucht, ihn von Asteroiden zu reinigen. Obwohl diese Aufgabe ei-



gentlich sehr nützlich ist, versuchen fremde Raumschiffe, dem Piloten das Leben zu erschweren. Um dies auszugleichen, gibt es den TEAM-Modus, bei dem zwei Spieler sich diese schwierige Aufgabe teilen. Allerdings sollten die Bewegungen vorsichtig ausgeführt werden, um den Mitspieler nicht allzu oft abzuschießen. Dies ist erst im COMPETITION-Mode angesagt, der natürlich dadurch seinen besonderen Reiz bekommt.

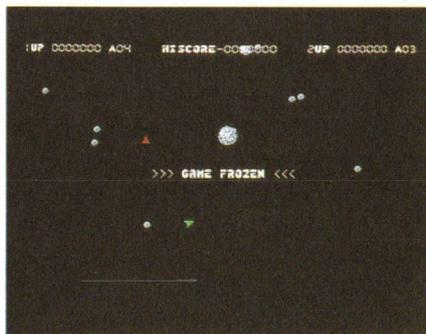


Bild 1: Ballern was das Zeug hält.

DEMOLITION

DEMOLITION ist eine etwas abgewandelte Variante von BREAK OUT, die sich beim Starten mit einer fetzigen Melodie vorstellt. Das Spiel an sich ist recht einfach: Mit einem Schläger wird eine Kugel gegen eine Steinwand gelenkt, die dadurch abgeräumt wird. Zusätzliche Punkte erlangt man, wenn es gelingt, die 'AMIGA'-Kugel abzulenken und die herumfliegenden Pacmans, Hähnchen usw. abzuschießen. Ohne Einfluß auf das Spiel ist der eindrucksvoll scrollende, dreidimensionale Hintergrund.

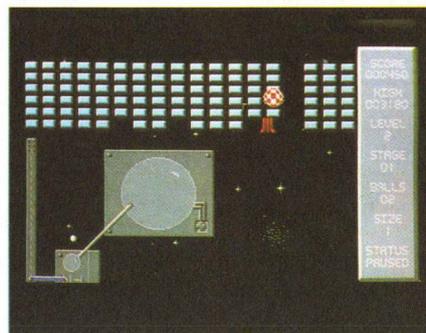
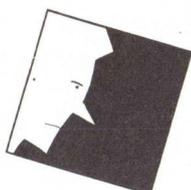


Bild 1: Geschicklichkeit und Strategie wird verlangt.

Eine große Hilfe bei unübersichtlichen Situationen ist der Hyperspace-Sprung, der das Raumschiff an einen zufälligen Ort transformiert. Während sich das Schiff am neuen Ort langsam kristallisiert, hat man Zeit, die Szene zu überblicken und gegebenenfalls gleich wieder 'wegzubeamen'.

Für knapp dreißig Mark erhält man mit SPACE BATTLE zwar kein neuartiges Spiel. Trotzdem ist es, besonders im COMPETITION-Mode, spannend.

FLIP FLOP

Hinter FLIP FLOP versteckt sich das bekannte Brettspiel Reversi oder Othello. Dabei geht es darum, möglichst viele gegnerische Steine einzuklemmen, die dann umgedreht und damit zu eigenen werden. FLIP FLOP bietet neun verschiedene Spielstufen, wobei jedoch erst die hinteren wirklich gut spielen, dafür aber auch viel Zeit benötigen. Die Smile-Grafik wirkt sehr naiv, die Farben sind viel zu grell. Optisch ist FLIP FLOP deshalb nicht recht überzeugend, was Freunde dieses Spiel jedoch nicht unbedingt abschrecken sollte.

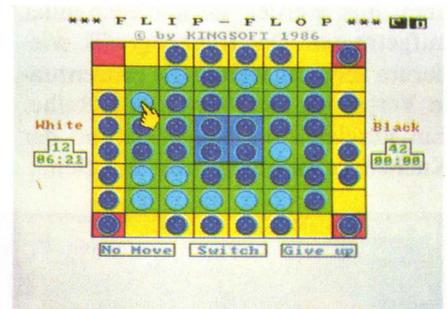
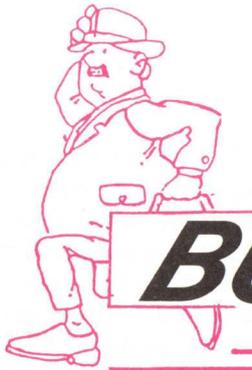


Bild 1: Nachdenken ist gefragt.





BUSINESS-GRAFIK

Jeder kennt das Problem: Wenn größere Mengen von Zahlen anfallen, sind sie meist recht unübersichtlich. Abhilfe schafft hier eine grafische Darstellung der Werte. Für diesen Zweck gibt es mehrere verschiedene Formen, wobei jede für eine bestimmte Aussage besonders geeignet ist. Das Programm BUSINESS_GRAFIK bietet Linien-, Balken-, Säulen- und Kuchengrafik zur Auswahl an. Die beste Übersicht zeigt die Liniengrafik: Obwohl grafisch recht unscheinbar, gibt sie den schnellsten Überblick der einzelnen Reihen und Spalten. Die optisch ansprechendere Form, die jedoch die gleiche Aussage zulässt, ist die Säulengrafik. Hier werden die Reihen perspektivisch hintereinander dargestellt. Sie ist sicher die schönste Darstellungsform. Den besten Vergleich der einzelnen Spaltenwerte einer Reihe erhält man mit der Balkengrafik, denn hier werden sie nebeneinander aufgetragen. Die Kuchengrafik wiederum zeigt am besten die prozentuale Verteilung der Wert einer Reihe. Ein Vergleich der Reihen ist mit der Kuchengrafik nur schwer möglich.

Start

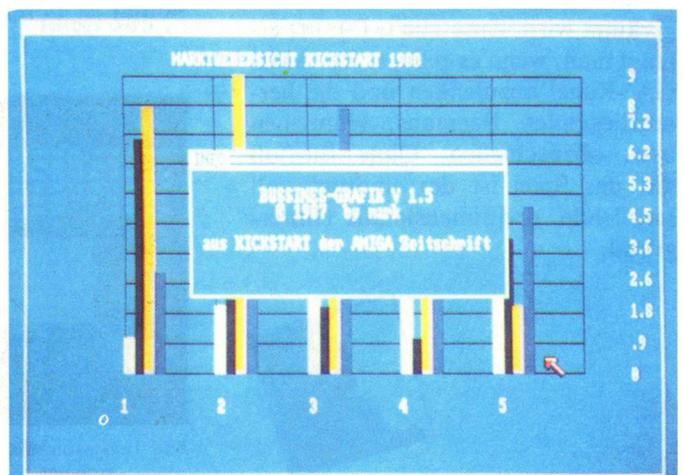
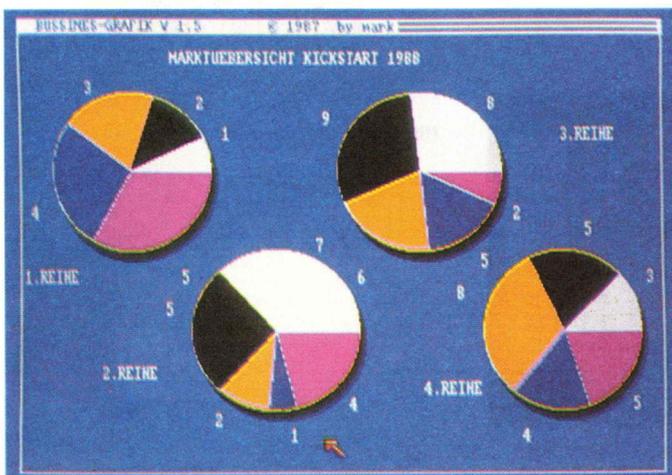
Nach dem Starten des Programms erscheint eine Einschaltmeldung, die mit der linken Maustaste weggeklickt wird. Danach werden die Daten entweder geladen oder eingegeben. Vier Reihen je 20 Spalten sind vorgesehen. Diese Kapazität kann mit geringem Aufwand vergrößert werden; allerdings sind bald die Grenzen erreicht, ab denen es nicht mehr sinnvoll ist, mehr Daten anzuzeigen.

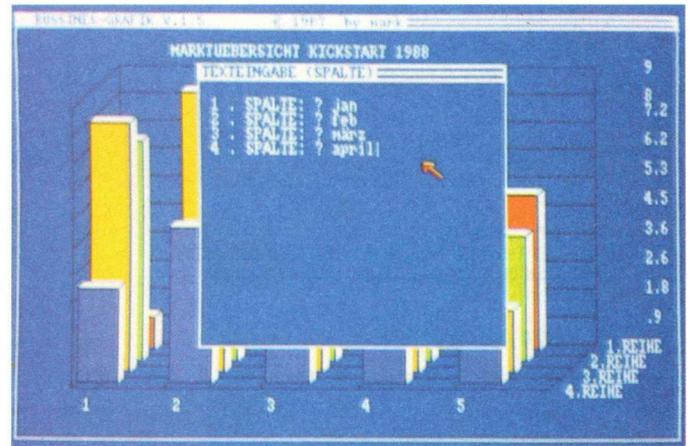
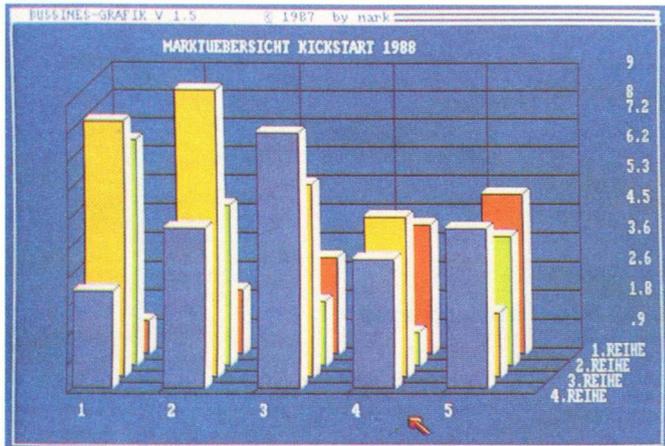
Nachdem die Daten vorhanden sind, kann eine der Darstellungsformen angewählt werden. Das Bild baut sich dann recht schnell auf – einzig die Kuchengrafik läßt sich eine Menge Zeit. Mit dem Menüpunkt OPTIONS können noch einige Details am Aussehen des Bildes geändert werden. Mit Y-MAX kann die Obergrenze der ersten drei Darstellungsarten geändert werden. Das Programm setzt normalerweise den größten y-Wert ein, dies ist allerdings nicht immer die beste Wahl. Die weiteren Eingaben betreffen die Beschriftung der Reihen und Spalten sowie die Titelzeile der Grafik. Alle Eingaben erfolgen in einem speziellen Fenster,

das danach wieder verschwindet. Auf dem Bildschirm verbleibt dann ein schwarzes Loch, denn der Speicher reicht nicht aus, um für das Ausgabefenster das entsprechende Flag zu setzen. Allerdings ist nach einer Eingabe sowieso ein Neuaufbau der Grafik mit den neuen Eingabewerten nötig. Deshalb stellt das Programm das zuletzt angewählte Bild neu dar.

Programmaufbau

Der erste Teil des Programms legt eine neue Screen an und öffnet das Hauptfenster. Danach werden die Menüeinträge und einige Variablen dimensioniert und vorbelegt. Nach der Aktivierung des Menüs kommt die obligatorische Endlosschleife. In ihr sind allerdings noch zwei Abfragen integriert: Beim ersten Lauf des Programms wird die INFO-Routine aufgerufen; während des weiteren Programms wird nach einer Eingabe ein Refresh durchgeführt, der – wie oben beschrieben – die letzte Grafik wieder aufbaut. Der Rest des Programms besteht nun nur noch aus Unterprogrammen, von denen MENUAUSWAHL das wichtigste ist.





Hier werden die Menü-Aktionen ausgewertet und in die entsprechenden Routinen verzweigt: (MN)

Viel Spaß mit dem Programm!

EINGABE:

Eingabe der Werte

DIA:

Darstellung der der LINIEN-, BAL- KEN- und SÄULENGRAFIK

KU:

Darstellung der KUCHENGRAFIK

INFO:

Infobox

RASTER:

zeichnet das Hintergrundraster für die Grafiken

MINIMAX:

ermittelt den größten y-Wert

SPEICHERN:

speichert Anzahl der Reihen, Spalten und die Werte

LADEN:

lädt die Werte wieder ein

TITEL:

stellt die Titelzeile dar

TITELTEXT:

Eingabe des Titels

REIHENTEXT:

Eingabe der Texte für die einzelnen Reihen

SPALTENTEXT:

Eingabe des Textes für die Spalten

```

REM   GRAFIK V 1.5
REM
SCREEN 4,640,240,4,2
WINDOW 13,"BUSINESS-GRAFIK V 1.5 (C) 1987 by mark",(0,0)-(630,220),2,4
WINDOW 13
WINDOW OUTPUT 13

MENU 1,0,1,"DESK  "
MENU 1,1,1,"INFO  "
MENU 2,0,1,"DATEI  "
MENU 2,1,1,"EINGEBEN "
MENU 2,2,1,"LADEN  "
MENU 2,3,1,"SPEICHERN"
MENU 2,4,0,"-----"
MENU 2,5,1,"QUIT  "
MENU 3,0,1,"GRAFIK "
MENU 3,1,1,"DIAGRAMM "
MENU 3,2,1,"BALKEN  "
MENU 3,3,1,"SAEULE  "
MENU 3,4,1,"KUCHEN  "
MENU 4,0,1,"OPTIONEN "
MENU 4,1,1,"Y-MAX  "
MENU 4,2,1,"TITEL-TEXT "
MENU 4,3,1,"REIHEN-TEXT "
MENU 4,4,1,"SPALTEN-TEXT"

VARIABLEN:
DIM W(4,20),WL(100),RS(4),SS(20)
RS(1)="1.REIHE": RS(2)="2.REIHE": RS(3)="3.REIHE": RS(4)="4.REIHE"
FOR SPALTE=1 TO 20: SS(SPALTE)=STR$(SPALTE): NEXT SPALTE
TEXT$="MARKTUEBERSICHT KICKSTART 1988"

ON MENU GOSUB MENUAUSWAHL
MENU ON
START=-1

WHILE -1
  IF START THEN
    START=0
    GOSUB INFO
  END IF
  IF REFRESH THEN

```

```

REFRESH=0: M=3: T=LAST: GOSUB LABEL
END IF
WEND

MENUAUSWAHL:
M=MENU(0)
T=MENU(1)
LABEL:
IF M=1 THEN GOSUB INFO
IF M=2 AND T=5 THEN MENU RESET:SCREEN CLOSE 4:END
IF M=2 THEN ON T GOSUB EINGABE,LADEN,SPEICHERN
IF M=3 THEN
LAST=T
ON T GOSUB DIA,DIA,DIA,KU
END IF
IF M=4 THEN ON T GOSUB MAXY,TITELTEXT,REIHENTEXT,SPALTENTEXT
RETURN

INFO:
WINDOW 14,"INFO",(170,70)-(470,130),2,4
PRINT : PRINT " BUSINESS-GRAFIK V 1.5"
PRINT " (C) 1987 by mark"
PRINT
PRINT " aus KICKSTART der AMIGA Zeitschrift"
WHILE MOUSE(0)<>-1
WEND
REFRESH=-1
WINDOW CLOSE 14
RETURN

REIHENTEXT:
WINDOW 14,"TEXTEINGABE (REIHE)",(180,70)-(460,130),2,4
PRINT
FOR REIHE=1 TO ANZREIHEN
PRINT REIHE;" REIHE: ";:INPUT R$(REIHE)
NEXT REIHE
REFRESH=-1
WINDOW CLOSE 14
RETURN

SPALTENTEXT:
WINDOW 14,"TEXTEINGABE (SPALTE)",(180,30)-(460,170),2,4
PRINT
FOR SPALTE=1 TO ANZSPALTEN
PRINT SPALTE;" SPALTE: ";:INPUT S$(SPALTE)
NEXT SPALTE
REFRESH=-1
WINDOW CLOSE 14
RETURN

MAXY:
WINDOW 14,"Y-MAX",(250,90)-(390,110),2,4
PRINT "ALTER WERT: ";YMAX
INPUT "NEUER WERT: ";WERT$
IF WERT$<>"" THEN YMAX=VAL(WERT$)
REFRESH=-1
WINDOW CLOSE 14
RETURN

TITELTEXT:
WINDOW 14,"TITEL-TEXT",(10,10)-(600,20),2,4
INPUT TEXT$
REFRESH=-1
WINDOW CLOSE 14
RETURN

TITEL:
LOCATE 2,(70-LEN(TEXT$))/2: PRINT TEXT$
RETURN

EINGABE:
WINDOW 14,"EINGABE DER WERTE",(10,10)-(200,100),1,4

INPUT "WIEVIEL REIHEN: ",ANZREIHEN
INPUT "WIEVIEL SPALTEN: ",ANZSPALTEN

FOR REIHE=1 TO ANZREIHEN
SU(REIHE)=0
FOR SPALTE=1 TO ANZSPALTEN
PRINT "WERT (";REIHE;" ";SPALTE;" ) = ";
INPUT E$
IF E$="" THEN
W(REIHE,SPALTE)=0
ELSE
W(REIHE,SPALTE)=VAL(E$)
END IF
SU(REIHE)=SU(REIHE)+W(REIHE,SPALTE)
NEXT SPALTE
NEXT REIHE

MENU 3,0,1
GOSUB MINIMAX
REFRESH=-1
WINDOW CLOSE 14
WINDOW OUTPUT 13
RETURN

DIA:
CLS
GOSUB TITEL
YABS=YMAX-YMIN
XM=450: YM=150
DX1=100: DX2=DX1+XM: DY1=20: DY2=DY1+YM
XA=XM/ANZSPALTEN
GOSUB RASTER
YABST=YM/10
YA=YM/YMAX
IF YMIN>=0 THEN
NP=DY2
ELSE
NP=INT((YA*YMAX)/YABST)*YABST+DY1
END IF

FOR REIHE=1 TO ANZREIHEN
IF T=1 THEN
Y1=NP-YA*W(REIHE,1)
X1=DX1
FOR SPALTE=2 TO ANZSPALTEN
X2=DX1+XA*(SPALTE-1)
Y2=NP-YA*W(REIHE,SPALTE)
LINE (X1,Y1)-(X2,Y2),REIHE
X1=X2: Y1=Y2
NEXT SPALTE
END IF
IF T=2 THEN
BREITE=10
Y1=NP-YA*W(REIHE,1)
X1=DX1+(REIHE-1)*BREITE
FOR SPALTE=1 TO ANZSPALTEN
Y1=NP-YA*W(REIHE,SPALTE)
X1=DX1+XA*(SPALTE-1)+(REIHE-1)*BREITE
LINE (X1,Y1)-(X1+10,NP),REIHE,BF
NEXT SPALTE
END IF
IF T=3 THEN
BREITE=.45*XA
TIEFE=6
HOPS=12
LINE (DX1,NP)-(DX2-(REIHE-1)*HOPS,NP),2
LOCATE (NP*24/200)+2,((DX2-(REIHE-1)*HOPS)*80/640)+2: PRINT R$(REIHE)

FOR SPALTE=1 TO ANZSPALTEN
Y1=NP-YA*W(REIHE,SPALTE)
X1=DX1+XA*(SPALTE-1)-(REIHE-1)*HOPS
AREA (X1-TIEFE,Y1+TIEFE/2)
AREA (X1,Y1)
AREA (X1+BREITE,Y1)
AREA (X1+BREITE,NP)
AREA (X1+BREITE-TIEFE,NP+TIEFE/2)
AREA (X1+BREITE-TIEFE,Y1+TIEFE/2)
AREA (X1-TIEFE,Y1+TIEFE/2)
AREAFILL 0
LINE (X1+BREITE-TIEFE,Y1+TIEFE/2)-(X1+BREITE,Y1),2
LINE (X1-TIEFE,Y1+TIEFE/2)-(X1+BREITE-TIEFE,NP+TIEFE/2),2,BF
LINE (X1-TIEFE+1,Y1+TIEFE/2+1)-(X1+BREITE-TIEFE-2,NP+TIEFE/2-1),REIHE+8,BF
NEXT SPALTE
NP=NP+HOPS/2
END IF
NEXT REIHE
RETURN

KU:
CLS
GOSUB TITEL
R=80: ER=.5: LF=1: MX=110: MY=70
FOR REIHE=1 TO ANZREIHEN
RA=6.28/SU(REIHE): ZF=0: W1=0
CIRCLE (MX+5,MY+5),R,2,,ER
PAINT (MX,MY),2
LINE (MX,MY)-(MX+R,MY),1
FOR SPALTE=1 TO ANZSPALTEN
ZF=ZF+1: IF ZF>7 THEN ZF=1
W2=W1+W(REIHE,SPALTE)*RA
CIRCLE (MX,MY),R,LF,-W1,-W2,ER
W=(W2+W1)/2
PX=MX+(.9*R*COS(W))
PY=MY-(.5*(.9*R*SIN(W)))
IF ABS(W2-W1)>.01 THEN

```

```

PX=MX+(.9*R*COS(W))
PY=MY-(.5*(.9*R*SIN(W)))
PAINT (PX,PY),ZF,1
PX=MX+(.9*(R+10)*COS(W))
PY=MY-(.5*(.9*(R+10)*SIN(W)))
PSET (PX,PY),ZF
LOCATE (PY*24/200)-(SGN(SIN(W))-1)*3,(PX*80/640)+SGN(COS(W))*2-1
PRINT W(REIHE,SPALTE)
END IF
W1=W2
NEXT SPALTE
MX=MX+140 : IF MY=70 THEN MY=150 ELSE MY=70
IF REIHE=1 THEN
  X=1: Y=16
END IF
IF REIHE=2 THEN
  X=11: Y=22
END IF
IF REIHE=3 THEN
  X=67: Y=7
END IF
IF REIHE=4 THEN
  X=50: Y=23
END IF
LOCATE Y,X: PRINT R$(REIHE)
NEXT REIHE
RETURN

RASTER:
IF M=3 AND (T=5 OR T=6) THEN OFFSET=XA ELSE OFFSET=0
J=0: REM      x-Parallelen
FOR I=DY1 TO DY2 STEP (DY2-DY1)/10
  LINE (DX1,I)-(DX2+OFFSET,I),2
  IF T=3 THEN LINE (DX1,I)-(DX1-48,I+24),2

  Y=(25/200)*I
  LOCATE 24-Y,75: PRINT INT((YMIN+J*(YABS/10))*10)/10
  J=J+1
NEXT I
REM          y-Parallelen
J=0
FOR I=DX1 TO DX2+OFFSET STEP XA
  J=J+1
  LINE (I,DY1)-(I,DY2),2
  LOCATE 24-(T+3)*2,(I*80/640)+(T+3)*5: IF J<ANZSPALTEN+1 THEN PRINT S$(J)
  IF T=3 THEN LINE (I,DY2)-(I-48,DY2+24),2
NEXT I
REM          raemlicher Gitterteil
IF T=3 THEN
  LINE (DX1-48,DY1+24)-(DX1-48,DY2+24),2
  LINE (DX1-48,DY2+24)-(DX2-48,DY2+24),2
END IF
RETURN

```

```

MINIMAX:
FOR REIHE=1 TO ANZREIHEN
FOR SPALTE=1 TO ANZSPALTEN
  WL(((REIHE-1)*ANZSPALTEN)+SPALTE)=W(REIHE,SPALTE)
NEXT SPALTE
NEXT REIHE
ANZAHL=ANZREIHEN*ANZSPALTEN
FOR I=1 TO ANZAHL
  FOR J=1 TO ANZAHL-1
    IF WL(I)<WL(J) THEN SWAP WL(I),WL(J)
  NEXT J
NEXT I
YMIN=0: YMAX=WL(ANZAHL)
YABS=YMAX-YMIN
IF YMAX<0 THEN YABS=-YABS
RETURN

SPEICHERN:
WINDOW 14,"DATEI SPEICHERN",(240,85)-(400,115),2,4
PRINT :PRINT " BITTE WARTEN ..."
OPEN "O",#1,"DATEN.DAT"
PRINT#1,ANZREIHEN
PRINT#1,ANZSPALTEN
FOR REIHE=1 TO ANZREIHEN
  FOR SPALTE=1 TO ANZSPALTEN
    PRINT#1,W(REIHE,SPALTE)
  NEXT SPALTE
NEXT REIHE
CLOSE 1
REFRESH--1
WINDOW CLOSE 14
RETURN

LADEN:
WINDOW 14,"DATEI LADEN",(240,85)-(400,115),2,4
PRINT :PRINT " BITTE WARTEN ..."
OPEN "I",#1,"DATEN.DAT"
INPUT#1,ANZREIHEN,ANZSPALTEN
ANZAHL=ANZREIHEN*ANZSPALTEN
FOR REIHE=1 TO ANZREIHEN
  SU(REIHE)=0
  FOR SPALTE=1 TO ANZSPALTEN
    INPUT#1,W(REIHE,SPALTE)
    SU(REIHE)=SU(REIHE)+W(REIHE,SPALTE)
  NEXT SPALTE
NEXT REIHE
CLOSE 1
GOSUB MINIMAX
REFRESH--1
WINDOW CLOSE 14
RETURN

```

★ Amiga ★

512-KByte-Karte mtr512

- extern steckbar
- durchgeschleifter Bus
- statisch RAM/Eprom-bestückbar
- aufrüstbar in 64-KByte-Schritten
- Adressbereich in 512-KByte-Schritten frei wählbar

Leerplatine (Industriequalität, Aufbauanleitung)	110,- DM
Bausatz (ohne RAMs/Eproms)	178,- DM
Fertigkarte (ohne RAMs/Eproms)	248,- DM
RAMs 32K x 8 ab 10	25,- DM
Platinenstecker 86pol.	25,- DM

Sonderaktion: bis 6.7.87

Leerplatine 98,- DM

Achtung!! Durch den Einsatz von Kickstart-Eproms kann die Karte als Betriebssystemplatine benutzt werden.

Zweitlaufwerk mtd880 voll kompatibel, mit NEC 1035 bestückt	nur noch 369,- DM
Bausatz (komplett, mit Gehäuse und Laufwerk)	299,- DM
Platine (Industriequalität, Aufbauanleitung)	12,- DM
Gehäuse (beige, für NEC 1035 passend)	22,- DM

Software zu Hammerpreisen

Aegis Draw Plus	495,- DM
Deluxe Video 2	275,- DM
New CLI Mate	76,- DM
Dynamic CAD	1065,- DM
The Pawn	66,- DM

andere Software auf Anfrage.

Es sind weitere Produkte für den Amiga in Vorbereitung

- Uhrenkarte
- Autokonfigkarte
- Eprommer
- 4-MByte-Karte
- Video-Digitizer

Fordern Sie
Unterlagen an.
Händleranfragen
erwünscht.
unv. empf. Verk. Preise



Ralf Tröps - Computertechnik - 5040 Brühl - Pingsdorferstr. 141 - Tel. 0 22 32/130 63 + 4 71 05

AMIGA PROGRAMMER'S HANDBOOK

Das „Amiga Programmer's Handbook“ von Eugene P. Mortimer ist ein weiteres Buch, das sich mit der Programmierung des Amiga beschäftigt. Für Programmier-Laien ist es allerdings weniger geeignet als für erfahrene Profis in der 68000er-Programmierung und in der Programmiersprache C. Gute Englischkenntnisse sollten ebenfalls vorhanden sein, da es in dieser Sprache verfaßt ist. Das Handbuch behandelt unter anderem alle grafik- und animationsbezogenen Funktionen des Amiga ROM Kernel, die in verschiedene Kapitel unterteilt sind. Zunächst erklärt der Autor Allgemeines über das Amiga-System, die Libraries, Strukturen und gibt andere Hinweise, die für die Programmierung nützlich sind. Im weiteren Verlauf folgt die Erklärung der Funktionen, aufgeteilt in die Kapitel 'The Exec Functions', 'The Graphics Display Functions', 'The Graphics Animation Functions', 'The Graphics Text Functions', 'The Layer Functions', 'The Intuition Functions' und 'The Workbench Functions', wobei in alphabetischer Reihenfolge jede einzelne Funktion behandelt und besprochen wird. Am Ende des fast 650 Seiten starken Bandes finden sich schließlich ein Glossary, das sich mit verschiedenen Begriffserklärungen auseinandersetzt, ein Kapitel über die verschiedenen Bildschirm-Modi des Amiga und ein Index-Register. Zu erwähnen ist, daß der Autor auch die neuen Funktionen der Version 1.2 berücksichtigt und besonders gekennzeichnet hat.

Das Buch kann für ernsthafte Amiga-Programmierer empfohlen werden; Anwendern mit geringer Programmiererfahrung ist jedoch vom Kauf

abzuraten, da Eugene P. Mortimore kein Einsteigerbuch verfaßt hat, was wohl auch nicht sein Grundgedanke war. (AK)



Eugene P. Mortimore
 Amiga Programmer's Handbook
 Volume 1
 Sybex Verlag 2. Auflage 1987
 642 Seiten, DM 50,80

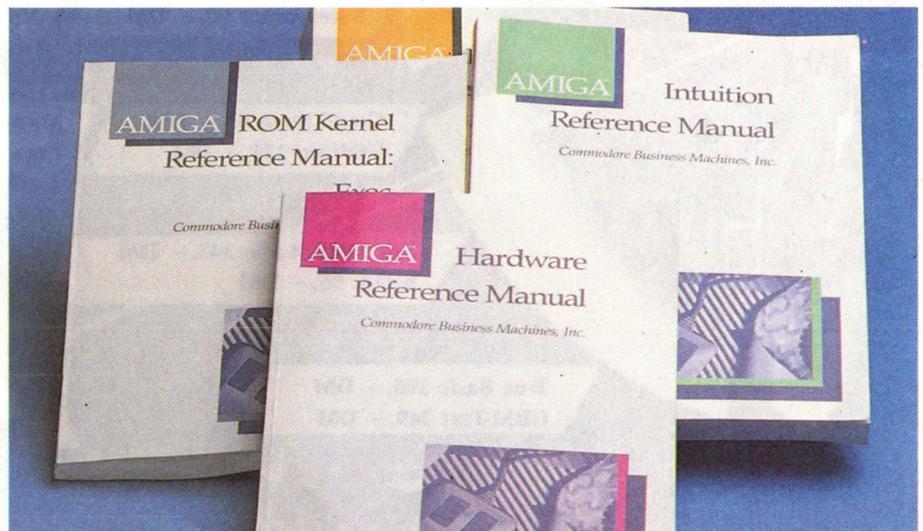
WAWAS

Seien Sie ehrlich: Sie haben sich einen Amiga zugelegt, da Sie von den überragenden Fähigkeiten des Rechners bei Sound und Grafik fasziniert waren. Will man jedoch zur Tat schreiten und die Möglichkeiten dieser Maschine in eigene Programme umsetzen, hilft die mitgelieferte Dokumentation nicht recht weiter. Dieses Manko zu beseitigen, helfen fast viereinhalb Kilo Papier, verteilt auf rund 1800 Seiten und vier Einbände.

Mit über 1000 Seiten ist der Teil über das ROM-Kernel-Reference-Manual 'Libraries and Devices' das dickste und mit einem Preis von 88,- DM auch das teuerste Buch dieser Serie. Es informiert seinen Leser detailliert über die im System-ROM enthaltenen Routinen zur Programmierung von Grafik, Sprites, VSprites, Bobs und deren Animation. Anhand zahlreicher Beispielprogramme kann auch ein wenig versierter Programmierer die Funktionsweise der Libraryroutinen verstehen und Programmteile direkt aus dem Buch in eigene Projekte übernehmen. Der zweite Teil dieses Buches befaßt sich mit den auf dem Amiga verfügbaren Ein- und Ausgabegeräten sowie mit deren Ansteuerung aus einem Programm heraus. Hierbei wird auf die Schnittstelle zwischen Programm und Gerätetreibern eingegangen und an einfachen Beispielen die Wirkung der möglichen Befehle auf die verschiedenartigen Devices gezeigt. Der Programmierer erhält somit ausgetestete Prozeduren, die er sofort in eigenen Applikationen verwenden kann. Teil drei zeigt die Verwendung der auf dem Amiga implementierten Matherroutinen. Dieser Teil enthält außerdem die Beschreibung der verwendeten Register beim Aufruf der einzelnen Funktionen. Dies ermöglicht es dem Assemblerprogrammierer, diese Routinen auf einfache Weise in seinen Programmen zu verwenden. Auch die Auswirkungen auf die Flags des Prozessors sind in diesem Kapitel dokumentiert. Außerdem wird in diesem Teil gezeigt, wie man eigene Programme unter der Workbench bzw. dem CLI zum Laufen bringt.

Die mehr als 500 Seiten Anhang des Buches enthalten eine komplette Auflistung der besprochenen Funktionen. Das zweite Buch befaßt sich ebenfalls mit dem Systemkern des Amiga. Es geht speziell auf den Multitaskingteil des Betriebssystems ein. Der Leser erhält hier wichtige Informationen über den Umgang mit Tasks, Messageports, I/O, Interrupts, Speicherallocation und Libraries des Amiga. Auch in diesem Band sind wieder zahlreiche Beispielprogramme abgebildet, an denen die Funktion der einzelnen Routinen und Aufrufe gezeigt wird. Das Hardware-Reference-Manual enthält Informationen über

geräten (Tastatur, Maus, Joystick usw.). Weiter 100 Seiten enthalten eine Liste aller Intuitionfunktionen und deren Beschreibung. Dies läßt das insgesamt ca. 240 Seiten starke Buch schnell zum Nachschlagewerk für die Programmierung der Benutzeroberfläche werden. Insgesamt gesehen bieten diese vier Bücher eine Fülle von Informationen, die es ermöglichen, die Fähigkeiten der hervorragenden Hardware des Amiga auszuschöpfen. Ein Wermutstropfen ist allerdings die Tatsache, daß alle vier Bände in englischer Sprache verfaßt sind. Für Anwender, die sich selbst intensiver mit der Program-



den direkten Zugriff auf die Hardware. Es beschreibt die Programmierung der einzelnen Kontrollregister auf Assemblerebene und gibt auf etwa 250 Seiten die Beschreibung von Copper, Blitter, der Audiohardware, Sprites und Interfaces. Das Buch schließt mit einer Auflistung aller Systemregister und deren Bitbedeutungen. Das letzte der vier Werke befaßt sich mit der Benutzerschnittstelle des Betriebssystems. Der Programmierer erfährt bei der Lektüre alles Wissenswerte über den Aufbau von Windows, Screens, Gadgets, Menüs, Requesters, Alerts sowie über den Umgang mit den vorhandenen Eingabe-

mierung des Amiga auseinandersetzen wollen, ist diese Serie trotz dieser Tatsache und trotz des hohen Preises eine lohnende Investition.

(GC)

ADDISON-WESLEY Verlag
ROM Kernel Reference Manual:
Libraries and Devices 88,-
ROM Kernel Reference Manual:
Hardware Reference Manual, Exec, Intuition
Reference Manual je 62,50

AMIGAWARE

Midi Interface Mimetics 115,- DM
Mirror Copier 110,- DM

Metacomco Pascal 185,- DM
Midi Interface ECE 140,- DM

Deluxe Paint II 265,- DM
Flight Simulator II 95,- DM

Sindbad 95,- DM
Star Glider 75,- DM

Hollywood Hijinx 80,- DM
Kings Quest I-III 95,- DM

Star-Fleet I 120,- DM
Surgeon 149,- DM

Marauder 185,- DM
Metacomco Assembler 185,- DM

Univited 110,- DM
Vip Professional 320,- DM

Bard's Tale 95,- DM
Bureaucracy 90,- DM

Portal 95,- DM
Prism 135,- DM

Grand Slam 95,- DM
Gridiron 135,- DM

Mouse Pad 14,50 DM
Page Setter 280,- DM

Pro Midi Studio 345,- DM
S.D.I. 95,- DM

True Basic 295,- DM
UBM-Text 249,- DM

und weitere
74 Programme

Porto bei Vorkasse 4,- DM
bei Nachnahme 6,- DM

Auslandsbestellungen nur gegen
Vorkasse + 6,- DM Porto



Bestellservice: 9-18.30 - 0 61 72- 2 47 48
18.30-22 - 0 61 71- 5 38 63

S - SPIEL

A - ANWENDER

SYBEX VERLAG

PROGRAMMIERUNG

DES AMIGA

Dieses Buch hält wirklich alles, was sein Titel verspricht: Es behandelt alle wichtigen Gebiete zur Programmierung des Amiga ausführlich – vom AmigaDOS über Intuition bis zur Erklärung des 'ED'-Editors zur Erstellung von Programmen.

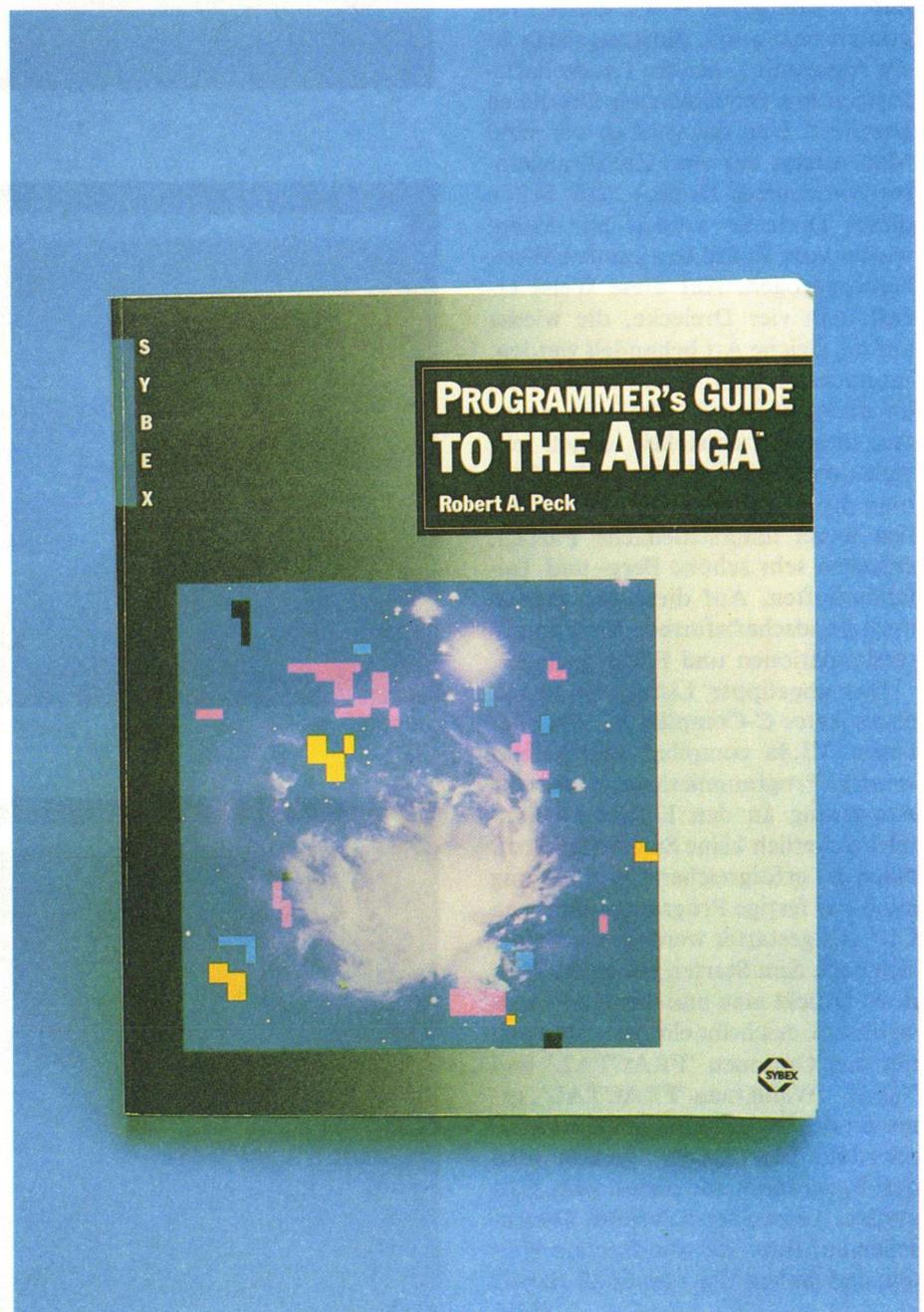
Leider ist deutschsprachige Literatur, die sich intensiv mit der Programmierung des Amiga beschäftigt, hierzulande noch recht dünn gesät. So ist auch dieses Buch in Englisch verfaßt und setzt also Leser voraus, denen die englische Sprache wenigstens vom Hörensagen ein Begriff ist.

Robert A. Peck beginnt sein Werk mit allgemeinen Informationen über den Amiga und erklärt, was man wissen sollte, um den Amiga zu programmieren. Danach behandelt er alle wichtigen Gebiete: Von AmigaDOS über Exec, Grafik, Intuition, Devices, Animation, Sound, Multitasking, ED bis hin zum Amiga-C-Compiler (Lattice C). Manchen Kapiteln sind Programmierbeispiele in C beigelegt, was natürlich Kenntnisse in dieser Sprache voraussetzt.

Insgesamt also ein sehr hilfreiches Buch, das sich für die Programmierung des Amiga durchaus eignet. Wer ein englisches Wörterbuch nicht scheut, könnte sich mit dieser Arbeit anfreunden.

(AK)

Robert A. Peck
Programmer's Guide to the Amiga
Sybex Verlag 1. Auflage 1987
352 Seiten, DM 50,80



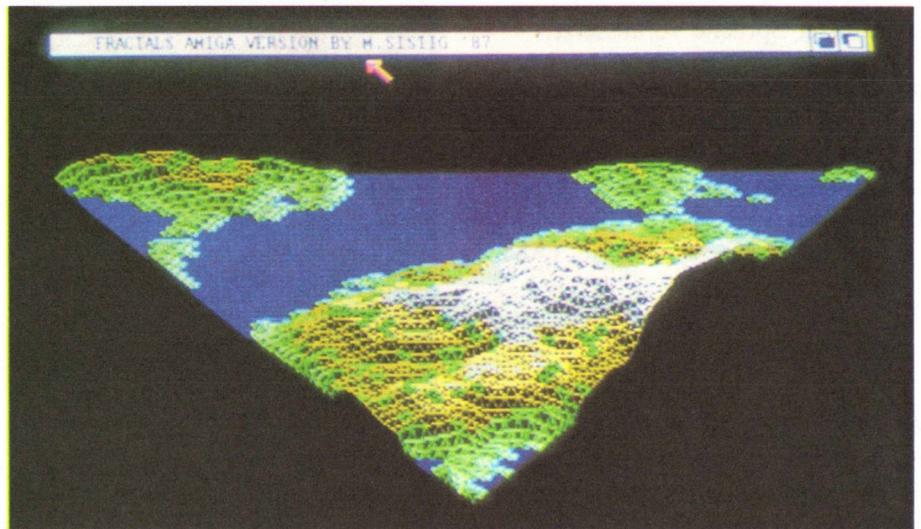
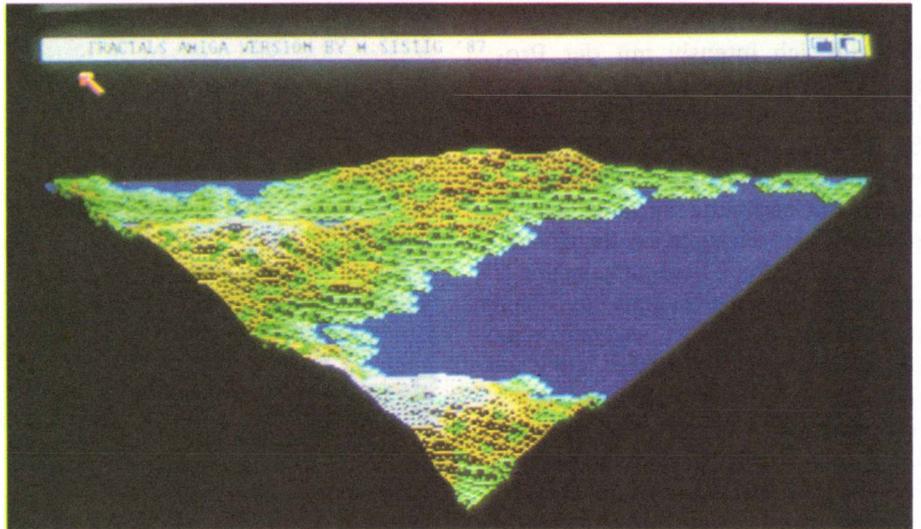
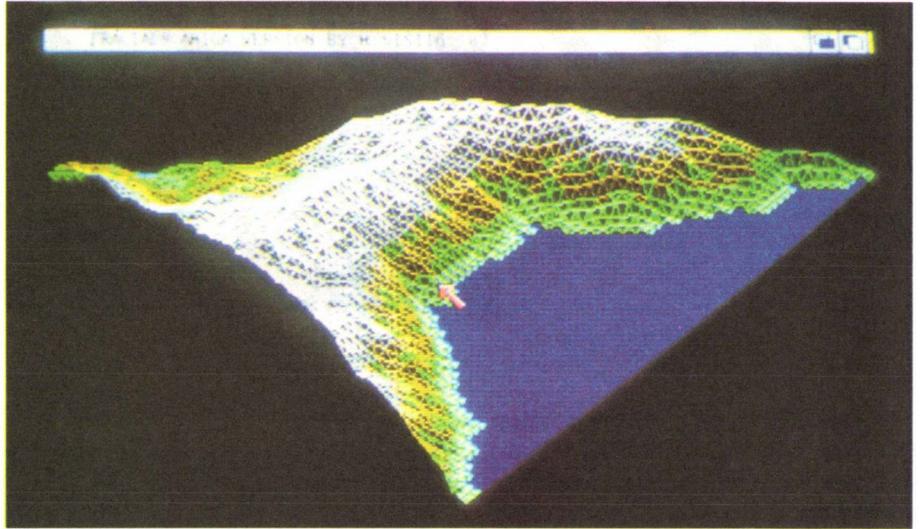
"FRACTALS:"

Fractale Landschaften in 10 Sekunden

Mit dem Programm „Fractals“ lassen sich sogenannte „fractale“ Landschaften erzeugen. Der Begriff 'fractal' läßt sich am besten als 'gebrochen' erklären. Die fractale Geometrie wurde von dem Mathematiker Benoit Mandelbrot entdeckt. Nach ihm wurden auch die „Mandelbrotmengen“ benannt, aus denen die phantastischen Grafiken, die sogenannten 'Apfelmännchen' stammen. Aber nicht nur 'Apfelmännchengrafiken' lassen sich mit fractaler Geometrie berechnen, sondern auch Computerlandschaften – wie dieses Programm zeigen soll. Ausgangsbasis ist ein Algorithmus, der die Landschaftsformen aus gewöhnlichen Dreiecken generiert. Und das geht so vor sich: Man nimmt ein vom Zufallsgenerator berechnetes Dreieck. Die Seiten dieses Dreiecks werden um einen, wieder vom Zufall bestimmten Wert, herausgezogen. Auf diese Weise erhält man vier Dreiecke, die wieder auf die gleiche Art behandelt werden. Im nächsten Schritt sind es dann 16, im übernächsten 32 Dreiecke u.s.w. Auf diese Art entstehen schnell sehr viele unterschiedliche Dreiecke. Läßt man diese nun zeichnen und gibt ihnen dabei unterschiedliche Farben, entstehen sehr schöne Berg- und Talandschaften. Auf diese Art werden auch Landschaftsformen für Computersimulationen und Filme gemacht.

Das abgetippte Listing sollte mit einem Aztec C-Compiler V3.20a oder besser V3.4a compiliert und gelinkt werden. Programmierhasen macht die Anpassung an den Lattice C-Compiler sicherlich keine Schwierigkeiten. Nach der erfolgreichen Compilierung kann das fertige Programm dann von CLI aus gestartet werden. Es meldet sich nach dem Starten mit einem Window. Drückt man nun die rechte Auswahl-taste, erscheint ein Auswahlmönü mit den Optionen 'FRACTAL' und 'QUIT'. Wählt man 'FRACTAL', generiert das Programm zufällige Landschaften. Mit 'QUIT' verläßt man das Programm und kehrt zum CLI zurück. Lassen Sie sich selbst überraschen auf Ihrer Reise in fractale Welten und haben Sie viel Spaß dabei!

(MS)



```

1:  /* FRACTALS
2:     Amiga Version von Michael Sistig
3:     Kuenstlerische Beratung W.Wolfson
4:     5' 87 */
5:
6:     #include "exec/types.h"
7:     #include "intuition/intuition.h"
8:     #include "functions.h"
9:     #include "stdio.h"
10:    #include "math.h"
11:    #include "graphics/gfxmacros.h"
12:    #include "exec/memory.h"
13:    #include "exec/devices.h"
14:    #include "devices/timer.h"
15:    #include "exec/exec.h"
16:    #include "graphics/copper.h"
17:    #include "graphics/gels.h"
18:    #include "graphics/regions.h"
19:    #include "graphics/gfxbase.h"
20:    #include "devices/keymap.h"
21:    #include "hardware/blit.h"
22:
23:    #define UP 0
24:    #define DOWN 1
25:
26:    extern long rnd();
27:
28:    struct IntuitionBase *IntuitionBase;
29:    struct GfxBase *GfxBase;
30:
31:    #define INTUITION_REV OL
32:    #define GRAPHICS_REV OL
33:
34:    struct NewScreen NewScreen = /* Screen Definition */
35:    {
36:        0, /* Muss immer 0 sein */
37:        0, /* TopEdge */
38:        640, /* High Resolution */
39:        200, /* Kein Interlace Modus */
40:        4, /* Depth (max. 16 Farben) */
41:        0,1, /* DetailPen und BlockPen Bestimmung */
42:        HIRES, /* Kein spezieller Anzeige Modus */
43:        CUSTOMSCREEN, /* Der Bildschirm Typ */
44:        NULL, /* Kein spezieller Font */
45:        (UBYTE *)" FRACTALS AMIGA VERSION BY M.SISTIG '87", /* Kein spezieller Font */
46:        NULL, /* Keine Gadgets */
47:        NULL, /* Kein CustomBitMap */
48:        };
49:
50:    struct Screen *Screen;
51:    struct NewWindow NewWindow;
52:    struct Window *Window;
53:
54:    struct RastPort *rp;
55:    struct ViewPort vp;
56:    struct ViewPort *mvp;
57:    long random();
58:    long h[64][64],sw,aw,a2,1,yg,xz,yz;
59:    static long rdiv[]={63,31,15,7,3,1,0};
60:    long punkte[8];
61:    unsigned long ss,hh;
62:
63:    ULONG Class;
64:    USHORT Code;
65:    struct MenuItem MenuItem[2];
66:    struct IntuiText MenuText[2],BodyText,PositiveText,NegativeText;
67:    struct Image MenuImage[2];
68:    struct Menu Menu[1];
69:    struct IntuiMessage *Message;
70:

```

```

71: main()
72: {
73:     int ende;
74:     fenster();
75:     Set_Colors();
76:     Das_Menue();
77:     SetAPen(rp,0L,11L,640L,200L);
78:     RectFill(rp,0L,11L,640L,200L);
79:     ende = FALSE;
80:     CurrentTime(&ss,&hh);
81:     while(!ende) /* Hauptschleife */
82:     {
83:         Wait(1L << Window->UserPort->mp_SigBit);
84:         while (Message = (struct IntuiMessage *) GetMsg (Window->UserPort))
85:         {
86:             Code = Message->Code;
87:             Class = Message->Class;
88:             ReplyMsg (Message);
89:             if (Class== MENUPICK)
90:             {
91:                 if (Code != MENUNULL)
92:                 {
93:                     switch (MENUNUM (Code))
94:                     {
95:                         case 0L:
96:                             SetAPen(rp,0L);
97:                             RectFill(rp,0L,11L,640L,200L);
98:                             Mache_Fractal();
99:                             Zeichne_Fractal();
100:                             break;
101:                         case 1L:
102:                             ende = TRUE;
103:                             break;
104:                         |
105:                         |
106:                         |
107:                         |
108:                         |
109:                         |
110:                         |
111:                         |
112:                         |
113:                         |
114:                         |
115:                         |
116:                         |
117:                         |
118:                         |
119:                         |
120:                         |
121:                         |
122:                         |
123:                         |
124:                         |
125:                         |
126:                         |
127:                         |
128:                         |
    }
    }
    }
    /* Ende Hauptschleife */
    /* Unterroutinen */
    Austieg();
    ClearMenuStrip (Window);
    CloseWindow (Window);
    CloseScreen (Screen);
    CloseLibrary (IntuitionBase);
}

```

```

129:
130:   CloseLibrary (GfxBase);
131:   }
132:
133:   zufall(stufe)
134:   long stufe;
135:
136:   {
137:   }
138:   return ((rnd((long)hh)&rdiv[stufe-1L])-(rdiv[stufe]));
139:   }
140:
141:   Set_Colors()
142:   {
143:   }
144:   rp = Window->RPort;
145:   vp = Screen->ViewPort;
146:  .mvp = &vp;
147:
148:   SetRGB4 (mvp, 0L, 0L, 0L, 0L); /* Black */
149:   SetRGB4 (mvp, 1L, 13L, 13L, 13L); /* White */
150:   SetRGB4 (mvp, 2L, 0L, 0L, 7L); /* Blue */
151:   SetRGB4 (mvp, 3L, 6L, 5L, 0L); /* Brown */
152:   SetRGB4 (mvp, 4L, 0L, 7L, 0L); /* Green */
153:   SetRGB4 (mvp, 5L, 5L, 5L, 5L); /* DGrey */
154:   SetRGB4 (mvp, 6L, 3L, 15L, 5L); /* LGreen */
155:   SetRGB4 (mvp, 7L, 8L, 8L, 8L); /* Grey */
156:   SetRGB4 (mvp, 8L, 15L, 15L, 15L); /* GWhite */
157:   SetRGB4 (mvp, 9L, 2L, 11L, 3L); /* MLGreen */
158:   SetRGB4 (mvp, 10L, 9L, 8L, 3L); /* MLBrown */
159:   SetRGB4 (mvp, 11L, 1L, 9L, 1L); /* 3Green */
160:   }
161:
162:   Mache_Fractal () /* Fractals berechnen */
163:   {
164:   }
165:   h[0][0]=0L;
166:   h[0][64]=0L;
167:   h[64][0]=0L;
168:
169:   sw=32; aw=64; a2=32;
170:   for (l=1;l<=6;l++)
171:   {
172:   yz=0L; yg=0L;
173:   do {
174:   if (yg++%2==0)
175:   { xz=sw;
176:   do {
177:   h[xz][yz]=(h[xz-a2][yz]+h[xz+a2][yz])/2+zufall(1);
178:   xz+=aw;
179:   }
180:   while (xz<64-yz);
181:   }
182:   else
183:   { xz=0L;
184:   do {
185:   h[xz][yz]=(h[xz][yz-a2]+h[xz][yz+a2])/2+zufall(1);
186:   xz+=a2;
187:   h[xz][yz]=(h[xz-a2][yz+a2]+h[xz+a2][yz-a2])/2+zufall(1);
188:   xz+=a2;
189:   }
190:   while (xz<=64-yz);
191:   }
192:   yg+=a2;
193:   }
194:   while (yz<64);
195:   aw/=2;
196:   a2/=2;
197:   sw/=2;
198:   }
199: }

```

LISTING

```
200:     Zeichne_Fractal() /* berechnete Fractals zeichnen */
201:
202: {
203:     long x,y,farbe;
204:
205:     for (y=0;y<=63;y++) for (x=0;x<=63-y;x++)
206:     {
207:         farbe=2L;
208:         punkte[0]=x*10L+y*5L;
209:         punkte[1]=y*2+60;
210:         if (h[x][y]>0) {
211:             punkte[1]=punkte[1]-h[x][y];
212:             farbe=6L;
213:         }
214:         punkte[2]=punkte[0]+10L;
215:         punkte[3]=y*2+60;
216:         if (h[x+1L][y]>0) {
217:             punkte[3]=punkte[3]-h[x+1L][y];
218:             farbe=9L;
219:         }
220:         if (h[x+1L][y]==2) {
221:             farbe=9L;
222:         }
223:         if (h[x+1L][y]==3) {
224:             farbe=11L;
225:         }
226:         punkte[4]=punkte[0]+5L;
227:         punkte[5]=y*2+62;
228:         if (h[x][y+1L]>4) {
229:             punkte[5]=punkte[5]-h[x][y+1L];
230:             farbe=3L;
231:         }
232:         if (h[x][y+1L]==4) {
233:             farbe=4L;
234:         }
235:     }
236:     if (h[x][y+1L]==7) {
237:         farbe=11L;
238:     }
239:     if (h[x][y+1L]>10) {
240:         farbe=10L;
241:     }
242:     if (h[x][y+1L]==12) {
243:         farbe=4L;
244:     }
245:     if (h[x][y+1L]>14) {
246:         farbe=7L;
247:     }
248:     if (h[x][y+1L]>18) {
249:         punkte[5]=punkte[5]-(int)((h[x][y+1L])/64);
250:         farbe=1L;
251:     }
252:     if (h[x][y+1L]>23) {
253:         punkte[5]=punkte[5]-(int)((h[x][y+1L])/16);
254:         farbe=8L;
255:     }
256:     if (h[x][y+1L]==20) {
257:         farbe=7L;
258:     }
259:     if (h[x][y+1L]==25) {
260:         farbe=7L;
261:     }
262:     if (h[x][y+1L]==29) {
263:         farbe=7L;
264:     }
265:     punkte[6]=punkte[0];
266:     punkte[7]=punkte[1];
267:
268:
```

```

269:         SetAPen(rp, farbe);
270:         Move(rp, punkte[0], punkte[1]);
271:         Draw(rp, punkte[2], punkte[3]);
272:         Draw(rp, punkte[4], punkte[5]);
273:         Draw(rp, punkte[6], punkte[7]);
274:     }
275: }
276:
277:
278: fenster() /* Window initialisieren und oeffen*/
279: {
280:     IntuitionBase = (struct IntuitionBase *)
281:     OpenLibrary("intuition.library", INTUITION_REV);
282:     if (IntuitionBase == NULL) exit(FALSE);
283:     GfxBase = (struct GfxBase *)OpenLibrary("graphics.library", GRAPHICS_REV);
284:     if (GfxBase == NULL) exit(FALSE);
285:     if ((Screen = (struct Screen *)
286:     OpenScreen(&NewScreen)) == NULL)
287:     exit(FALSE);
288:
289:     /* Windowdefinitionen */
290:
291:     NewWindow.LeftEdge = 0;
292:     NewWindow.TopEdge = 0;
293:     NewWindow.Width = 640;
294:     NewWindow.Height = 200;
295:     NewWindow.DetailPen = 0;
296:     NewWindow.BlockPen = 1;
297:     NewWindow.Title = NULL;
298:     NewWindow.Flags = SMART_REFRESH|ACTIVATE|BORDERLESS
299:     |BACKDROP|NOCAREREFRESH|REPORTMOUSE;
300:     NewWindow.IDCMPFlags = MOUSEBUTTONS|MOUSEMOVE|MENUPICK;
301:     NewWindow.Type = CUSTOMSCREEN;
302:     NewWindow.FirstGadget = NULL;
303:     NewWindow.CheckMark = NULL;
304:     NewWindow.Screen = Screen;
305:     NewWindow.BitMap = NULL;
306:     NewWindow.MinWidth = 100;
307:     NewWindow.MinHeight = 25;
308:     NewWindow.MaxWidth = 640;
309:     NewWindow.MaxHeight = 200;
310:     if ((Window = (struct Window *)OpenWindow(&NewWindow)) == NULL) exit(FALSE);
311: }
312:
313: Das_Menu() /* Menueleiste erstellen und initialisieren */
314: {
315:     long z;
316:
317:     for (z=0; z<2; z++){
318:         MenuItem[z].LeftEdge=0L; MenuItem[z].TopEdge=z*10L;
319:         MenuItem[z].Width=64L; MenuItem[z].Height=10L;
320:         MenuItem[z].Flags=ITEMTEXT|ITEMENABLED|HIGHCOMP;
321:         MenuItem[z].MutualExclude=NULL;
322:         MenuItem[z].ItemFill=(APTR) &MenuText[z];
323:         MenuItem[z].SelectFill=NULL;
324:         MenuItem[z].Command=NULL;
325:         MenuItem[z].SubItem=NULL;
326:         MenuItem[z].NextSelect=NULL;
327:
328:         MenuText[z].FrontPen=0L; MenuText[z].BackPen=1L;
329:         MenuText[z].TopEdge=1L;
330:         MenuText[z].DrawMode=JAM2;
331:         MenuText[z].ITextFont=NULL;
332:         MenuText[z].NextText=NULL;
333:     }
334: }

```

```

341:
342: MenuItem[0].NextItem=&MenuItem[1];
343: MenuItem[1].NextItem=NULL;
344:
345: MenuItem[0].LeftEdge=6L; MenuItem[0].IText=(UBYTE *)"FRACTAL";
346: MenuItem[1].LeftEdge=6L; MenuItem[1].IText=(UBYTE *)"QUIT";
347:
348: Menu[0].NextMenu=NULL;
349: Menu[0].LeftEdge=10L; Menu[0].TopEdge=0L;
350: Menu[0].Width=63L; Menu[0].Height=10L;
351: Menu[0].Flags=MENUEENABLED;
352: Menu[0].MenuName="PROJECT";
353: Menu[0].FirstItem=&MenuItem[0];
354:
355: SetMenuStrip (Window,&Menu[0]);
356: }
357:
358: /* Assembleroutine zum erstellen der Funktion rnd (Zufallszahlengenerator) */

359:
360: #asm
361:
362:
363:         blanks off
364:         section code
365:         xdef     _rnd
366:
367: _rnd     lea     rndseed,a0      ;Get address of seed
368:         move.l  4(sp),d1        ;Get range argument
369:         tst.l   d1
370:         ble.s  setseed         ;Go reset seed
371:
372:
373:         move.l  (a0),d0         ;Get seed
374:         ADD.L  D0,D0
375:         BHI.S  over
376:         EORI.L # $1D872B41,D0
377: over
378:         move.l  d0,(a0)         ;Save new seed
379:         andi.l # $ffff,d0      ;Coerce into word
380:         andi.l # $ffff,d1
381:         divu   d1,d0           ;Divide by range
382:         swap   d0              ;and get remainder (modulus)
383:         ext.l  d0
384:         rts
385:
386: setseed  neg.l   d1             ;Probably don't need this
387:         move.l  d1,(a0)
388:         rts
389:
390:         section data
391: rndseed  dc.l   0
392:         section code
393:
394:
395: #endasm /* Ende der Assembleroutine rnd */
396:

```

Einkaufsführer

Hier finden Sie Ihren
Commodore/Amiga Fachhändler

Anzeigenschluß Heft 9/87: 24. Juli 1987

1000 Berlin

 **RUNOW**
Büroelektronik
Keithstraße 26 · 1000 Berlin 30
☎ 26 111 26

2000 Hamburg

Bit Computer Shop
Osterstraße 173 · 2000 Hamburg 20
Telefon: 040/49 4400
createam
Computer Hard & Software
Bramfelder Chaussee 300 · 2000 Hamburg 71
Telefon Sa. Nr. 040/641 50 91

2072 Bargteheide

MSD Computer GmbH
Rathausstr. 9 · 2072 Bargteheide
☎ 045 32 / 2 13 15

2160 Stade

 **BERGAU**
Büromaschinen · EDV-Systeme
Neue Straße 5, 2160 Stade
Telefon: (04141) 23 64 + 23 84

2390 Flensburg

 **electronic
computer
laden ohg**
Norderstr. 94-96 · D-2390 Flensburg
(04 61) 2 81 81 & 2 81 93

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

3043 Schneverdingen

**Bürosystem
Gerhard Hinrichs**
Verdener Str. 15a
3043 Schneverdingen
Tel. 0 51 93 / 40 31-32

3250 Hameln

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 

4000 Düsseldorf

**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
☎ 061 51 - 5 60 57

4100 Duisburg

Rennen Datentechnik
Helmut Rennen GmbH & Co. KG
Mercatorstr. 80
4100 Duisburg
☎ 02 03 - 2 49 26

4650 Gelsenkirchen-Horst


Hard- und Software, Literatur
Bauteile, Service, Versand
Groß- und Einzelhandel
Poststr. 15 · 4650 Gelsenkirchen-Horst
Tel. 02 09 / 5 25 72

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

5400 Koblenz

Micro-Electronic GmbH

Schützenstr. 9 - 11
5400 Koblenz
Telefon 02 61-182 00

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

Hardware
Software
Service



Büro-Computer + Organisations GmbH

Commodore
TOSHIBA
ATARI OKI/DATA

Ihr Partner,
wenn es um
Computer geht

6000 FRANKFURT/M. 1, Oeder Weg 7-9, ☎ 0 69/55 04 56/57

6200 Wiesbaden

Henneveld KG

Schossbergstr. 21
6200 Wiesbaden
Tel. 0 61 21 - 27 70

6250 Limburg

wir
bürosysteme vertriebs gmbh

diezer strasse 10
6250 limburg
tel. 0 64 31 - 2 00 30

6300 Gießen

Ihre Tür zur Zukunft:

KARSTADT
computer-center
hardware · software · problemlösungen

Gießen, Seltersweg 64, Telefon (06 41) 70 04 - 318

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

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



Computer-Center
am Hauptbahnhof GmbH

L 14, 16-17
6800 Mannheim 1
Tel. (06 21) 2 09 83/84

7000 Stuttgart

FRITZ COMPUTER GMBH

SCHULZE-DELITZSCH-STR. 22
7000 STUTTGART
TEL. (07 11) 78 00-230/238/239

Hier könnte

Ihre Anzeige

erscheinen.

Anruf genügt: Heim - Verlag
Tel. 0 61 51 - 5 60 57

7000 Stuttgart

»If AMIGA, go to Schreiber«
Stuttgart's starker Computer-Laden.

**SCHREIBER
ELEKTRONIK**

Alte Poststraße 2, 7000 Stuttgart-City
1. Stock, Telefon (0711) 22 70 99

7157 Murrhardt

Bürocomputer oHG
Förster & Scharf

Hauptstr. 12
7157 Murrhardt 3
Tel. 0 71 92-70 61

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

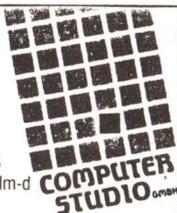
hettler-data
service gmbh

Lenzburger Straße 4
7890 Waldshut-Tiengen
Telefon 0 77 51 / 30 94

7900 Ulm

EDV-Systeme
Software-
erstellung
Schulung

Systemhaus:
Frauenstr. 28
7900 Ulm/Donau
Tel. 07 31/2 80 76
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
INGOLSTADTER 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

8300 Landshut-Ergolding

Anton Sneganas

Landshuter Str. 64
8300 Landshut-Ergolding
Tel. 0 89 71-7 10 11

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

Hier könnte

Ihre Anzeige

erscheinen.

Anruf genügt:

Heim-Verlag

☎ 0 61 51-5 60 57

8400 Regensburg

**Zimmermann
elektroland**

8400 Regensburg
Dr.-Gessler-Str. 8
☎ 09 41/9 50 85

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

8580 Bayreuth

**COMPUTERSYSTEME
R. SCHLÄGER GMBH**

Parkstr. 3 · 8580 Bayreuth
☎ 09 21-6 91 63

8700 Würzburg

SCHULL
BÜROTEAM
Hardware · Software
Service · Schulung
computer center
am Domlnikanerplatz
Ruf (09 31) 5 04 88

ST-Computer Einkaufsführer

Werbewirksam, aktuell
und preiswert.

Sprechen Sie mit uns:
Heim-Verlag
0 61 51/5 60 57

FÜNF ABOS

für 36 Kreuzchen!

Um diese Zeitschrift auch nach den Wünschen der Leser gestalten zu können, möchten wir gerne wissen, wo Ihre Interessen liegen. Um Sie für Ihre Mühe zu belohnen, verlosen wir unter allen Einsendern unseres Leserfragebogens fünf Abonnements von Kickstart, der AMIGA-Zeitschrift, außerdem gibt es verschiedene aktuelle Spiele und Public-Domain-Disketten.

Selbstverständlich werden wir alle Ihre Angaben streng vertraulich behandeln und nicht länger im Computer speichern, als es für die Auswertung erforderlich ist.

Einsendeschluß für die Verlosung ist der 15. August '87.

1. Ich habe einen Amiga 500 1000 2000
2. Ich habe 1/2M 1M 2M Speicher
3. Ich habe eine Zweitfloppy ja nein
4. Ich habe einen Drucker ja nein
5. Welchen? _____
6. Ich habe einen Sidecar ja nein
7. Ich habe ein Bridgeboard ja nein
8. Welches? XT AT 68020
9. Ich habe eine Festplatte ja nein
10. Ich habe ein Grafiktablett ja nein
11. Ich habe ein Midi-Interface ja nein
12. Ich benutze auch andere Erweiterungen
(Welche? _____)

Ich plane mir in der nächsten Zeit folgende Geräte und Erweiterungen zu kaufen:

- 13. AMIGA 500
 AMIGA 2000
- 14. Bridgeboard XT
 Bridgeboard AT
- 15. 500 K Speicher
 1 M Speicher oder mehr
- 16. Zweitlaufwerk
 Festplatte

Vergeben Sie bei den folgenden Fragen bitte Noten von 1 (sehr gut) bis 4 (gar nicht):

17. Meine Kenntnisse im Bereich Computer sind (natürlich ohne zu mogeln) _____

Ich beherrsche folgende Programmiersprachen:

- 18. PASCAL
- 19. BASIC
- 20. C
- 21. Assembler
- 22. _____

(bitte die Noten in die Kästchen eintragen!)

Am meisten interessieren mich:

- 23. Softwaretests
- 24. Hardwaretests
- 25. Programmierkurse
- 26. Anwendungsbeispiele
- 27. Listings
- 28. Aktuelle Meldungen
- 29. Spiele
- 30. MS-DOS auf Amiga

- 31. Tips & Tricks
- 32. Grafik
- 33. Musik
- 34. Hardwarebasteleien
- 35. Buchkritiken
- 36. _____
- 37. Ich benutze den Computer beruflich
 Büro
 Wissenschaft/Technik
- 38. Ich schreibe eigene Programme
 ja nein
- 39. beruflich privat
- 40. Ich lese sogar die Konkurrenz; welche _____

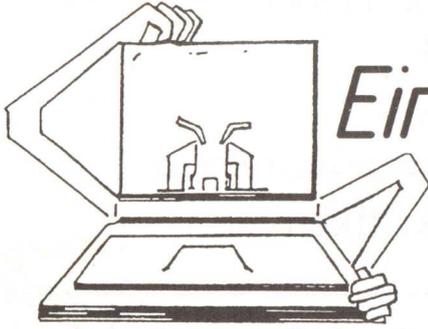
Für Ihre Antwort verwenden Sie bitte die beigelegte Postkarte.

NAME _____
STRASSE _____
VORNAME _____
PLZ/ORT _____
ALTER _____
Das finde ich an KICKSTART toll/nicht so toll:

So, das war's schon. Jetzt aber schnell zum Briefkasten! Und vielen Dank fürs Mitmachen.

Ihre Kickstart-Redaktion.

Künstliche Intelligenz



Ein kontroverses Thema mit eindeutigen Antworten

Einleitung

Die Phantasie des Menschen wurde schon immer von der Idee ange-regt, eines Tages könnten Maschinen geistige Tätigkeiten vollbringen. Je nach Zeitgeist und Neigung des Au-

tors wurden solche Vorstellungen als Fortschritt bejubelt oder als Gotteslästerung verdammt. Ein kontroverses Thema also, die künstliche Intelligenz. Aber was ist eigentlich künst-

liche Intelligenz? Oder ist es vielleicht richtiger, erst einmal nach der natürlichen Intelligenz zu fragen? Und welche Themen und Forschungsarbeiten verstecken sich hinter dem Kürzel KI?

Intelligenz

Intelligenz ist laut Brockhaus Klugheit bzw. die Fähigkeit zur geistigen Anpassung. Beides sind Begriffe, die im Zusammenhang mit dem Computer und künstlicher Intelligenz zunächst erneut Stoff zur kontroversen Auseinandersetzung liefern, denn die Fähigkeit zur geistigen Anpassung setzt die Existenz des Geistes in der Maschine voraus. Die erste sinnvolle und bis heute gültige Definition künstlicher Intelligenz stammt folglich aus dem pragmatischen England, wo nach dem Zweiten Weltkrieg der junge Mathematiker Alan Turing eine auf den ersten Blick äußerst banal erscheinende Definition künstlicher Intelligenz gab:

Man setze einen Menschen an ein Terminal und lasse ihn daran mit einem Partner kommunizieren. Dieser Partner könnte ein entsprechend programmierter Computer, aber auch ein Mensch sein. Läßt sich aus den Reaktionen des Partners nicht eindeutig herleiten, daß er ein programmierter Computer ist, dann besitzt das Programm (falls der Partner tatsächlich ein Computer war) künstliche Intelligenz.

Tatsächlich ist der genannte Test äußerst anspruchsvoll, weil der Mensch in der Wahl seiner Themen nicht eingeschränkt ist und damit das Gespräch in jede erdenkliche Richtung lenken kann. Damit muß der Computer, soll er dem Anspruch an Weltwissen des Menschen genügen, mit Informationen über all die All-

tagsobjekte gefüttert werden, für die der normale Mensch Jahrzehnte kognitiven Lernens und Jahrtausende genetischer Erfahrung aufbringen mußte.

Die Masse macht's

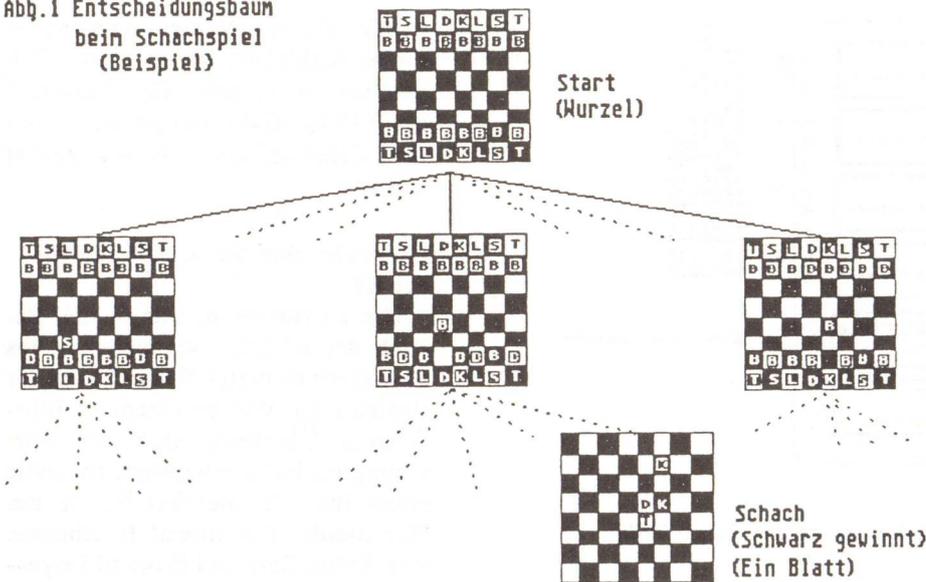
Damit ist ein Grund für den zögerlichen Durchbruch von künstlicher Intelligenz bereits geortet: Die enormen Datenmengen müssen schnell und effektiv verwaltet werden. Das klassische Beispiel gefährlicher Unterschätzung ist das Schachspiel. Sie erinnern sich an die Sage von dem Erfinder des Schachbretts, der an seinen König als Lohn für die Erfindung des königlichen Spiels folgende Forderung stellte: 1 Weizenkorn auf das erste Feld, zwei Weizenkörner auf das zweite, vier Weizenkörner auf das dritte.... Der Mann wurde wegen seiner Unverschämtheit gehängt! Allerdings wurde er zunächst für seine Bescheidenheit belächelt, bis die Brutalität exponentieller Wachstumsgesetze den König das harte Urteil fällen ließ. Und wie so oft in der Geschichte wiederholte sich das Schauspiel in der KI-Forschung. Gewiß ist das Schachspiel auf Grund seiner eindeutigen Regelungen gut als Experimentierfeld für KI geeignet. Die ersten Ansätze resultieren aus der frühen Faszination für die scheinbar ins gigantische gestiegene Rechenfähigkeit des Computers. Man wollte einen Sieg erzwingen, indem man alle aus einem Zug des Gegners resultierenden Schachzüge durchrechnete,

um den sicher zum Schach führenden Zug dann auszuführen. Diese als rohe Gewalt (engl.: brute force) bekannte Methode scheiterte an dem gleichen Phänomen wie die Weizenkornforderung, der kombinatorischen Explosion. Abb. 1 zeigt einen hypothetischen Spielverlauf beim Schachspiel als sogenannten Entscheidungsbaum. Jeder Zug, ob Schwarz oder Weiß, bedeutet einen Schritt von der Wurzel des Baumes (sie befindet sich wie bei allen Bäumen in der Informatik oben, am Ausgangspunkt aller Entscheidungen) in Richtung auf die Blätter des Baumes (unten im Bild. Sie bedeuten den Endzustand des Spieles). C. Shannon hat die Zahl der Entscheidungen in einem solchen Entscheidungsbaum auf 10¹²⁰ geschätzt. Engagierte Schachspieler brauchen also keine Angst vor dem allwissenden Computer zu haben. Nach der Methode der rohen Gewalt hätte auch der schnellste Computer seit der Schöpfung des Weltalls das Eröffnungsspiel noch nicht überwunden.

Heuristik, das Zauberwort

Nun weiß natürlich jeder Schachspieler, daß es durchaus möglich ist, ein Schachspiel zu gewinnen. Aber auch der genialste Schachspieler ist ebensowenig wie der Computer in der Lage, alle Züge eines Spiels im voraus zu überlegen. Es muß also eine Möglichkeit geben, das Entscheidungsdickicht zu durchbrechen. Diese Möglichkeit heißt Heuristik. Man versteht darunter Erfahrungsregeln,

Abb.1 Entscheidungsbaum beim Schachspiel (Beispiel)



re (fertige Software gibt es natürlich noch nicht) zu tätigen. Natürlich wird auch hier der computerspezifische Preisverfall einsetzen, sodaß (vielleicht noch dieses Jahr?) auf Preise unterhalb der 5000-DM-Grenze gehofft werden darf. Natürlich kann man solche Computer nicht mehr in BASIC programmieren. Speziell die Transputer werden in OCCAM programmiert, einer Sprache, die von vornherein von der Parallelverarbeitung Gebrauch macht, ohne allerdings speziell in den Bereich der KI eingeordnet zu werden (dazu ist diese Sprache wohl noch zu neu). Überhaupt hat man sich in der KI sehr schnell anderer Sprachen bedient, als sie die Mehrheit der Computeranwender kennt.

die nicht herleitbar sind, aber im allgemeinen Erfolg haben. Ein Beispiel ist die Regel „Springer am Rand“. Durch solche Regeln läßt sich der Entscheidungsbaum drastisch reduzieren. Dennoch bleibt die Größenordnung der Datenmengen sehr groß im Vergleich zu konventionellen Problemen der Informatik. Mit der Suche nach Heuristiken alleine ist es deshalb bei weitem nicht getan. Ein weiteres Konzept, das in jüngerer Zeit in greifbare Nähe gerückt scheint, ist die parallele Datenverarbeitung.

Parallele Datenverarbeitung

Die Grundidee ist einfach: wenn eine Aufgabe zu groß für eine einzelne Person ist, teilt man die Aufgabe in Teilaufgaben und läßt jede Teilaufgabe von einer einzelnen Person lösen. Solches Teamwork kann die benötigte Lösungszeit bekanntlich beträchtlich reduzieren. Umgesetzt in den Bereich der Informatik beutet dieses Konzept eine Hardware, bestehend aus mehreren CPU's, die autark arbeiten, über Verbindungsleitungen aber miteinander kommunizieren können. Abb.2 zeigt das am weitesten vorausgeplante Projekt einer solchen Maschine. Es handelt sich um den Computer der 5. Generation, wie er anlässlich einer Konferenz über die 5. Computergeneration in Japan im Jahre 1982 konzipiert wurde. Erstaunlicherweise sind aber in Europa die ersten lauffähigen Mehrprozessorsysteme vorgestellt

worden. Die neuesten Mitglieder dieser Transputer genannten Mehrprozessorfamilie sind sogar zu floating-point-Operationen fähig. Abb. 3 zeigt das Transputerkonzept der engl. Firma INMOS. Hier erfolgt die Kommunikation der verschiedenen Transputer untereinander über die sogenannten LINKS. Als Besitzer eines Amiga 2000 bzw. eines Sidecar kann man sogar von dem Fortschritt der Transputertechnik profitieren, weil für den IBM-PC und Kompatible (d.h. Slotbesitzer) Einsteckkarten existieren, die einen PC in ein Transputersystem verwandeln. Vorausgesetzt, man ist in der Lage die lächerliche Investition von rund zwölf-tausend Mark incl. Entwicklungssoftwa-

Software für KI-Probleme

Für den Menschen sind Maschinen Hilfsmittel, mit denen er sich lästiger Probleme entledigen will, um sich angenehmeren Beschäftigungen hinzugeben. Eine der lästigsten Beschäftigungen ist für die meisten Menschen auch heute noch der Umgang mit Zahlen. Daher sind die ersten Computer als Fortsetzung einer Entwicklung zu verstehen, an deren Anfang die mechanischen Rechenmaschinen von Leibniz und Schickard standen. Um die ersten Rechner (man beachte die Bezeichnung!) zu programmieren, war es üblich, Schalter zu setzen, die eine Binärzahl codieren und damit einen Befehl für die CPU darstellten (1.

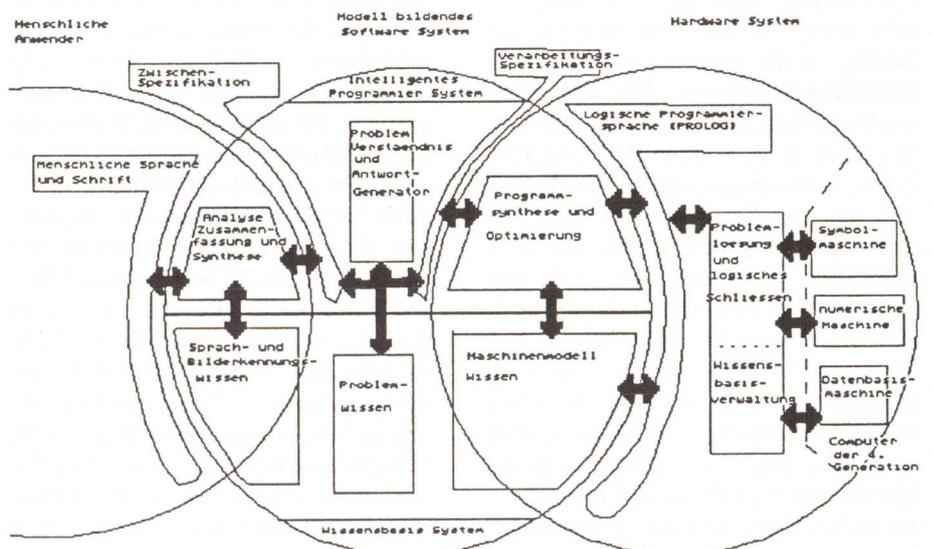


Abb. 2 FGCS Modell nach Moto-Oka.

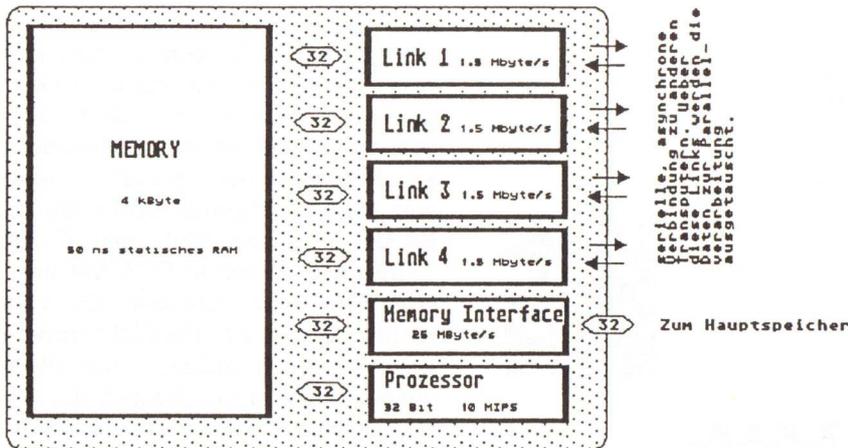


Abb.3: Blockschaltbild des IMS T424 Transputer der Firma INMOS.

Sprachgeneration). In der Folge sind Hilfen entstanden, die die Effektivität der Programmierung erheblich gesteigert haben. Nach der reinen Maschinensprache (s.o.) wurde die Assemblersprache erfunden, die die Befehle der CPU durch Kürzel (Mnemonics) symbolisch darstellt und anschließend in die Maschinensprache übersetzt (2. Generation). Der Durchbruch in der Anwendung von Computern begann aber erst mit der Entwicklung der Hochsprachen (3. Generation), die auch einem außerhalb der Informatik Tätigen die Entwicklung eigener Anwendersoftware ermöglichte. Eine der ältesten Sprachen dieser 3. Generation ist FORTRAN (FORMular TRANslator), dessen Kürzel schon die Zielgruppe verrät: Wissenschaftler, die für ihre Forschung immer leistungsfähigere Rechenknechte benötigten. Das im Mikrocomputerbereich fast unsterbliche BASIC ist übrigens ein vereinfachter FORTRAN-Ableger. Mit der Entwicklung natürlich rekursionsfähiger Sprachen (z.B. C oder PASCAL) ist die 4. Softwaregeneration ins Leben gerufen, der Fortschritt in der Effizienz des Programmierens war hier allerdings nicht mehr so stark wie noch bei dem Schritt von der 1. zur 2. Softwaregeneration. Die meisten Leser werden Erfahrung mit dem Programmieren in einer dieser Sprachen haben und bestätigen können, daß diese Sprachen sich am leichtesten im Bereich der Naturwissenschaften bzw. einfacher Dateiverwaltung anwenden lassen. Im Alltag des Menschen spielen diese Bereiche natürlich eine un-

tergeordnete Rolle. Nach der oben gegebenen Definition künstlicher Intelligenz von Alan Turing sind es aber gerade die alltäglichen Dinge, die ein Computer beherrschen muß, wenn er menschenähnlich reagieren soll. Und die wesentliche Eigenschaft menschlichen Denkens ist das Denken in Symbolen. Folglich bedarf es zur Programmierung künstlicher Intelligenz Sprachen, die der Symbolmanipulation fähig sind. Diese Erkenntnis reifte den Informatikern schon früh, denn bereits im Jahre 1958(!) entwickelte John McCarthy die Sprache LISP (LIST Processing), eine Symbolmanipulationssprache, bei der anfangs (entsprechend der Aufgabenstellung) Gleitkommaarithmetik nicht vorgesehen war. LISP wurde im Laufe der Zeit oft verändert, mit COMMON LISP steht nun aber ein Standard fest. Diesem Standard ist auch das im Public-Domain-Service erhältliche XLISP von David Michael Betz aus den USA angepaßt worden. Es steht dem an Problemen der KI interessierten AMIGA-Benutzern zum kostenlosen Einstieg in die Welt der KI zur Verfügung. In jüngster Zeit wird die Popularität von LISP etwas durch die Sprache PROLOG eingeschränkt, die als einzige Sprache im oben erwähnten FGCS-Projekt der Japaner namentlich Beachtung findet. Dies ist hauptsächlich auf die angestrebte Fähigkeit zur Parallelverarbeitung zurückzuführen, denn PROLOG ist auf Grund seiner Sprachstruktur und internen Arbeitsweise für die Abarbeitung paralleler Prozesse geeignet. PROLOG

wurde 1972 von der Groupe d'Intelligence Artificielle unter Prof. Colmerauer entwickelt. Der Standard der PROLOG-Versionen ist jedoch von Clocksin und Mellish gesetzt worden.

Übersicht über die Arbeitsgebiete der KI

Um zu verstehen, was in den Bereich der KI fällt, schauen wir uns einmal die bisherige Entwicklung der Projekte an. Wer an einem ausführlicheren Überblick über das Forschungsgebiet KI interessiert ist, sollte einen Blick in die drei Bände des Handbook of Artificial Intelligence von Avron Barr und Edward Feigenbaum werfen. Natürlich handelt es sich bei allen Problemen um Dinge, die uns Menschen leicht erscheinen, eben um Alltägliches. Spötter definieren KI deshalb auch als das, was dem Menschen leicht und dem Computer schwer fällt.

Sehen Parallel zur Entwicklung der Videotechnik ist der Fortschritt im Bereich visueller Datenverarbeitung ungeheuer rasch gewachsen. Besonders der AMIGA mit seinen hervorragenden grafischen Fähigkeiten und seiner großen CPU-Leistungsfähigkeit wird sicher als preisgünstiges Entwicklungssystem in diesem Bereich eingesetzt werden, zumal die Ankoppelung an Industrieprojekte über die PC-Slots leicht möglich ist. Abb. 4 zeigt das Blockdiagramm für ein maschinelles Bilderkennungssystem nach Berthold K.P. Horn.

Hören und Sprechen

Während im visuellen Bereich anfangs große technische Schwierigkeiten einen schnellen Fortschritt im Bereich des maschinellen Sehens unwahrscheinlich erscheinen ließen, erwartete man im Bereich des maschinellen Hörens relativ raschen Fortschritt. Es zeigte sich aber, daß die ebenfalls in diesen Bereich fallende Spracherkennung bzw. Sprachverarbeitung ein erheblich schwierigeres Problem ist als angenommen. Im Jahr 1964 sagte der Gordon Helmer Bericht die ersten Programme zur maschinellen Übersetzung bereits für 1978 voraus. Wie jeder weiß, wird dagegen auch heute noch keine Schreibmaschine mit akustischer Eingabe angeboten.

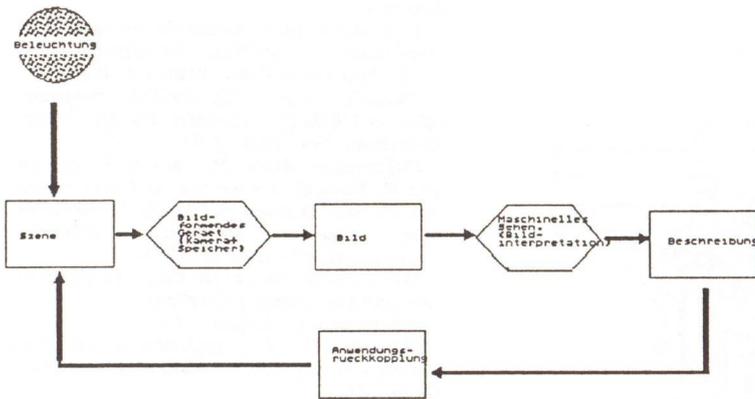


Abb.4: Anforderungen an ein System zum maschinellen Sehen, nach B.K.P. Horn.

Geschweige denn eine Maschine zur akustischen Simultanübersetzung, die die akustische Information analysiert, die Semantik der Quellsprache erkennt, in die Syntax der Zielsprache übersetzt und akzentfrei in der Zielsprache akustisch ausgibt. Die größte Schwierigkeit ist hier die Mehrdeutigkeit menschlicher Sprache. Die erste grundlegende Arbeit auf dem Gebiet der Spracherkennung (allerdings nicht der akustischen) stammt von Terry Winograd. Abb. 5 zeigt die verschiedenen Programmteile, die bei der Analyse einfacher geschriebener Sätze in einer eng begrenzten Welt aus Bauklötzen miteinander wechselwirken um einen als umgangssprachlichen Satz formulierten Befehl in eine Aktion zu überführen. Abb. 6 zeigt die Bauklotzwelt zu Beginn des Dialoges. Der Bildschirmdialog liest sich dann etwa so (entnommen aus D. R. Hofstadters Buch Gödel, Escher, Bach):

Nimm einen großen roten Klotz. OK. Ergreife die Pyramide. ICH VERSTEHE NICHT, WELCHE PYRAMIDE SIE MEINEN. Suche einen Klotz, der größer ist als der, den du hältst und lege ihn in die Schachtel.

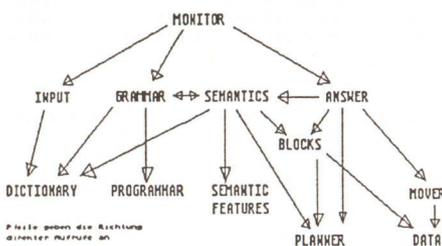


Abb. 5 Organisation von T. Winograds SHRDLU

ICH NEHME AN, DASS SIE UNTER „IHN“ DEN KLOTZ VERSTEHEN, DER HÖHER IST ALS DER, DEN ICH HALTE. OK.

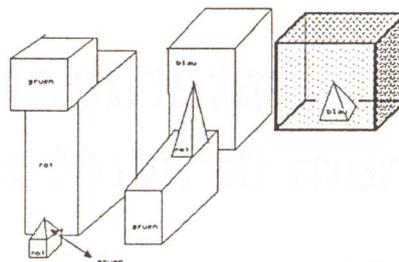


Abb.6: Blockwelt nach T. Winograd

Fühlen

Mit Fühlen ist natürlich Sensorik gemeint und nicht das Gefühl als Ausdruck von Leidenschaft. Der Tastsinn ist schließlich einer der wichtigsten menschlichen Sinne, entsprechend schwierig gestaltet sich seine technische Nachbildung. Einfache Sensoren arbeiten mit einer Gummimatte, unter der eine Matrix von Schaltern verborgen ist. Bessere Sensoren verwenden Piezo-Elemente. Allerdings haben die den Nachteil, daß bei längerer Belastung eine Signaler-müdung eintritt, d.h. bei konstanter Kraft läßt die Stärke des elektrischen Signals nach. Eine der neueren Entwicklungen ist in Abb. 7 gezeigt. Verblüffend ist die äußerliche Ähnlichkeit mit dem Aufbau der menschlichen Haut.

Robotik

Eine Integration aller menschlichen Sinne ist erforderlich, wenn man von

einer Maschine menschliches Orientierungsvermögen und Fingerfertigkeit verlangt. Dies ist natürlich das Ziel der Roboterforschung. Zwar können Roboter schon enorme Kräfte zielgenau einsetzen, was aber noch fehlt, ist die Koordination mehrerer Arme zur Erledigung feinmechanischer Arbeiten. Ebenfalls noch verbesserungswürdig ist die autonome Navigation, d.h. die Fähigkeit des Roboters, sich in einer fremden Umgebung zielstrebig zu bewegen, ohne an irgendwelche Hindernisse anzustoßen. Die heutigen Transportroboter laufen meistens durch unsichtbare Induktionsschleifen geführt durch die Werkshallen und sind kaum in der Lage, auf Störungen zu reagieren.

Nutzen und Moral

Ganz zweifellos bringt die Entwicklung intelligenter Computer enormen Nutzen. Die Produktivität von Betrieben wird ins Unermessliche gesteigert, was umgekehrt sinkende Preise zur Folge hat. Das wiederum hat einen Abbau im Bereich traditioneller Arbeitsplätze zur Folge. Die Spekulation auf neue Arbeitsplätze durch neue Technologien ist durch die Geschichte der industriellen Entwicklung sicher berechtigt. Immerhin hat uns die bisherige Industrialisierung und damit verbundene Produktivitätssteigerung einen ganz beachtlichen Lebensstandard gebracht. Allerdings setzen neue Arbeitsplätze für neue Technologien auch die Fähigkeit zur Produktion eben solcher neuer Produkte voraus. Die in Deutschland inzwischen übliche Strategie, in Fernost produzierte Waren mit einem deutschen Etikett zu bekleben, bringt weder Arbeitsplätze noch (zumindest langfristig) Überlebenschancen für die entsprechenden Industrien. Unabhängig von eventuellen moralischen Bedenken (der Mensch schafft sich sein Ebenbild!) ist die Beschäftigung mit der KI ein Baustein bei der Sicherung des bisherigen Lebensstandards in der Zukunft. Und soll man den Computer alles machen lassen, was er in naher oder ferner Zukunft kann? Nun, ich finde, man sollte alles Machbare probieren, aber immer den Stromschalter in Reichweite...

(Dr. K. Sarnow)

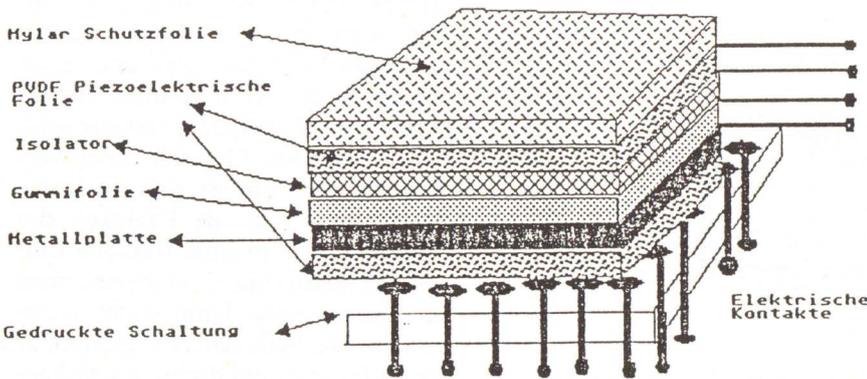


Abb.7: Hautaehnlicher Sensor nach Dario und De Rossi.

Literatur

Barr, Avron und Edward Feigenbaum. *The Handbook of Artificial Intelligence* (Vol. 1-3). Heuristech Press. Stanford, 1981.

Clocksin, W.F., C.S. Mellish. *Programming in PROLOG*. Springer Verlag, Berlin-Heidelberg-New York, 1981.

Colmerauer, Alain, H. Kanoui, R. Pasero und P. Roussel. *Un Systeme de Communication Homme-Machine en Franais. Internal Report. Groupe d'Intelligence Artificielle. Universit Aix-Marseille II*. 1973.

Dario, Paolo, Danilo De Rossi. *Tactile Sensors and the Gripping Challenge*. IEEE Spektrum, August 1985.

Hamann, C. M. *Einführung in das Programmieren in LISP*. Walter de Gruyter. Berlin 1985.

Hofstadter, Douglas Robert. *Gödel, Escher, Bach: Ein endlos geflochtenes Band*. Stuttgart: Klett-Cotta, 1985.

Horn, Berthold Klaus Paul. *Robot Vision*. Mc Graw Hill, 1986. Moto-Oka, T. *Fifth Generation Computer System*. Elsevier. New York 1982.

Winograd, Terry. *Understanding Natural Language*. New York 1972.



Soyka Datentechnik Bochum 02 34/41 19 13



Original **AMIGA**

Made in Germany



- F₁ 3,5"-Einzelfloppy, anschluf. 369,- DM
- F₂ 3,5"-Doppelfloppy, anschluf. 669,- DM
- F₃ 5,25"-Einzelfloppy, anschluf. 489,- DM
- NEC 1036 A + Interface + Kabel + Stecker + Anleitung zum Selbstbau einer 3,5" **Amiga-Floppy 289,- DM**
- 3,5"-Gehäuse, Kunststoff, beige mit Befestigungsmaterial 34,90 DM
- NEC 1036A/NEC 1035LP, 1 MB, 3,5" 259,- DM
- Amigo-Sound Mono 129,- DM · Stereo 199,- DM
- Amigo-Bootselector 39,90 DM
- Amiga 500 - 1 MB Erweiterung incl. Uhr 299,- DM
- Amiga 2000 - 1,5 MB Erweiterung 198,- DM

SOFTWARE

Spiellesoftware

Anwendersoftware

Alien Fires	84,-	Acquisition	599,-
Arena	79,-	Aegis SONIX V2.0	169,-
Barbarian*	79,-	Aztec C Dev. V3.4a	599,-
Bard's Tale	99,-	Aztec C Com. V3.4a	899,-
Chessmaster 2000	89,-	CLI Mate V1.2	69,-
Cruncher Factory	29,-	Deluxe Music Con. Set	199,-
Defender of the Crown	89,-	Deluxe Paint IIB	249,-
Demolition	29,-	Deluxe Video V1.2	249,-
Faery Tale	99,-	Dynamic Cad	999,-
Fire Power*	49,-	Dynamic Word*	379,-
Flightsimulator II	109,-	Galileo	209,-
Flip Flop	39,-	Grabbit	69,-
Galactic Invasion*	49,-	Instant Music	99,-
Grand Prix*	89,-	Lattice C V3.1	349,-
Gunship*	89,-	Lisp	419,-
Karate King	39,-	Macro Assembler	199,-
King of Chikago*	109,-	Marauder II	109,-
Land of Legends*	99,-	Metacomco Pascal	199,-
Phalanx	29,-	Modula II Standart	189,-
Return to Atlantis*	99,-	Modula II Developers	299,-
S.D.I	109,-	Pagesetter europ.Ver.	329,-
Shanghai	79,-	Planetarium*	99,-
Silent Service	89,-	Printmaster Plus	109,-
Sinbad	95,-	Scribble	199,-
Starglider	79,-	K-Seka Assembler	139,-
Strip Poker	79,-	Shell (Cli deluxe)	139,-
Surgeon	119,-	Superbase (deutsch)	229,-
Terrorpods*	79,-	Toolkit	99,-
Turbo*	49,-	UBM Text V2.2	249,-
Uninvited	99,-	Ucsd Pascal	169,-
Quiwi	59,-	Vip Professional	449,-
Wintergames	59,-	Vizawrite (deutsch)	498,-
Worldgames	59,-		

* = in Kürze lieferbar

Amiga-Bücher

Amiga Reference Manuals :	Public Domain Software
Hardware	62.50
Intuition	62.50
Exec	62.50
Libraries and Devices	88,-
	10 Disks 89,-
	30 Disks 249,-
	incl.Disketten

Komplette Softwareliste mit ca.300 Prg. anfordern !

Harald Soyka · Hattinger Straße 685 · 4630 Bochum 5

▶ LOGISTIX

Die vierte Dimension, die Zeit, kalkulierbar zu machen verspricht, ist das neue Softwarepaket LOGISTIX für den Amiga.

Diese neue Software ist nicht einfach ein weiteres Terminplanungsprogramm. Sie beinhaltet nämlich insgesamt folgende Komponenten:

Tabellenkalkulation:

Berechnungen für mathematische Modelle und Analysen

Terminplanung:

Zuordnung von Produktionsfaktoren zu Aufgaben unter Berücksichtigung von Zeitperioden

Datenbank:

Management für die Speicherung und den Abruf der Datenbankinformationen

Grafik:

Darstellungen zur Information und Trendanalyse

Weil alle Funktionen gemeinsam auf einem elektronischen Arbeitsblatt verknüpft werden, sollten nahezu alle Anforderungen an jeden dieser Bereiche erfüllt werden können. Da es in jedem dieser Bereiche einzeln schon gute und auch für den professionellen Anwender brauchbare Programme für den Amiga gibt (VIP, Superbase), muß ein neues Programm schon sehr hohen Anforderungen genügen, um sich hier einen Platz erobern zu können.

Um es vorweg zu sagen: LOGISTIX erfüllt diese hohen Erwartungen. Es ist ein Programm, bei dem jeder Teil für sich schon professionellen Ansprüchen genügt; in der Verbindung aller Komponenten ist dieses Programm momentan fast einzigartig auf dem Markt. LOGISTIX gibt es für MS-DOS-Rechner, für den Amiga und für den Atari. Noch

läuft das Programm ausschließlich mit Tastaturbedienung. Gegen Ende des Jahres soll eine Version herauskommen, die auch mit der Maus bedient werden kann. Es wäre aber unsinnig, auf diese Version zu warten: Zum einen erhalten die Besitzer der bisherigen Version sehr günstig die neue, außerdem erfüllt das Programm auch jetzt schon alle Anforderungen.

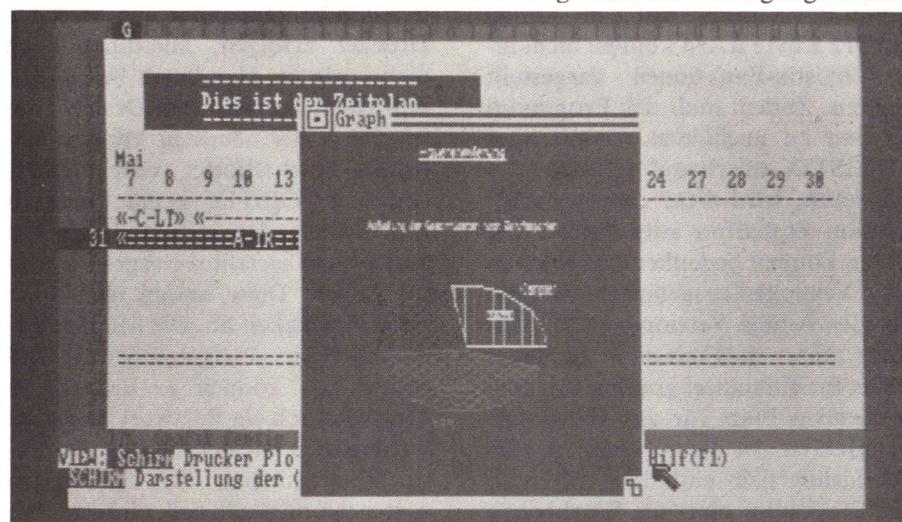
Schon hier möchte ich darauf hinweisen, daß ein Test in einer Computerzeitschrift einem solchen Programm kaum gerecht werden kann. Die Möglichkeiten, die es dem Anwender bietet, sind in kurzer Zeit kaum auszuloten. Deshalb ist dieser Bericht eher eine Aufzählung der vielfältigen Möglichkeiten von LOGISTIX.

Zum Lieferumfang:

LOGISTIX wird mit einer sehr umfangreichen deutschsprachigen Dokumentation geliefert. Ein Ordner im A5-Format mit rund 300 Seiten enthält eine ausführliche Beschrei-

bung aller Möglichkeiten und Befehle. Dabei wird nach folgendem Schema vorgegangen: – Name des Befehls/Unterbefehls – Was bewirkt der Befehl? – Wie wird der Befehl eingesetzt? – Beispiele Dieser Ordner scheint universell für alle Programmversionen zu sein, was der Übersichtlichkeit jedoch nicht schadet. Zusätzlich zum Ordner erhält man noch zwei kleinere Handbücher: Eine Einleitung, die in kurzen Beispielen den mächtigen Umfang von LOGISTIX verdeutlicht, und einen Band mit Zusatzbefehlen, die offensichtlich erst ab der Version 1.1 gültig sind. Zur Zeit ist die Version 1.15 aktuell.

Zusätzlich erhält man eine kartonierte Kurzübersicht mit einer anschaulichen Zusammenfassung der wesentlichen Befehle und Funktionen, gedacht für Anwender, die mit dem Handbuch bereits vertraut sind. In diesem Zusammenhang sollten bereits die sehr ausführlichen „Help“-Funktionen erwähnt werden. Über 90 Kbyte Hilfstexte stehen jederzeit im Programm zur Verfügung. Wenn

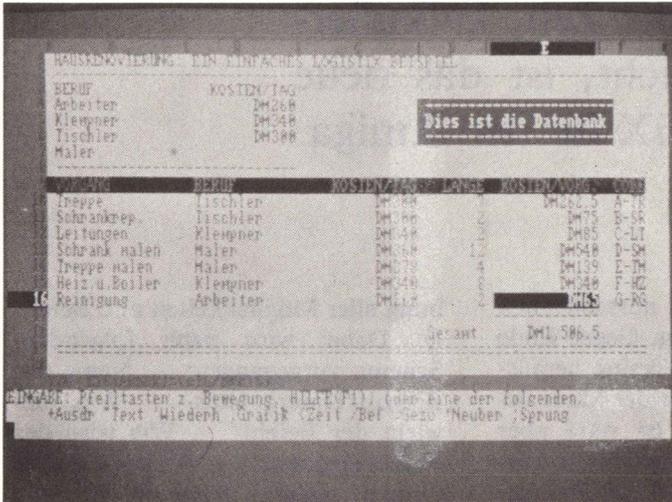


SOFTWARETEST

man bei einem Befehl Hilfe wünscht, kann man über die Funktionstaste F1 jederzeit die zu diesem Befehl passenden ausführlichen Hilfstexte abrufen.

starten kann, dennoch aber den deutschen Zeichensatz erhält. Dadurch stehen ca. 40 KB mehr Speicher für Anwendungen zur Verfügung.

die Tastatur eingegeben werden. Das Programm überträgt diese Tastaturbewegungen dann in die Macro-Sprache! Zusätzlich zu den Macros gibt es



Kopieren Sie von Ihrer Workbench die Dateien system/setmap in das C-Directory Ihrer Logistix-Diskette

devs/keymaps/dl in das devs/keymaps-Directory Ihrer Logistix-Diskette

Erstellen Sie folgende startup-sequence auf der Logistix-Diskette:

```
stack 13000
setmap d
Lgx
```

Bitte arbeiten Sie nur an einer **Kopie** Ihrer Original Diskette!

Zum Lieferumfang gehören außerdem:

- Programmdiskette mit der „Mini-Version“ für den Amiga mit 512K Speicher,
- Programmdiskette mit zwei „Normalversionen“, je nach dem zur Verfügung stehenden Speicherplatz,
- ein Aufkleber mit der Belegung der Funktionstasten,
- ein „Dongle“, ein Kopierschutzstecker, der in den Joystickport 2 gesteckt werden muß, damit das Programm lauffähig ist. Von der Diskette selbst können also Sicherungskopien nach Bedarf erstellt werden.

Mit der „Mini-Version“ hat es folgende Bewandnis: Auf dem Amiga mit 512 Kbyte RAM können nicht alle Logistix-Funktionen dargestellt werden. Zudem muß das Programm ab und zu nachladen. Damit aber LOGISTIX überhaupt genutzt werden kann, wird diese „abgespeckte“ Version kostenfrei mitgeliefert. Die Daten können bedenkenlos zwischen allen Versionen ausgetauscht werden. Bei der Amiga-Version mit 512 KB sollte man zusätzlich noch zu einem weiteren Hilfsmittel greifen, um ausreichenden Platz für das Datenblatt zu erhalten. In dem Kasten ist eine Befehlsübersicht enthalten, wie man das Programm ohne die Workbench

Das Arbeitsblatt:

Das Format des Arbeitsblattes ist 2048 Reihen mal 1024 Spalten. Auf Wunsch kann man mit verschiedenen Formatierungsoptionen unterschiedliche Farben für Ränder, Spalten, Reihen, Cursor, geschützte Bereiche, Befehle oder für den Hintergrund wählen. Wer schon auf anderen Rechnern mit Lotus 123 oder Multiplan gearbeitet hat, kann sich ohne Mühe in die Arbeitsweise von Logistix einarbeiten. Die Vorgehensweise ist gleich. Nach Eingabe des Schrägstriches steht ein umfangreiches Menü zur Verfügung. Für die Übernahme von Daten folgender Programme stehen Optionen zur Verfügung: LOTUS (VIP), Super Calc, dBase, Text, KGD (Komma-getrennte-Dateien), DIF. Die Datenausgabe kann über alle gängigen Drucker erfolgen, allerdings noch nicht, wie im Handbuch bereits beschrieben, im seitlichen Druck. Dies ist erst in der nächsten Version enthalten. Auch Plotter werden unterstützt. LOGISTIX kennt ebenfalls Macros, die den selbständigen Ablauf einmal erstellter Programme ermöglichen. Diese laufen mit hoher Geschwindigkeit ab. Die Möglichkeiten der Macro-Programmierung sind enorm. So können z. B. erstellte Macros durch ein Passwort geschützt werden oder im Lernmodus alle Tastaturbewegungen, die das Macro später durchführen soll, einfach über

die Möglichkeit, bestimmte Abläufe über AUTO-Befehle zu steuern. AUTO-Befehle sind in ihrer Länge auf eine Programmzeile begrenzt, Macros jedoch nur durch den Speicher.

Die Ausführung von AUTO-Befehlen ist an eine bestimmte Taste gebunden. Hier ein Beispiel: Die EDIT-Funktion ist normalerweise nur durch folgende Kombination zu erreichen: Shift und Schrägstrich, dann E. Insgesamt also drei Tasten. Durch folgenden AUTO-Befehl kann man die Funktionstaste F5 mit dem EDIT-Befehl belegen:

```
a)Home: /A Auto
E Edit
F6 Funktionstaste F6
Home Auto-Name
>ANF Macro-Befehl
```

Ab sofort ist die Funktionstaste F5 als EDIT-Taste zu gebrauchen. Diese AUTO-Belegung wird dann mit dem Datenblatt abgespeichert. In diesem Zusammenhang noch zwei Tips: Die Tastatur des Amiga kennt keine „Home-“ und keine „End-“Tasten. Diese werden aber im Programm manchmal gebraucht. Durch den oben beschriebenen Auto-Befehl können diese Tasten auf F6 und F7 gelegt werden. Dazu sind folgende Eingaben notwendig:

```
a)Home: /A Auto
E Edit
F6 Funktionstaste F6
Home Auto-Name
```

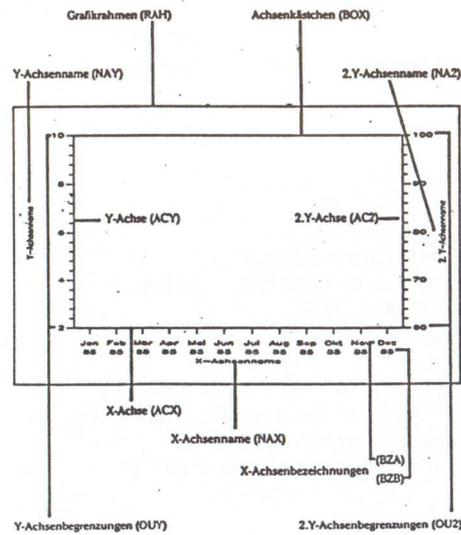
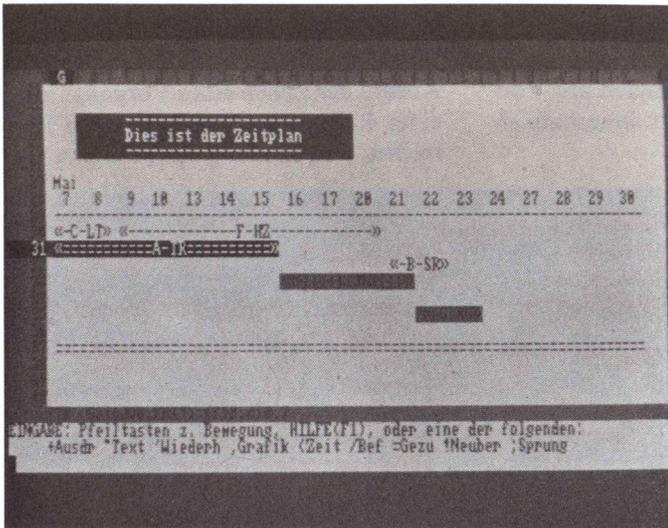


Bild 3:
Bestandteile
einer Grafik.

- > ANF Macro-Befehl
- b) End: /A Auto
- E Edit
- F7 Funktionstaste F7
- End Auto-Name
- > END Macro-Befehl

Eine Übersicht der AUTO- und MACRO-Befehle ist weiter unten zu finden.

Daten von anderen Arbeitsblättern können zur einmaligen oder dauernden Verwendung mit dem aktuellen Datenblatt verbunden werden. Sie können also z.B. die Verkaufszahlen jedes Monats auf einem eigenen Datenblatt speichern und sie dann auf einem neuen Datenblatt einlesen und bearbeiten lassen. Änderungen werden dabei auch später noch berücksichtigt.

Einige Funktionstasten sind mit festen Funktionen belegt:

- F1 - Hilfe
- F2 - Dateien
- F3 - Neuberechnung
- F4 - Ansicht
- F9 - Seite links
- F10 - Seite rechts

Zu dem riesigen Befehlsumfang von LOGISTIX zunächst einige Be-

merkungen. Der Autor dieses Artikels mußte sich folgende Gedanken machen: Zeigt er nur einige wenige Möglichkeiten des Programms, dann wird er den Möglichkeiten eines solchen Produktes wohl kaum gerecht. Er ist auch demjenigen keine Hilfe, der diesen Bericht liest, um sich für ein bestimmtes Programm zu entscheiden. Gerade diese Programme sind es doch, die, mit Ausnahme der Textsysteme, ihre größte Verbreitung

in den Büros haben. Und vielleicht ist es gerade eine oder mehrere Möglichkeiten, die der Leser bei einem Programm vermißt und die er bei diesem Programm findet. Derjenige, der ein solches Programm nicht anwenden kann, wird diesen Text ohnehin nicht lesen oder höchstens überfliegen. Ich habe mich deshalb dazu entschlossen, zwar nicht alle, aber doch die wichtigsten Befehle und Möglichkeiten aufzulisten:

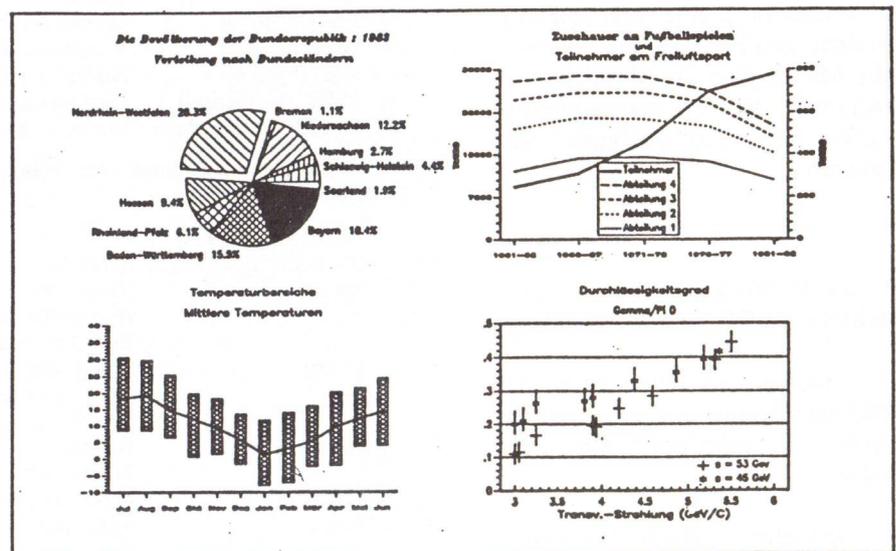


Bild 4:

Zusätzlich ist es möglich, mehrere Grafiken auf einem Schirm auszugeben.

SOFTWARETEST

Mathematische Funktionen:

Sämtliche Winkelfunktionen mit den zusätzlichen Möglichkeiten Arc, Hyperbolisch, Grad, Rad.

PI	3,1415926536
ABS(Wert)	Absoluter Wert
GANZ(Wert)	Wert ganzer Zahlen
RUNDEN(Wert,n)	Gerundeter Wert mit n Stellen
E	2,718281828459
EXP(Wert)	e(2,718281828459) zur Potenz des Wertes
LN(Wert)	Natürlicher Logarithmus (Basis e)
LOG(Wert)	Dekadischer Logarithmus (Basis 10)
WURZEL(Wert)	Quadratwurzel des Wertes
ZUFALLZ	Zufallszahlen zwischen 0 und 1

LAENGE(Bezeichnung,Reihe/Bereich):

Anzahl der besetzten Zellen pro Vorgang, auf einer Reihe o. innerhalb eines Bereiches.

PUFFER(Bezeichnung,Reihe/Bereich):

Anzahl der Zellen eines Vorgangs auf einer Reihe oder innerhalb eines Bereiches. (Anwendung nur bei der Ana-

Logische Funktionen:

WAHR	Anzeige: 1
FALSCH	Anzeige: 0
UND(Werte)	Anzeige: 1, wenn alle Werte ungleich 0, sonst
ODER(Werte)	1, wenn mind. ein Wert 0 ist, sonst 0
NICHT(Wert)	1, wenn Wert 0 ist, sonst 0
WENN(Kond.Wert,richtiger Wert,falscher Wert)	Richtiger Wert, wenn Kond.Wert ungleich 0, falscher Wert, wenn kond. Wert gleich 0
FEHLER	Anzeige FEHLER
ISTFEHL(Wert)	1, wenn Wert FEHLER, sonst 0
NV	Anzeige N/V (nicht vorhanden)
ISTNV(Wert)	1, wenn Wert N/V, sonst 0

Das Zeitmanagement

Logistix erlaubt mit einfachen Methoden ein sinnvoll nutzbares Zeitmanagement. Dabei wird jeder Spalte eine Zeiteinheit zugeteilt und jeder Reihe eine Aufgabe. Die Zeiteinheiten können von Minuten bis zu Jahren lauten. Feiertage werden dabei automatisch berücksichtigt. Das Programm bietet nun eine Vielzahl von Funktionen, diese Zellen zu bearbeiten. Das geht sogar so weit, daß Logistix den kritischen Weg herausucht; also den Pfad, an dem alle anderen Aufgaben zeitlich hängen und wo eine Verzögerung bei einer Aufgabe zwangsläufig eine Verzögerung der endgültigen Fertigstellung bedeutet. Es besteht also die Möglichkeit, im Rahmen des Zeitmanagements mit „Was ist wenn“-Aufgaben zu kalkulieren.

Die einzelnen Zeitfunktionen:

(Bis zu 10 Vorgangsnamen, oder Bezeichnung JEDE für jede Funktion)

START(Bezeichnung,Reihe/Bereich):
Erstmaliger Beginn eines Vorgangs auf einer Reihe oder innerhalb eines Bereiches.

ENDE(Bezeichnung,Reihe/Bereich):
Letztes Erscheinen eines Vorgangs auf einer Reihe oder innerhalb eines Bereiches.

NACH(Bezeichnung,Reihe/Bereich):

Abläufe, die nach Beendigung eines oder mehrerer Vorgänge erfolgen, auf einer Reihe oder innerhalb eines Bereiches.

lyse des „kritischen Weges“ als „Pufferzeit“).

JNAME(Zelle):

Name des Vorgangs (JOB) in Zellen.

Kalenderfunktionen:

ZSM(Spalte)	Zeit der Spalte(Stunden und Min.)
TAG(Spalte)	Tag(Zahlenwert) der Spalte
MONAT(Spalte)	Monat(alphabetisch) der Spalte
JAHR(Spalte)	Jahr der Spalte
TUM(Spalte)	Tag und Monat der Spalte
TUMUJ(Spalte)	Tag, Monat und Jahr der Spalte
MUJ(Spalte)	Monat und Jahr der Spalte
TDW(Spalte)	Tag d. Woche (alphabetisch) der Spalte
MON(Spalte)	Monat der Spalte
WNR(Spalte)	Wochennummer der Spalte
NRT(Spalte)	Nummer des Tages der Spalte
DATUM(Datum)	Datumsanzeige der Spalte
DAZT(Datum,Zeit)	Datums- und Zeitanzeige der Spalte

Dazu kommen noch im Rahmen der „Schrägstrich“-Funktionen:

ZEIT-BEFEHLE:

(Name,Länge,(Beginn(Reihe))

Name	Name des Vorgangs: Text in Anführungszeichen (") oder Angabe einer Zell-Referenz mit Textinhalt
Länge	Länge des Vorgangs: Anzahl der Spalten (jeder Ausdruck, dessen Ergebnis ein numerischer Wert ist)
Start	Beginn des Vorgangs: Nummer der Spalte (jeder Ausdruck, dessen Ergebnis ein numerischer Wert ist)
Reihe	Reihe des Vorgangs: Angabe der Reihe (jeder Ausdruck, dessen Ergebnis ein numerischer Wert ist)

sowie die Funktionen „Calendar“ und „Kritisch“ (siehe unten).

Statistische Funktionen:

SUM(Werte)	Summe numerischer Zahlen
Anzahl(Werte)	Anzahl der Zahlenwerte
MIT(Werte)	Mittelwert der Werte
MAX(Werte)	Maximalwerte der Werte
MIN(Werte)	Minimalwerte der Werte

Die Datenbank von Logistix bietet folgende Funktionen:

(Ein Feld kann mit einem Feld „Offset“ oder einem Namen gekennzeichnet sein.)

DSUM(Eingabe,Kriterium,Feld)
Summe der numerischen Werte von Feldern aus ausgewählten Datensätzen.

DANZAHL(Eingabe,Kriterium,Feld)
Anzahl der numerischen Werte von Feldern aus ausgewählten Datensätzen

DMIT(Eingabe,Kriterium,Feld)
Mittelwert der numerischen Werte von Feldern aus ausgewählten Datensätzen

DMAX(Eingabe,Kriterium,Feld)
Maximum der numerischen Werte von Feldern aus ausgewählten Datensätzen

DMIN(Eingabe,Kriterium,Feld)
Minimum der numerischen Werte von Feldern aus ausgewählten Datensätzen

Finanzfunktionen:

KRZ(Schätzung,Werte)	Kapitalrückflußrate
KWM(Z-Satz,Werte)	Wertberechnung nach Kapitalwertmethode
ZW(Zahlung,Z-Satz,Zeit)	Zukünftiger Kapital-Wert
GW(Zahlung,Z-Satz,Zeit)	Gegenwärtiger Kapital-Wert (Barwert)
HYZ(Kapitel,Z-Satz,Zeit)	Hypothekenzahlung

Spezielle Funktionen:

SPA	Nummer der Spalte
NSPA(Spalte)	Nummer der Spalte (alphabetisch)
REI	Nummer der Reihe
ERMITT(n,Werte)	Ergibt den numerischen Wert
SUCHEN(Wert,Bereich)	Ermittelt den Wert, der im Bereich dem vorliegenden am nächsten liegt
JTMJ(Wert)	Julianischer Tag, Monat, Jahr
JDATUM(Datum)	Julianische Zahl des Datums
SDAT	Systemdatum
SZEIT	Systemzeit

Schrägstrich-Befehle:

/Auto Anzeige	Auflistung aller Befehle, die eine Zuordnung zu einer Spezialtaste bekommen haben
Editieren	Editieren des automatischen Befehls, der einer Spezialtaste zugeordnet wurde
Löschen	Löschen des automatischen Befehls
/Blank	Nimmt den Inhalt aus einer oder mehreren Zeilen heraus
/Calendar Editieren	Editieren der Kalendermonate und Arbeitstage
Verknüpfen	Zuordnung des Kalenders zum Terminplan
Optionen	Wahl der Zeiteinheiten und der Ar-

Löschen	Entfernung des Kalenders vom Terminplan und Wiederherstellung der Standardvorgabe
/Delete	Löscht eine oder mehrere Spalten oder Reihen
/Edit	Ändert den Inhalt der aktuellen Zelle
/Format	Bestimmung des Anzeigenformats für Zellen, Zellblöcke, Reihen, Spalten oder Arbeitsblatt
/Global	Bestimmung verschiedener globaler Optionen: Relative/absolute Zell-Referenzen Umstellung relativer zu absoluten Zell-Referenzen Manuelle/automatische Neuberechnung Reihe für Reihe/Spalte für Spalte Neuberechnung Ränder Ein/Aus Geschützte Zellen verfügbar/nicht verfügbar Grafik-Befehlsanzeige Ein/Aus Dezimalstellen mit Punkt/Komma
/Heading	Bestimmung einer oder mehrerer Spalten oder Reihen als fixierte Spalten/Reihen
/Insert	Einfügen einer oder mehrerer leerer Spalten oder Reihen
/Join Anzeige	Auflistung aller bisher vorgenommenen Verknüpfungen zu anderen Arbeitsblättern
Hinzu	Zufügen einer neuen Verknüpfung zu einem anderen Arbeitsblatt
Löschen	Löschen der Verknüpfung mit einem anderen Arbeitsblatt
/Kritisch Weg	Errechnen und Aufzeigen des „kritischen Weges“
Freie Pufferz.	Errechnen und Aufzeigen des „kritischen Weges“ und der freien Pufferzeiten
Gesamz. Pufferz.	Errechnen und Aufzeigen des „kritischen Weges“ und der Gesamt-Pufferzeit
Bereich	Definition des Bereiches des „kritischen Weges“
Löschen	Löschen der Kennzeichnung des „kritischen Weges“
/Laden	Laden einer Datei (unterschiedlichen Formats) auf das Arbeitsblatt
/Move	Bewegung der aktuellen Reihe oder Spalte an eine andere Position
/Name Anzeige	Auflistung aller Namen von Zellen oder Zell-Blöcken
Hinzu	Zufügen eines Namens für eine Zelle oder einen Zell-Block
Löschen	Löschen eines Namens einer Zelle oder eines Zell-Blocks
/Output Drucker Speicher	Drucken der Arbeitsblattdaten Speichern der Arbeitsblattdaten auf einen Datenträger
Optionen	Festlegung des Druckertyps und d. Optionen
/Replcate	Vervielfältigung einer oder mehrerer Zeilen in regulärer oder rechtwinkliger (um 90 Grad gedreht) Anordnung

SOFTWARETEST

/Speichern	Speichern des Arbeitsblattes in einer LOGISTIX-, DIF- oder KGD-Datei
/Tabelle	
Ermittlung	Finden, Kopieren oder Löschen selektierter Datensätze einer Datenbank
Sortieren	Sortieren von Reihen eines Zell-Blockes gemäß der Kriterien bestimmter Spalten
Füllen	Füllen eines Zell-Blockes mit einer Folge von Zahlen
Was-Wenn	Erstellen einer Entscheidungstabelle, die die Auswirkung von Veränderungen in einer oder mehrerer Zellen aufzeigt
Laden	Laden eines selektierten Teiles einer dBase-Datei
/Utilities	
Status	Aufzeigen des Speicher-Status
Dateien	Auflistung des Inhaltsverzeichnis der Dateien
Kopieren	Kopieren einer Datei
Neuname	Umbenennung einer Datei
Löschen	Löschen einer Datei
Vorsilbe	Erstellen einer Vorsilbe, die die Zugehörigkeit zu Dateien definiert
/View	
Schirm	Anzeigen einer Grafik auf dem Bildschirm
Drucken	Druck einer Grafik
Plotter	Zeichnen einer Grafik auf einem Plotter
Bereich	Spezifizieren des aktuellen Grafik-Bereiches
Fenster	Spezifizieren des Grafikfensters
Color	Farben auf dem Bildschirm, dem Drucker oder Plotter aus den Farbtafeln
Optionen	Spezifizieren des Bildschirm-, Drucker- oder Plottertyps und Optionen
/Window	Aufteilung des Bildschirms in zwei Fenster; Anzeige der Ausdrücke
/Zap	Entfernung aller Daten vom Arbeitsblatt
	ACHTUNG! ERST SICHERN!
Xecute	
Begrenzung	Definieren einer Arbeitsblatt-Begrenzung mit Passwort
Lernen	Automatische Eingabe von Macros
Einzelschritt	Schrittweise Ausführung von Autos und Macros

>RCK	Entfernt d.links befindlichen Zeichen
>LOE	Löschen
>HLF	Hilfsanweisungen
>DAT	Datei-Anzeige
>TAS	Wartet, bis eine Taste gedrückt wird
>TEI	Wartet auf Dateneingabe
>TON	Signalton
>WARn	Wartet n-Sekunden (n: 1 - 99)
>WDTn	Wiederholt die vorhergehende Taste n-mal
>WID	Wiederholt eine Auto- oder Macro-Zeile

Zusammenfassende Bewertung:

Logistix, das wird der Leser sicher schon an der Befehlsfülle festgestellt haben, bietet dem Anwender eigentlich alles, was er zu der Projektplanung, zum Zeitmanagement, für die Tabellenkalkulation, für Datenbankzwecke und vor allem im Bereich der Grafik benötigt. Das Programm ist nicht in kurzer Zeit zu erfassen, dazu sind seine Möglichkeiten zu vielfältig. Das gute Handbuch, die ausgezeichneten Hilfe-Funktionen und die gute Unterstützung durch den Importeur bieten bei der Einarbeitung aber eine große Hilfe. Der Preis für Logistix liegt bei knapp 500 DM. Der Gegenwert, den man dafür erhält, ist enorm. Ich habe bisher noch kein professionelles Programm gesehen, das ähnliche Möglichkeiten bei gleichem Preis-/Leistungsverhältnis bietet. Kurz: Logistix ist absolut empfehlenswert.

Michael Groneberg

Auch die Strichstärken, die Symbole u.a.m. können aus einer großen Auswahl ausgesucht werden.

Die Darstellungen selbst können als:

- Einzel-Torten-Grafik
 - Mehrfach-Torten-Grafik
 - Balken-Grafik
 - Flächen-Grafik
 - Raster-Grafik
 - Streuungs-Grafik
 - Bereichs-Grafik
- dargestellt werden

Im Bereich Auto und Macro stellt

Logistix u. a. folgende Möglichkeiten zur Verfügung:

>CBR	Rechtsbewegung
>CBL	Linksbewegung
>CBO	Bewegung nach oben
>CBU	Bewegung nach unten
>ANF	Bewegung zur Anfangszeile
>END	Bewegung zur Endzeile
>SBL	Seitenbewegung nach links
>SBR	Seitenbewegung nach rechts
>SBU	Seitenbewegung nach unten
>SBO	Seitenbewegung nach oben
>EIN	Dateneingabetaste (ENTER o. RETURN)
>ESC	Abbruch einer Operation

Interplan Muhlert
Nymphenburger Str. 134
8000 München 19
Tel.: 0 89/ 123 40 66



ZU SCHNELL

Sidecar-Benutzer, die mit MS-DOS 3.2 arbeiten, werden eventuell Schwierigkeiten mit dem Formatbefehl haben. Unser Artikel erklärt das Problem und bietet eine Lösungsmöglichkeit an.

Bei manchen Sidecars (solchen, die mit den BASF-Laufwerken mit Doorlock-Funktion ausgestattet sind) scheint es nicht möglich zu sein, unter MS-DOS 3.2 oder PC-DOS 3.2 Disketten zu formatieren. Nach Drücken der ENTER-Taste zum Formatieren bricht das Programm sofort mit der Meldung „Spur 0 defekt – Medium unbrauchbar“ den Formatiervorgang ab. Unter anderen DOS-Versionen sind die Disketten jedoch sehr wohl brauchbar. Bei der Suche in Handbüchern und Manuals entdeckt man sehr schnell den Grund für diesen Formatierfehler: Er verbirgt sich hinter den Diskettenparametern des MS-DOS. Mit diesen Parametern werden die physikalischen Größen des Diskettenhandlings vorgegeben. Im Speicher stehen die Daten ab der Adresse 0000:0522 in 11 Bytes. Einen Überblick gibt Tabelle 1.

DOS 3.2 ist ungeduldig

In der Speicherzelle 0000:052C steht der Wert für die Zeit, die dem Diskettenlaufwerk gegeben wird, um die Nenndrehzahl zu erreichen. Dieser Parameter wird ab der Version 3.2 von MS-DOS kleiner angesetzt. Und genau das ist der Fehler beim Formatieren: Das Laufwerk erreicht nicht rechtzeitig seine Solldrehzahl, folglich können die Daten nicht korrekt auf die Diskette geschrieben werden. Bei der Verifizierung entdeckt der Rechner den Fehler, interpretiert ihn aber als Fehler des Speichermediums und bricht das Formatieren ab. So läßt sich auch erklären, warum MS-DOS 3.2 die externe 3-Zoll-Floppy ohne weiteres akzeptiert. Um dem Rechner diese Unart auszutreiben, muß man also diesen Wert verändern. Im ROM des PC stehen voreingestellte Werte, die dann von der IO.SYS (bzw. IBMBIO.COM beim PC-DOS) verändert werden. Dafür gibt es die Möglichkeit, ein kleines Assembler-Programm zu schreiben, das nach dem Systemstart aufgerufen wird und das die entsprechenden Werte in den Diskettenparameterblock hineinschreibt. Das Programm ist als Beispiel für die Veränderung der Motorhochlaufzeit in Listing 1 abgedruckt. Prinzipiell kann auf diese Art

und Weise jeder Diskettenparameter verändert werden. Ratsam ist es jedoch nur bei wenigen Werten, etwa der Steprate, der Motornachlauf- oder Motorhochlaufzeit. Bei Veränderung der anderen Werte kommt es sehr schnell zu den wildesten Diskettenfehlern. So traten beim Sidecar nach einer drastischen Verringerung der Kopf-Beruhigungszeit sporadisch Schreibfehler auf. Ohne weiteres sollte man sich nur an die Veränderung zeitbestimmender Werte in die unkritische Richtung wagen. Ansonsten hat man sich sehr schnell die wertvolle Diskettensammlung zerstört. Eine elegantere Methode, das System zu ändern, ist das „Patching“ des MS-DOS. Für die Einstellung der Motorhochlaufzeit ist diese Methode vorzuziehen, da dieser Wert ja bleibend verändert werden soll. Aber Vorsicht! Nie an den Originalen ändern, nur Backups patchen. Am sichersten ist es, zum Patchen eine neue Systemdiskette zu erstellen. Da beim Booten neue Werte in den Parameterblock geschrieben werden, braucht man diese nur ausfindig zu machen und zu verändern. Die Änderung wird an einer der verborgenen Dateien vorgenommen (IO.SYS bzw. IBMBIO.COM), was ein kleines Problem aufwirft. Der Debugger kann „hidden files“ zwar einlesen, aber nicht mehr schreiben. Wer im Besitz eines Hilfsprogramms ist, das verdeckte Dateien in unverdeckte umwandelt, sollte das auf der Kopie der Systemdiskette tun. Wer einen Dis-

Adresse	Bezeichnung	ROM	Wert 2.11	3.2
0522	Steprate (in ms)	12	12	12
	Rückstellverzögerung(*16 ms)	15	15	15
0523	DMA-Flag (LSB)	0	0	0
	Ladeverzögerung (& 2 ms)(1..7 Bit)	1	1	1
0524	Nachlaufdauer des Motor (& 55 ms)	37	37	37
0525	Sektorgröße (Tab. 2)	2	2	2
0526	Sektoren/Track	8	9	9
0527	GAP-Länge	2Ah	2Ah	2Ah
0528	Datenlänge	FFh	FFh	FFh
0529	GAP-Länge für Format	50h	50h	50h
052A	Full-Byte für Format	F6h	F6h	F6h
052B	Kopfberuhigungszeit(in ms)	25	15	15
052C	Hochlaufzeit für Motor (& 125 ms)	4	5	2

```

MOV     AH,09
MOV     DX,0115
INT     21
XOR     AX,AX
MOV     DS,AX
MOV     BYTE PTR [052C],04
MOV     AX,4C00
INT     21
DB      '0D,0A,'Wartezeit geändert',0d,0a,'$'
    
```

Listing 1. Dieses Programm kann im Debugger mit acs:100 direkt geschrieben und als .COM-Datei gespeichert werden

Tab. 1. Belegung des Disketten-Parameterblocks

kettenmonitor besitzt, kann die Dateien ebenfalls konvertieren. Dazu muß man im Directory (logischer Sektor 5) die Werte hinter den Dateinamen ändern. Bei Dateien, die im Inhaltsverzeichnis aufgelistet werden, enthält das erste Byte hinter dem Dateinamen den Wert 20h. Verdeckten Dateien wird der Wert 27h angehängt. Wenn man nun diesen Wert in 20h umändert, erscheinen die Dateien danach im Inhaltsverzeichnis der Diskette. Dann kann man die Datei auch vom Debugger aus auf Diskette schreiben. Nach der Änderung kann man die Dateien wieder verbergen. Wer nicht im Besitz eines Diskettenmonitors oder eines Hilfsprogrammes zur Änderung des Hidden-Status ist, braucht jedoch nicht den Mut zu verlieren. Auch in diesem Fall gibt es eine Möglichkeit, die Systemdateien wieder auf Datenträger zu bannen. Nur das Vorgehen ist etwas umständlich. Was man dazu braucht, ist eine leere, formatierte Diskette. Man lädt nun IO.SYS (bzw. IBMBIO.COM bei PC-DOS) mit dem Debugger von der Systemdiskette in den Rechner. Dann legt man die leere Diskette ein und schreibt die Datei darauf. Dadurch werden die Dateien als unverdeckte Dateien auf Diskette gespeichert, ein Zurückschreiben nach der Änderung der Diskettenparameter wird ermöglicht. Auf gleiche Weise wird noch die MSDOS.SYS (bzw. IBMDOS.COM) auf die neue Floppy übertragen. Daß die Systemdateien nicht mehr verdeckt sind, stört den Computer nicht im geringsten. Für den Rechner ist nur die Reihenfolge wichtig, da er sonst nicht booten kann. Außerdem braucht man die Systemdateien nur mit dem SYS-Befehl auf eine neue Diskette zu übertragen, und schon sind sie wieder „verschwunden“.

Zum Patchen lädt man den Debugger und sucht in der IO.SYS die Stellen, an denen die Werte im RAM abgeändert werden. Am einfachsten sucht man nach den Adressen, in denen der Parameterblock steht, also 0000:0522 ... :052C. Dabei kann es vorkommen, da man die eigentlich gesuchte Adresse nicht findet. Dann sollte man es einmal mit der vorausgehenden Adresse versuchen, da manchmal zwei Werte zugleich als Wort in den Parameterblock geschrie-

ben werden. Hat man den Wert gefunden, den man verändern will, kann man ihn mit der Debug-Funktion „e“ auf die gewünschte Größe abändern und die Datei mit dem „w“-Befehl auf Diskette zurückschreiben.

Ein Beispiel für eine Debug-Sitzung ist in Abb. 2 dargestellt. Danach müßte das Problem mit MS-DOS 3.2 ein für allemal gelöst sein.

(chk)

```

A:debug
-n io.sys
-l
-rcx
CX 3FOA
:
-scs:0 3FOA 2C 05
OF10:340F
-ucs:3400 3410
OF10:3400 02A12000      ADD     AH,[BX+DI+0020]  : An diese
OF10:3404 A32200        MOV     [0022],AX      : Stelle
OF10:3407 0E           PUSH   CS              : wurde
OF10:3408 1F           POP    DS              : der Wert
OF10:3409 E88E08        CU L   3C9A           : auch ver-
OF10:340C A01C00        MOV    AL,[001C]      : wendet. Nicht
OF10:340F 2C05         SUB    AL,05           : ändern!!!!
-scs:0 3FOA 2B 05
OF10:2A28
-ucs:2A20 2A30
OF10:2A20 3E           DS:
OF10:2A21 E227         LOOP  2A4A
OF10:2A23 FD           STD
OF10:2A24 7COB        JL    2A31
OF10:2A26 C7062B050F02 MOV    WORD PTR [052B],020F
OF10:2A2C C6062205DF   MOV    BYTE PTR [0522],DF
-ecs:2A2B
OF10:2A2B 02.04
-ucs:2A26 2A2B
OF10:2A26 C7062B050F04 MOV    WORD PTR [052B],040F
-w
Writing 3FOA bytes
-a
    
```

Abb. 2 Diese Debug-Sequenz zeigt das Umändern von MS-DOS 3.2

```

Debug-Befehle, die verwendet werden:

n - Benennung der zu bearbeitenden Datei

l - Laden der Datei

s - Suchen nach Inhalten von Speicherzellen
Syntax: scs:<untere Grenze> <obere Grenze>
<Suchwert> <Suchwert>...

u - Disassemblierte Ausgabe des angegebenen
Speicherbereiches
Syntax: ucs:<untere Grenze> <obere Grenze>

e - Füllen der angegebenen Adresse mit dem ein-
zugebenden Wert
Syntax: ecs:<Adresse> Danach wird der Wert,
der in der Speicherzelle steht, angezeigt und
die Eingabe eines neuen Wertes erwartet

w - Schreiben von n Bytes. Dabei entspricht n dem
Inhalt des CX-Registers.

r - Zeigt Registerinhalt an und ermöglicht eine
Änderung. Syntax: r<Register> Danach wird der
Inhalt des Registers angezeigt und die Eingabe
eines neuen Wertes erwartet. Nach dem Laden
enthält CX die Länge der Datei.

Das Kürzel "cs" bei den Befehlen bedeutet,
das sich alles auf das "Codesegment", also den
64k-Block bezieht, in den das Programm vom
Debugger geladen wurde.
    
```

Abb. 3 Erklärung der verwendeten Debug-Befehle



Der „Command Line Interpreter“ (CLI) ist eine äußerst flexible und komplexe Oberfläche, dagegen verzichtet die Alternative zum CLI, die Workbench, auf einige wichtige Features. Freilich ist durch den komplizierteren Aufbau das Erlernen des CLI nicht gerade einfach.

Nachdem ich im ersten Kursteil ausführlich auf zwei markante CLI (Command Line Interpreter) Befehle eingegangen bin, möchte ich an dieser Stelle auf das Geheimnis des 'execute'-Befehls mit seinem recht großen Umfeld eingehen.

Dieser Befehl startet eine Befehlsdatei. Dies sind Dateien, die aus CLI-Kommandos bestehen, wobei ein Befehl nach dem anderen aufgerufen und durchgeführt wird. Um eine solche Datei noch effizienter zu gestalten, gibt es Befehle, die speziell für diese Art von Dateien bestimmt sind. Darunter fallen Befehle wie 'echo', 'if', 'lab', 'skip' oder auch 'failat'. Mit ihnen hat es eine bestimmte Bewandnis: Im direkten Eingabemodus, wenn Sie also die Befehle direkt in die Tastatur des Rechners eingeben, ergeben sie keinen näheren Sinn. Zusammen mit Befehlsdateien aber haben die verschiedenen Kommandos doch einen recht großen Nutzen.

Befehlsdateien – wofür?

Den Sinn von Befehlsdateien möchte ich an einem kleinen Beispiel zeigen: Die bekannteste und bei jedem Booten der Workbench Diskette

auf tretende Befehlsdatei ist die 'startup-sequence', die sich im Ordner 's' der Diskette befindet. Schauen Sie sich diese Datei einmal an. Sie werden feststellen, welchen Nutzen sie bringt. Andere Gebiete für den Einsatz von Befehlsdateien sind unter anderem Programm-Compilierungen. Meist werden sogenannte Quell-Programme zuerst compiliert und danach noch durch einen Linker 'gejagt'. Diese beiden Schritte können hervorragend in Befehlsdateien implementiert werden.

Richtig genutzt

An dieser Stelle möchte ich auf die Erklärung des 'execute'-Befehls mit seinen dazugehörigen Kommandos eingehen. Beginnen wir also zunächst mit 'execute'. Die Syntax lautet: EXECUTE Befehlsdatei Argumente. 'Befehlsdatei' repräsentiert jene Datei, welche die eigentlichen Befehle enthält; 'Argument' ist ein Ausdruck, der eine Variable in der Befehlsdatei ersetzt. Weiterhin erlaubt der 'execute'-Befehl die Übergabe von sogenannten Parametern, die dann automatisch an der vorher bestimmten Stelle der Befehlsdatei eingefügt werden. Hierbei sind folgende Parameter zu verwenden: '.key' oder '.k' spezifiziert ein Argument, das über die Tastatur des Rechners eingegeben wird. '.dot Symbol' verändert das Zeichen '.' in 'Symbol'. '.bra Symbol' und '.ket Symbol' ändern das Zeichen '<' bzw '>' in 'Symbol'. '.dol Symbol' und '.dollar Symbol' ändert das '\$' Zeichen in das Zeichen 'Symbol' um. '.def Schlüsselwort Wert' definiert einen Wert für ein Schlüsselwort. '<Leerzeichen>' leitet eine Kommentarzeile ein. '<Return>' schließlich definiert eine leere Zeile in der Befehlsdatei.

Nun prüft das Betriebssystem des Amiga nach dem Starten einer Datei erst einmal, ob es sich überhaupt um eine Datei handelt, die mit 'execute' aufgerufen werden kann. Wenn ja, wird festgestellt, ob Parameter definiert worden sind. Wenn es sich nicht um eine Befehlsdatei handelt, wird eine Fehlermeldung ausgegeben.

Ich kann mir leicht vorstellen, daß die Handhabung mit solchen Parametern nicht leicht zu verstehen ist.

Bevor ich jedoch versuche, die Sache an einem Beispiel zu verdeutlichen, möchte ich auf andere Befehle näher eingehen, die mit Befehlsdateien zusammenhängen. Bevor ich den Befehl 'if' aufgreife, möchte ich 'failat' ein wenig näher beleuchten. Wie im AmigaDOS Handbuch nachzulesen ist, definiert 'failat' einen Fehlergrenzwert. Hierzu muß man wissen, daß AmigaDOS bei einer fehlerhaften Ausführung einen Code liefert, der größer als Null ist – jenachdem, wie schwerwiegend der Fehler ist, desto größer ist der zurückgegebene Fehlercode. Durch eine feste Definition eines Fehlergrenzwertes wird bestimmt, ob der Befehl abgebrochen oder fortgesetzt wird. In der Regel liegt dieser Wert zwischen 5 und 20. Der 'if' Befehl, dessen Syntax wie folgt festgelegt ist: if[not][warn][error][fail][Text1 EQ Text2][exists Name] [...]else[...]endif, gebraucht den Fehlercode für die Ausdrücke 'warn', 'error' und 'fail', wobei die 'if'-Bedingung bei einem Code von $>=5$, $>=10$ und $>=20$ erfüllt ist, jenachdem, welcher Ausdruck gewählt wurde. 'else' bietet eine Alternative, falls die 'if' Bedingung nicht erfüllt ist, und 'endif' schließt den 'if'-Befehl. Die Kommandos 'lab' und 'skip' sind ebenfalls nur für Befehlsdateien bestimmt, wobei ihre Funktionen eng zusammenhängen: 'lab' definiert eine Sprungmarke und 'skip' veranlaßt den Sprung dorthin. 'echo' bewirkt ähnliches wie der BASIC-Befehl 'print', druckt aber nur Zeichenketten auf den Bildschirm, dient also nur dazu, Kommentare aus einer Befehlsdatei auszugeben. Das Kommando 'quit' gestattet einen Ausprung aus einer mit 'execute' gestarteten Datei mit einem festen Fehlercode. Die genaueren Erklärungen der einzelnen Befehle können Sie dem mitgelieferten AmigaDOS-Handbuch entnehmen.

Befehlsdatei selbst erstellt

Um eine Befehlsdatei selbst zu erstellen, brauchen Sie nichts weiter zu tun, als den 'ED'-Editor des CLI aufzurufen und die gewünschten Befehle einzugeben. Nach dem Sichern des Programms können Sie die Datei dann mit dem 'execute'-Befehl starten.

Als Beispiel für eine Befehlsdatei möchte ich die 'startup-sequence' ein wenig abändern. Dazu möchte ich auch auf einige interessante Tips eingehen. Laden Sie zuerst die 'startup-sequence' in den Editor. Sie wissen: Diese Sequenz befindet sich im Directory 's' auf der Workbench-Diskette. Löschen Sie nun alle Zeilen, so daß keine Einträge mehr in der Datei stehen. Geben Sie nun Listing 1 ein und speichern Sie danach die neu erstellte 'startup-sequence' ab. Jetzt kann durch das erneute Booten die neue Sequenz begutachtet werden. Um die Kommentarzeilen schneller auf dem Bildschirm auszugeben, gibt es einen kleinen Trick: Hierzu erstellen Sie eine extra Datei mit einem separaten Namen; in dieser geben Sie nun nur die Kommentare ein und sichern die Datei auf der Workbench-Diskette im Haupt-Directory. In der 'startup-sequence' geben Sie statt der 'echo'-Zeilen 'type Kommentardatei' ein. 'Kommentardatei' repräsentiert hierbei den Namen ihres Files. Durch erneutes Booten können Sie dann leicht erkennen, daß der Text jetzt wesentlich schneller auf den Bildschirm ausgegeben wird.

Listing 2 verdeutlicht den genaueren Umgang mit der Parameterübergabe des 'execute'-Befehls, hierbei wurden alle wichtigen Details an einem Beispiel erläutert. Das Beispiel bewirkt die Ausgabe einer beliebigen Datei auf den Bildschirm, zuerst in 'normaler' Form und dann in der hexadezimalen Schreibweise, wobei sich die erste Ausgabe nicht für binäre Dateien eignet. Sie können durch Drücken der Tastenkombination CTRL-C die Ausgabe unterbrechen; die Befehlsdatei führt dann den nächsten Befehl aus. Noch ein Hinweis: Eine mit 'execute' gestartete Datei können Sie mit CTRL-D unterbrechen. Sie können das Listing wiederum mit dem System-Editor eingeben und danach mit dem 'execute'-Kommando starten. Sie können der Datei einen beliebigen Namen zuweisen, jedoch sollte er noch nicht existieren. Der Aufruf der selbsterstellten Datei muß dann wie folgt aussehen: execute 'Dateiname' 'Schlüsselwort'. Dabei repräsentiert 'Dateiname' die Befehlsdatei und 'Schlüsselwort' die ausdruckende Datei.

(AK)

Listing 1

```
echo ""
echo "      Eine neue **Startup-Sequenz**"
echo "Es werden alle AmigaDOS Befehle ins RAM kopiert"
echo "      deshalb dauert das Booten ein wenig länger !"
echo ""
echo "      Einen kleinen Moment noch !"
makedir ram:c
copy sys:c to ram:c all quiet
echo ""
echo "      Gleich fertig..."
echo "      nur noch die Workbench aufrufen."
assign c: ram:c
loadwb
echo ""
echo "      Puuh geschafft..."
wait 5      .wartet 5 Sekunden
```

Listing 2

```
.key file . 'file' wird hier als beliebige Datei aufgefaßt,
      . die nach der Befehlsdatei folgt.
if exists <file>
      type <file>
else
      echo "<file> kann nicht gefunden werden"
endif
. der erste Teil druckt eine beliebige Datei aus
. es folgt der selbe Abschnitt, jedoch mit anderen Parametern
.dot - - der Punkt '.' wird in Bindestrich '-' umgewandelt
-bra * - das '<' Zeichen wird mit '*' vertauscht
-ket # - das '>' Zeichen wird mit '#' vertauscht
if exists *file#
      type *file# opt h - hexadezimaler Ausdruck der Datei
else
      echo " *file# kann nicht gefunden werden"
endif
. Die execute Datei ist damit beendet
```



Sound Sampling

– Klang im Computer

In diesem Beitrag soll es um das Prinzip der Tonerzeugung des Amiga gehen. Wie digitalisiert man Klänge, wo liegen die Stärken und die Schwächen des Systems?

Lassen Sie uns zuerst überlegen, wie Klänge und Geräusche in der Natur überhaupt aussehen. Ein Geräusch verursacht Luftdruckschwankungen, die mehr oder weniger periodisch verlaufen. Diese Luftdruckschwankungen werden vom Ohr in Hörerfahrungen umgesetzt. Genauso kann man sie mit Hilfe eines Mikrofons in Spannungsschwankungen umsetzen und dann elektronisch weiterverarbeiten. Dabei ist dann entweder die vom Mikrophon erzeugte Spannung oder der erzeugte Strom dem Luftdruck proportional. Der Verlauf der elektronischen Größen entspricht dann dem Verlauf des Luftdrucks (Bild 1).

Die vom Mikrophon erzeugten Spannungen sind, wie auch die Luftdruckschwankungen, kontinuierliche, also analoge Größen. Damit kann ein Computer nichts anfangen. Wie bringt man einen Computer, der nur digitale Werte verarbeiten kann (also Folgen von Nullen und Einsen) dazu, natürliche Geräusche und Klänge zu verarbeiten? Man benötigt einen Analog-Digitalwandler, eine Schaltung also, die Spannungen oder Ströme in digitale Werte umsetzen kann.

Es gibt verschiedene Schaltungstechniken, die dies bewerkstelligen. Alle haben gemeinsam, daß sie keinen kontinuierlichen Datenstrom liefern. Sie nehmen sozusagen einen Schnappschuß aus dem elektrischen Signal heraus. Der Spannungswert dieses elektrischen Signals wird zu

einem bestimmten Zeitpunkt gemessen. Je öfter man diese Messung wiederholt, desto genauer repräsentieren die Messwerte das Originalsignal (Bild 2), und umso genauer kann das Originalsignal nach der digitalen Verarbeitung rekonstruiert werden.

Wie hoch diese sogenannte 'Sampling Rate' (engl.: Wandlungsrate mindestens sein muß, um eine gute Reproduktion zu ermöglichen, ergibt sich aus einem Theorem, das der amerikanische Physiker C.E. Shannon im Jahr 1948 formuliert hat. Da-

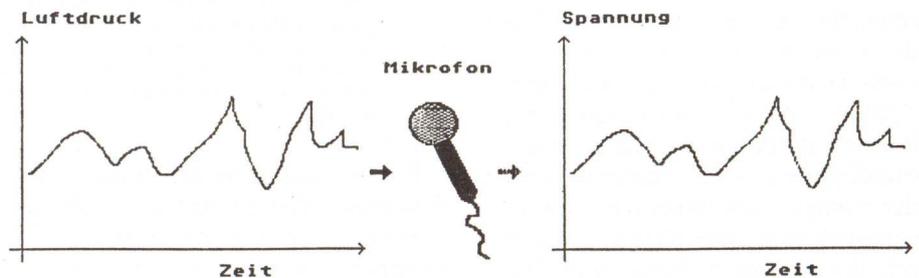


Bild 1: Das Mikrophon wandelt die Luftdruckschwankungen in einen Spannungsverlauf, der den Klang repräsentiert, um.

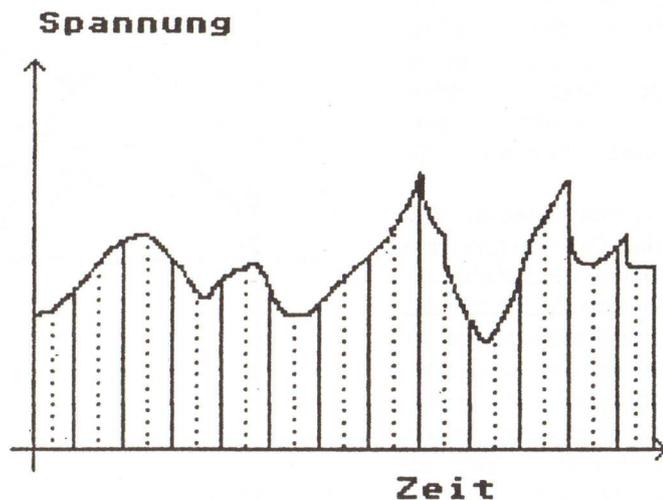


Bild 2: Das Analogsignal wird an den mit geschlossenen Linien markierten Stellen gemessen. Wie man am Bild sieht, kann das Originalsignal genauer durch die Meßwerte repräsentiert werden, wenn man die Abtastrate erhöht (Im Bild durch gestrichelte Linien angedeutet), also häufiger mißt. Feine Details des Originals gehen bei niedriger Abtastrate einfach verloren, was eine Klangveränderung und Verzerrungen verursacht.

nach muß bei jeder Form der Abtastung eines Signals die Abtastfrequenz, also die Wandlungsrate, mindestens doppelt so hoch sein wie die höchste im Signal vorkommende Frequenz.

Dabei ist es wichtig zu wissen, daß alle Klänge mit Ausnahme reiner Sinustöne aus einem Frequenzgemisch bestehen, dessen Grundton wir als die eigentliche Tonhöhe hören. Wenn eine Geige einen Ton in mittlerer Lage spielt, liegt die Grundfrequenz dieses Tones bei einigen hundert Hertz. Die höchsten Frequenzen, die in den Frequenzgemisch, das für den Geigenklang verantwortlich ist, vorhanden sind, reichen weit über 20 000 Hz hinaus. Mit einem mathematischen Analyse-Verfahren, der sogenannten Fourier-Analyse, kann man Klänge in ihre Komponenten zerlegen. Darauf baut auch eine Definition von Klang im Gegensatz zum Geräusch auf. Die Komponenten von Klängen sind im wesentlichen ganzzahlige Vielfache, die sogenannten harmonischen oder Obertöne des Grundtones. Zu diesen Teiltönen kommen noch relativ geringe Anteile von Frequenzen, die keine ganzzahligen Vielfache der Grundfrequenz sind, sondern mehr oder weniger kontinuierlich im Frequenzspektrum eines Klanges verteilt sind. Solche, meist nichtperiodische Frequenzgemische nennt man, wenn sie alleine vorkommen, Geräusch. Das Ohr kann glücklicherweise auch ohne höhere Mathematik sehr gut zwischen beidem unterscheiden. Ein Flügel hat im Bereich der tiefen Frequenzen einen gewissen Geräuschanteil, während der Klang von Schlaginstrumenten wie Trommeln praktisch nur aus einem Geräuschspektrum besteht.

Für den Klang eines Instrumentes ist vor allem das Obertonspektrum und dessen zeitlicher Verlauf charakteristisch. Ist bei einer Aufnahme der Frequenzgang des Aufnahmeegerätes zu schlecht, werden Teile des Obertonspektrums verändert und das aufgenommene Instrument klingt unecht. Jeder kennt diesen Effekt vom Telefon. Der Frequenzgang der Telefonmikrofone ist extrem gering, so daß nicht einmal der volle Tonumfang der Stimme übertragen werden kann. Also hört sich eine Stimme im Telefon verändert an.

Aber nun zurück zum Abtasttheorem. Um also den vollen HiFi-Frequenzgang bis 20 000 Hz übertragen zu können, muß man eine Abtastrate von mindestens 40 000 Hz verwenden. Es gibt praktische Gründe, die dazu führen, daß man tatsächlich etwas mehr als die doppelte Rate verwenden muß (bei CD-Spielern benutzt man zum Beispiel 44,1 KHz), aber dazu später mehr.

ten reproduzieren können. Es gibt aber von verschiedenen Herstellern entsprechende Zusatzgeräte. Den Test eines solchen Systems finden Sie in diesem Heft.

Die Qualität einer digitalen Signalübertragung hängt jedoch nicht nur von der Häufigkeit der Abtastung ab. Auch die Genauigkeit der Messung zu einem bestimmten Zeitpunkt ist äußerst wichtig (Bild 3). Es ist ein-

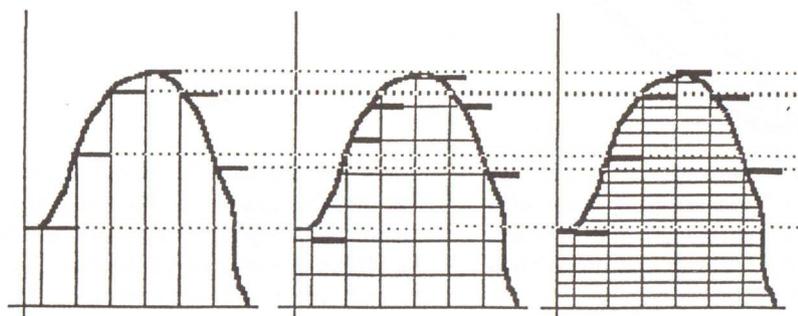


Bild 3: Die Auflösung wirkt sich erheblich auf die Qualität der Digitalisierung aus. Im rechten Bild sind die 'richtigen' Messwerte für den abgebildeten Signalauschnitt eingetragen. Im mittleren Bild wurden die Meßwerte auf ein 7-stufiges Meßraster gerundet. Wie Sie sehen, ist die Abweichung zu den richtigen Werten (gepunktete Linien) recht groß. Im rechten Bild wurde ein 19-stufiges verwendet. Die Genauigkeit ist erheblich besser. Im professionellen Bereich benutzt man 16 Bit Auflösung, also ein aus 65536 Stufen bestehendes Raster.

Ein Analog/Digitalwandler, mit dem man selbst Klänge digitalisieren könnte, ist im Amiga übrigens nicht eingebaut, wohl aber vier Digital/Analogwandler, die umgekehrt das Originalsignal aus den digitalen Wer-

sichtig, daß ein Signal nur dann genau reproduziert werden kann, wenn man es sehr exakt gemessen hat. Heute verwendet man in der HiFi-Technik 16 Bit Meßgenauigkeit, das heißt, man kann jeden Spannungswert, z. B. zwischen -1 und 1 Volt durch binäre Werte zwischen -32 768

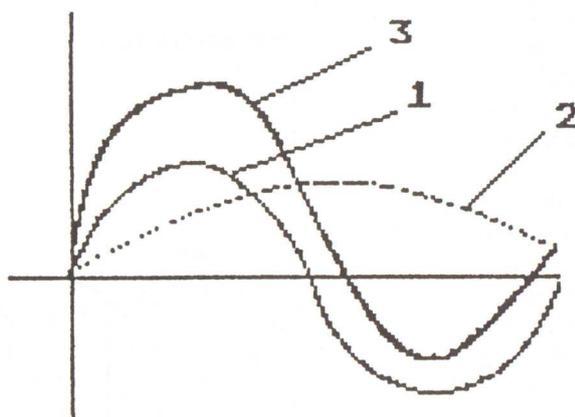


Bild 4: Schwingungssynthese nach Fourier. Der französische Mathematiker Fourier führte den Nachweis, daß sich beliebige periodische Schwingungen auf Additionen verschiedener Sinus- und Cosinuswellen zurückführen lassen. In unserem Beispiel ergibt sich die asymmetrische Kurve Nummer 3 aus der Addition der Sinus wellen 1 und 2, die unterschiedliche Frequenz und Amplitude besitzen. Die mathematischen Grundlagen für die Fouriersynthese sind recht kompliziert. Das Prinzip läßt sich übrigens auch für die Analyse von Schwingungen verwenden. Noch erheblich komplizierter ist die FM-Synthese, die im Gegensatz zur Fouriersynthese auch fast jeder anschaulichen Basis entbehrt.

und 32768 repräsentieren. Die Tongeneratoren des Amiga haben eine Genauigkeit von 8 Bit. Das heißt, sie erzeugen einen Spannungswert aus digitalen Werten zwischen -128 und $+128$.

Die Wiedergabe von digitalisierten Klängen funktioniert einfach umgekehrt. Ein Digital/Analogwandler verwandelt mit der gleichen Wandlungsrate, mit der die Klänge zuvor digitalisiert wurden, die binären Werte wieder in analoge Signale.

Gegenüber anderen Verfahren für die Tonerzeugung hat die Sampling-Technik den Vorteil, daß sich nahezu jeder natürliche Klang ziemlich originalgetreu (von Samplingrate und Auflösung abhängig) wiedergeben läßt. Darüber hinaus kann mit Hilfe geeigneter Software nahezu jede andere Klangsynthese simuliert werden – schließlich müssen die digitalen Werte, die über die Wandler ausgegeben werden, nicht unbedingt vorher gesampelt worden sein. Um zum Beispiel einen Sinuston zu erzeugen, kann man einfach die entsprechenden Werte des Sinus berechnen und in ihrer Reihenfolge über den Wandler ausgeben. Mit komplizierteren Synthese-Techniken, wie zum Beispiel der Fourier- (Bild 4) oder FM-Synthese sind auch sehr komplexe Klänge möglich, die kaum etwas mit natürlichen Klängen zu tun haben.

Ganz allgemein hat die Digitaltechnik bei der Signalverarbeitung den Vorteil, daß die Weiterverarbeitung nach der Analog/Digitalwandlung im Gegensatz zur Analogtechnik ohne jeden Qualitätsverlust möglich ist. Da ein Digital-Signal nur noch aus Folgen von Nullen und Einsen besteht, ist die Störsicherheit sehr groß. Es muß schon einiges passieren, damit aus einer Null eine Eins wird; davon abgesehen, daß man solche Fehler durch geeignete Kodierungsverfahren in vielen Fällen reparieren kann. Bei einem Analogsystem verändert jede Störung sofort den Klang. Der Unterschied läßt sich ungefähr so verdeutlichen: Wenn Sie in lauter Umgebung mit jemandem sprechen, stört Sie der Lärm in der Umgebung. Wenn Sie Ihre Sprache aufs Papier bringen, den Klang also durch Buchstabenfolgen repräsentieren, kann der Lärm diese Symbolisierung nicht beeinträchtigen. Ein weiterer Vorteil ist

die extrem einfache Bearbeitung digitaler Signale mit Hilfe von Computern. Die Herstellung und Bearbeitung von Musik wird dadurch sehr erleichtert.

Probleme ergeben sich bei der Digitalisierung in verschiedenen Bereichen.

Ein A/D-Wandler darf nicht übersteuert werden. Im Gegensatz zu einem analogen System, das „weich“ übersteuert, gibt es bei digitalen Systemen eine scharfe Grenze, bei der jede noch so geringe Überschreitung zu stark hörbaren Verzerrungen führt. Bei analogen Systemen steigen die Verzerrungen langsam mit der Übersteuerung an, eine geringe Übersteuerung ist praktisch nicht hörbar. Bei Digitalaufnahmen, vor allem wenn sie live ohne Korrekturmöglichkeit gemacht werden, kann dies eine Menge Schwierigkeiten bereiten.

Andererseits darf man ein digitales System auch nicht zu niedrig ansteuern, weil der Rauschabstand digitaler Systeme dann stark abnimmt; das sogenannte Quantisierungsrauschen nimmt im Vergleich zum aufgenommenen Signal zu, weil bei zu niedrigem Eingangspegel ja nicht alle Bits der zur Verfügung stehenden Auflösung ausgenutzt werden können (Bild 5).

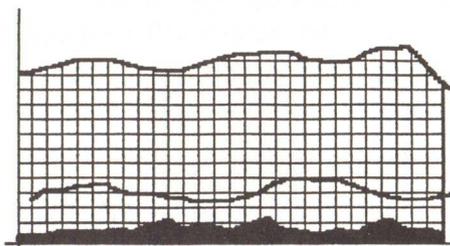


Bild 5: Das Quantisierungsrauschen wirkt sich erheblich stärker aus, wenn der Eingangspegel niedrig ist. Im Beispiel stehen für beide Signale die gleiche Anzahl von Quantisierungsstufen zur Verfügung. Das untere Signal nutzt jedoch nur ungefähr ein Drittel davon. Das Quantisierungsrauschen, dessen Pegel durch die schwarze Fläche unten angedeutet werden soll, ist jedoch in beiden Fällen gleich, da es im wesentlichen von der Größe des Rasters und nicht vom ankommenden Signal abhängig ist.

Das Quantisierungsrauschen ist ein anderes Problem, das sich direkt aus dem Prinzip der Digitalisierung ergibt. Die Spannungswerte in einem Analogsignal können nun einmal nahezu beliebige, kontinuierliche Werte annehmen. Mit den zur Verfügung stehenden digitalen Werten kann man aber nicht alle diese Werte exakt repräsentieren. Der Meßfehler wird im

Signal als Rauschen hörbar. Je weniger Stufen zur Verfügung stehen (je geringer also die Auflösung ist, je weniger Bits pro Abtastwert zur Verfügung stehen), desto größer ist dieses Quantisierungsrauschen. Bei modernen 16-Bit-Systemen ist dieses Rauschen allerdings niedriger als das Rauschen, das in den allermeisten Analogsystemen entsteht.

Es wurde bereits angesprochen, daß man normalerweise die Abtastrate höher wählt, als das Abtasttheorem es eigentlich erforderte. Das liegt daran, daß Frequenzen, die höher als die halbe Abtastrate sind, nicht einfach ignoriert werden, sondern als tiefe, sehr unharmonische Störungen wieder im Signal auftauchen (es bilden sich die Summen- und die Differenzfrequenzen von Samplingrate und dem zu digitalisierenden Signal). Mathematisch gilt das auch für Computerbilder:

Eine digitalisierte schräge Linie zeigt die computertypischen Treppchen, die ein Aliasing-Effekt sind. Um das bei der Klangwandlung zu vermeiden, schaltet man vor den Wandler einen Filter, der alle Frequenzen, die zu groß für die Abtastrate sind, ausfiltert. Leider haben alle Filter den kleinen Nachteil, daß sie nicht schlagartig ab einer be-

stimmten Frequenz sperren. Ein Filter beginnt weit vor der Grenzfrequenz, Signale abzdämpfen. Man kann diesen Effekt zwar fast beliebig umgehen, nur werden die Filter dann unbezahlbar. Also nimmt man lieber einen einfacheren Filter und dafür eine höhere Abtastrate.

Einen Filter benötigt man übrigens auch hinter dem Digital/Analogwand-

ler. Der Wandler liefert ja nur diskrete Werte, die unbedingt 'gerundet' werden müssen, um starke Quantisierungsverzerrungen zu vermeiden (Bild 6)

Der Wandler braucht ein unverändertes Signal während seiner Arbeit. Ein kleines Ärgernis bei digitaler Speicherung längerer Klangereignisse ist der hohe Speicherbedarf. Bei 8-Bit

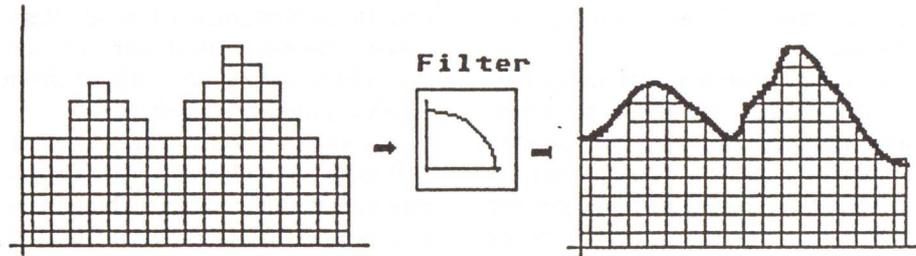


Bild 6: Das Glättungsfilter rekonstruiert den originalen Kurvenablauf aus dem gestuften, aber schon gewandelten Signal.

Beim Amiga ist dieser Glättungsfilter übrigens fest auf 7 KHz eingestellt. Dies hat zur Folge, daß auch der nutzbare Frequenzgang auch ca. 7 KHz beschränkt ist. Das reicht gerade für gute Wiedergabe von Sprachen aus, und dafür wurden Digital/Analog-Wandler des Amiga ursprünglich wohl auch konzipiert. Die Samplingrate kann allerdings erheblich höher als die doppelte Filterfrequenz eingestellt werden. Das bringt zwar nichts für den Frequenzgang, die Qualität wird aber dennoch verbessert, da das Originalsignal sich auf jeden Fall besser rekonstruieren läßt, wenn die Abtastwerte nahe beieinander liegen (Bild 7).

Wandlung und 10 KHz Abtastrate werden pro Sekunde Ton 10 KByte verbraucht. Wenn alle vier Generatoren des Amiga ein eigenes Programm bekommen sollen oder wenn man eine höhere Rate verwendet, sind das schon mindestens 40 KByte pro Sekunde. Deshalb haben viele Programme, die gesampelte Geräusche benutzen, ein recht beschränktes Repertoire. Aber rechnen Sie doch mal zum Spaß den Speicherplatz einer CD aus: ca. 70 Minuten, 44,1 KHz Samplingrate, 16-Bit Auflösung und das in Stereo, also mal 2. Dazu kommt noch einiges an Hilfsdaten. Ganz schön viel, nicht?

Zum Schluß soll noch eine Mög-

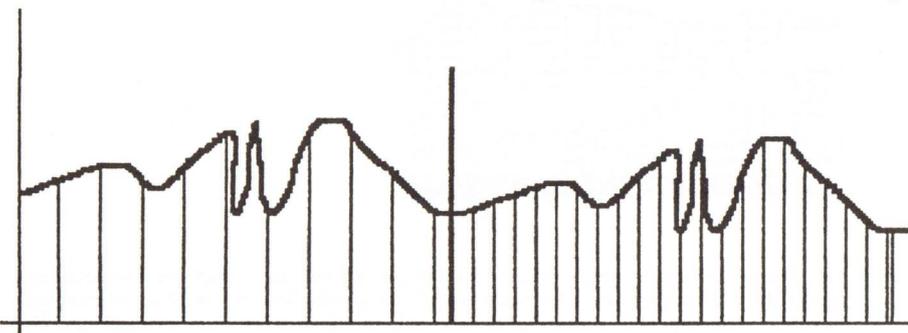


Bild 7: Eine höhere Abtastrate verbessert die Kurventreue. Die kurze Schwankung in der Mitte des linken Bildes wird bei der linken Abtastung ignoriert und nicht wieder rekonstruiert. Bei höherer Abtastrate können feinere Unterschiede wiederhergestellt werden. Das ist auch dann wichtig, wenn die höhere Rate wegen eines Glättungsfilters nicht zu einem besseren Frequenzgang führt – Der Klang verbessert sich.

Auf Bild 8 finden Sie noch einmal übersichtlich alle Teile einer digitalen Signalverarbeitung. Das Sample-and-Hold-Glied vor dem Wandler dient dazu, einen 'Schnappschuß' des Signals solange festzuhalten, bis der Wandler seine Arbeit beendet hat.

lichkeit der Bearbeitung von digitalisierten Klängen vorgestellt werden. Bisher sind wir immer davon ausgegangen, daß ein Klang möglichst originalgetreu aus den digitalen Werten rekonstruiert werden soll.

Der Traum bei der Entwicklung

elektrischer und elektronischer Musikinstrumente war, akustische Instrumente möglichst gut simulieren zu können.

Die ersten Versuche waren mehr oder weniger mechanischer Art. Die größte Bedeutung erlangte in den 70er Jahren ein Instrument namens 'Mellotron'. In diesem Gerät waren für alle Tasten der Klaviatur Endlos-tonbänder, die über eine gemeinsame Achse angetrieben wurden, eingebaut. Wenn eine Taste der Klaviatur gedrückt wurde, bewegte sich ein Tonkopf gegen das Band und der auf dem Band aufgezeichnete Ton wurde hörbar. Die wichtigsten Klänge, die auf Bändern lieferbar waren, waren Streicher- und Chorklänge. Das ganze Instrument war, der komplizierten Mechanik wegen, sehr teuer und anfällig und daher auch auf der Bühne völlig unbrauchbar. Außerdem konnte man Klänge nicht selbst aufnehmen.

Erst die Digitaltechnik erlaubte die Konstruktion vielseitigerer und einfacherer Geräte. Die ersten Musikcomputer im heutigen Sinne entstanden in den 70er Jahren nach Entwicklung der Mikroprozessortechnik. Die Veteranen aus dieser Zeit gibt es in weiterentwickelter Form bis heute zu Preisen von um die 100 000 bis über 900 000 DM. Diese Geräte heißen Fairlight CMI und Synclavier und sind bei den meisten modernen Popproduktionen zu hören. Inzwischen gibt es aber auch in Preisklassen von unter 10 000 DM sogenannte Sampling-Keyboards, die es erlauben, Klänge zu digitalisieren und über eine Klaviatur zu spielen. Dabei können die Klänge noch auf die unterschiedlichsten Arten bearbeitet werden. Wie funktioniert das?

Es ist nicht unbedingt erforderlich, einen digitalisierten Klang mit der gleichen Sampling-Rate wiederzugeben, mit der er aufgenommen wurde. Wenn eine höhere Rate verwendet wird, hört man mehr Schwingungen pro Sekunde, also einen höheren Ton. Nimmt man eine niedrigere Rate, klingt alles tiefer. Wenn man die Frequenzen entsprechend der Stimmung eines Musikinstrumentes verändert, kann man mit einem Ton, den man auf einer Taste gesampelt hat, ganze Melodien spielen. Beispiel: Man nimmt einen Ton eines

Klavieres auf. Diesen Ton kann man dann über mehrere Tasten verwenden. Leider gibt es dabei ein kleines Problem, das diese Technik limitiert: Der Klang eines Musikinstrumentes verändert sich, je nach Instrument manchmal schon in einem Bereich von wenigen Halbtönen. Erhöht man die Ausgabefrequenz zu stark, ergibt sich eine Art Micky-Maus-Effekt. Besonders gut kann man diese Veränderung bei gesampelter Sprache beobachten. Darüber hinaus wird lei-

der nicht nur der Klang verändert, sondern auch die Länge der Klänge: Wenn die digitalisierten Daten schneller ausgegeben werden, ist auch die Dauer des Klanges kürzer.

Um dieses Problem zu umgehen, muß man, um z. B. ein Klavier halbwegs brauchbar wiederzugeben, mehr als ein Sample machen. In der Praxis hat es sich bewährt, ungefähr auf jeder Terz oder Quart ein neues Sample aufzunehmen. Auf einer Tastatur von fünf Oktaven kommt man dabei

mit 12 bis 15 Samples für die 61 Tasten aus. Das liegt auch im Rahmen der bei bezahlbaren Samplern verfügbaren Speicherzeiten. Normalerweise stehen bis zu 30 Sekunden in der Preisklasse um 5000 DM zur Verfügung (bei 12-Bit-Auflösung). Das reicht, um brauchbare Reproduktionen von Naturklängen zu produzieren.

Falls Sie Interesse an diesem Thema haben, schreiben Sie uns. Wir werden uns bemühen, Sie detaillierter zu informieren.

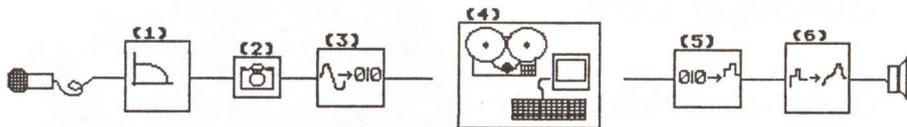


Bild 8: Eine vollständige digitale Übertragungskette. Das Signal, das vom Mikrofon kommt, läuft, nach ausreichender Verstärkung, in den Filter (1), der zu hohe Frequenzen abschneidet. Danach wird ein Schnappschuß im Sample-and-hold-Glied (2) festgehalten und vom Analog/Digitalwandler (3) digitalisiert. Wie oft das Sample-and-Hold-Glied einen Schnappschuß macht, wird von der Abtastrate bestimmt. Hinter dem Wandler folgt die eigentliche Verarbeitungseinheit (4). Das kann ein Computer oder ein digitales Tonbandgerät sein. Der Ausgang von (4) wird dem Digital/Analogwandler (5) zugeführt. Aus dem dabei entstehenden gestuften Analogsignal macht der Glättungsfilter (6) ein Signal, das mit einem Lautsprecher abgehört werden kann.

Inserentenverzeichnis:

AB-Computer	15	PDC	29, 65
ASH	51	Philgerma	11
CWTG	15	Raab	2
DATA Becker	45, 48	SoftwareLand	23, 100
Dolphin	51	Soyka	82
Kupke	13	Tröps	60
M.Ware	13	Versalia	15
Padercomp	47		

— Anzeigenschluß für die Ausgabe 9/87 ist der 24. Juli 1987 —

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)

Industriestraße 26
Postfach 55 69
6236 Eschborn
Tel.: 061 96/412 45
FAX: 061 96/411 37

Ständige Mitarbeiter:

Christian Keller (CHK)
Christian Schormann (CS)
Michael Kroneberg (MK)
Sven Schuler (SS)

Verlag:

Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt 13
Tel.: 061 51/5 60 57
FAX: 061 51/5 56 89

Verlagsleitung:

Hans-Jörg Heim

Anzeigenverkaufsleitung:

Uwe Heim

Anzeigenpreise:

nach Preisliste Nr. 1, gültig ab 1.7.86

Produktion:

Klaus Schultheis

Grafische Gestaltung:

Rosina Altkorn

Fotografie:

Klaus Ohlenschläger

Titelgestaltung:

Fabian & Maier

Titelfoto:

Rainer Spirandelli

Satz:

Mohr Pfungstadt

Druck:

Ferling Druck Darmstadt

Bezugsmöglichkeiten:

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 Datenver-
arbeitungsanlagen, sind nur mit schriftlicher Ge-
nehmigung des Herausgebers erlaubt.

Programmlistings, Bauanleitungen und Manu-
skripte werden von der Redaktion gerne entge-
gengenommen. Sie müssen frei von Rechten
Dritter sein. Mit ihrer Einsendung gibt der Ver-
fasser 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 Nichtfunktionie-
ren oder evtl. zum Schadhafwerden von Bau-
elementen führen, wird keine Haftung über-
nommen.

© Copyright Heim Verlag

VORSCHAU:

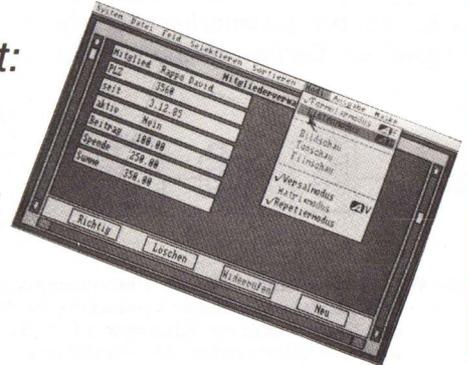
Softwaretest:

*ST-Emulator für den Amiga:
Gerücht oder Wahrheit?*

*Schweizer Wertarbeit:
GoAmiga! Datei*

*Basic-Beschleuniger:
AC Basic-Compiler*

*Nachfolger:
Die neueste Version von UBM-Text*



Hardware:

*Handarbeit:
Doppellaufwerk im Selbstbau*

Preiswerte Speichererweiterung im Selbstbau

MIDI-Interface im Test

Grundlagen:

*Darauf haben Sie lange lange gewartet:
Der C-Kurs beginnt*

*Ganz einfach:
Programmieren von Intuition mit AmigaBasic*

Ab 15. August an Ihrem Kiosk!

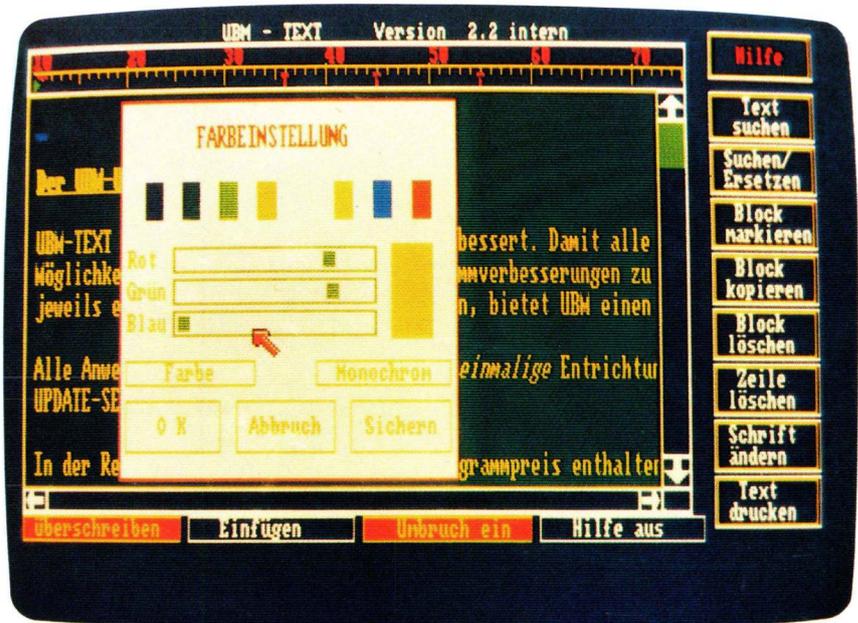
Jetzt gibt es etwas noch Besseres als UBM-TEXT-v2.1:

Die neue Version v2.2 ist da !

UBM-TEXT-v2.2

Das konnte schon die Version 2.1:

- Text suchen
- Suchen und ersetzen
- Blöcke markieren, kopieren, löschen, speichern, ...
- Textbausteine verarbeiten
- Phrasen speichern
- Schriftarten ändern
- Tastaturanpassung ändern
- Druckeranpassung ändern
- Textdarstellung wahlweise in 12 Zeilen mit 60 Zeichen oder in 20 Zeilen mit 80 Zeichen
- und natürlich alles, was man zum Erstellen und Bearbeiten von Texten braucht.



Der klare und übersichtliche Bildschirmaufbau blieb natürlich auch bei der neuen Programmversion erhalten.

Im Austausch gegen die Version 2.1 kostet die Version 2.2 nur 20,-DM!

Und das kann die Version 2.2 zusätzlich:

- **Kopf- und Fußzeilen**
Es können für jeden Text bis zu 10 Kopf- und Fußzeilen definiert werden, die dann auf jeder Seite automatisch ausgedruckt werden.
- automatischer Seitenumbruch
- Ausdruck von Seitennummern
- individuelle Druckeranpassung
- Drucken von Serienbriefen
- individuelle Farbeinstellung
- und einiges andere mehr ...

Trotz höherer Leistung konstanter Preis:

nur **249,-DM**
(unverbindliche Preisempfehlung)

UBM-TEXT läuft auf jedem Commodore-AMIGA ab Kickstart-Version 1.2

UBM-TEXT bekommen Sie in Deutschland

in allen guten Fachgeschäften, in einigen Warenhäusern oder direkt bei UBM.

in Österreich

bei Ueberreuter Media, Alser Str. 24, A-1091 Wien, Tel. 0222 481538-0

und in der Schweiz

bei ELEPRO AG, Im Langacker 6, CH-8304 Wallisellen, Tel. 073 411841



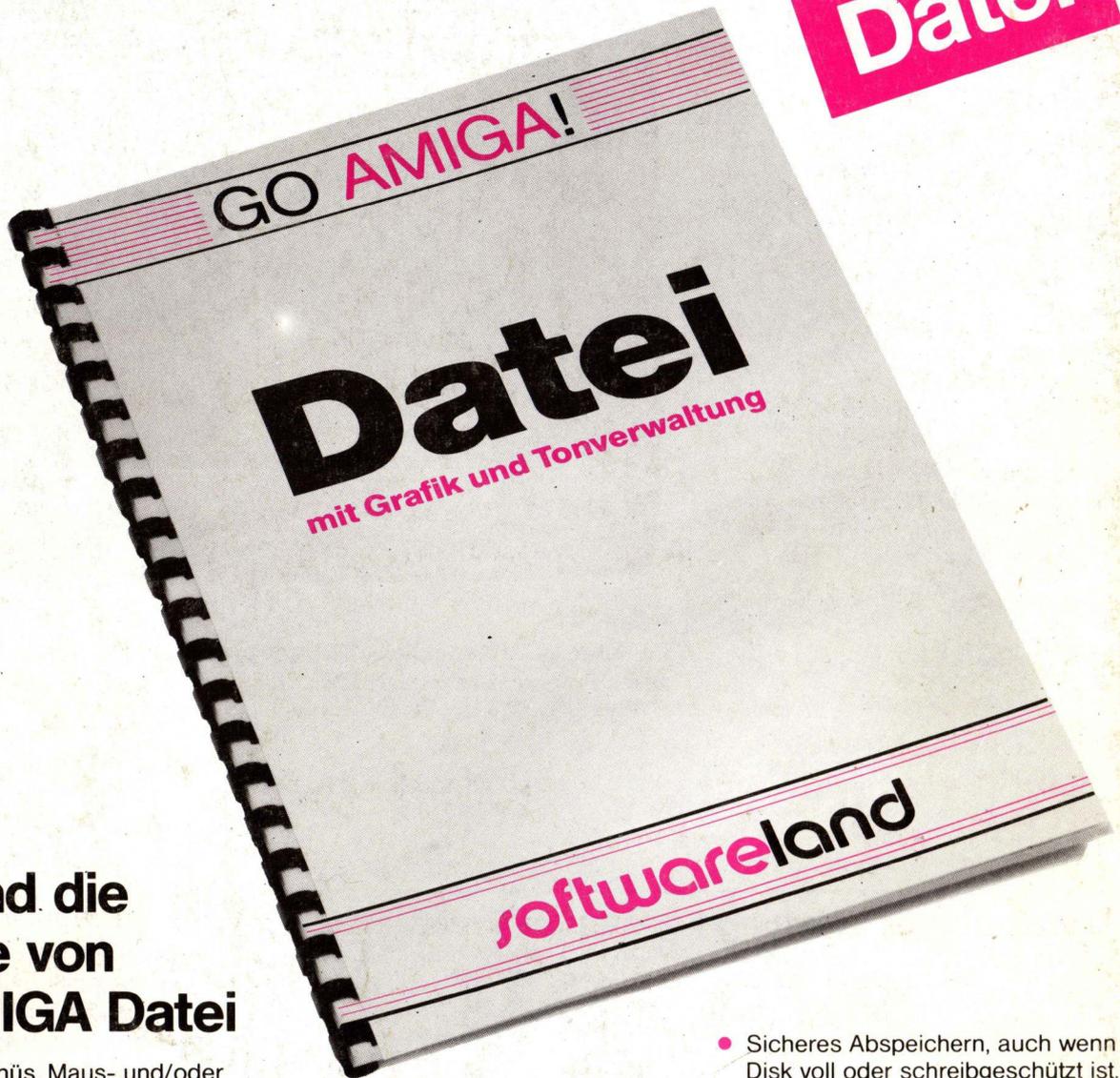
SOFTWARE made in Germany !

UBM Drecker GmbH
Baaken 4
2371 Hamdorf
Tel. (04332) 1634

Klasse statt Masse

GO AMIGA!

Datei



Das sind die Vorteile von GO AMIGA Datei

- Pull-Down Menüs, Maus- und/oder Tastatursteuerung.
- Arbeiten mit internem und externem Speicher (RAM), kompatibel zu Harddisk.
- Schnittstelle zu anderen Programmen mittels ASCII-Datei.
- Leistungsfähiges Such- und Sortierverfahren.
- Editieren der Bildschirmmasken während der Arbeit.
- Vielseitige, komfortable Druckersteuerung (Etikettendruck, Listendruck).
- Mehrere, individuelle Bildschirmmasken für ein und dieselbe Datei.
- Frei wählbare Darstellung von Zahlen, internationalen Währungen (z.B.: DM, sFR., ÖS).
- Mail Merge Funktion.
- Sicheres Abspeichern, auch wenn Disk voll oder schreibgeschützt ist.
- Diaschau – verwaltet und zeigt Grafikbilder am Bildschirm.
- Tonschau – verwaltet und spielt digitalisierte Töne.
- Filmschau – Dia- und Tonschau gleichzeitig.
- Ausführliches deutsches Handbuch.

Der Preis wird Sie erstaunen!
GO AMIGA Datei kostet DM 199,-
oder sFR. 178,-.

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)
zzgl. DM 7,- Porto. Händleranfragen erwünscht.

softwareland
Franklinstraße 27
CH-8050 Zürich (Schweiz)