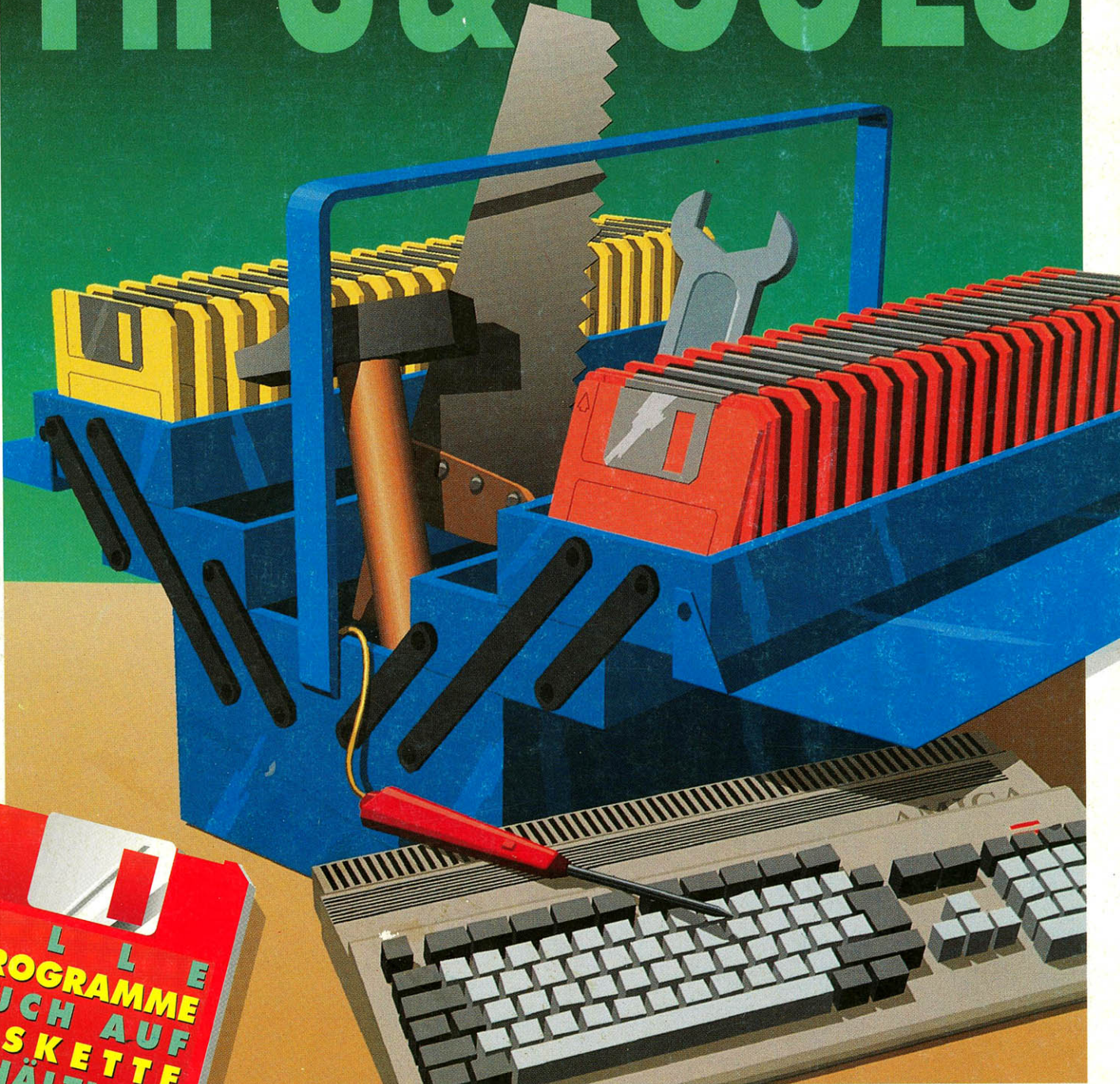


TIPS & TOOLS



ALLE
 PROGRAMME
 AUCH AUF
 DISKETTE
 ERHÄLTlich

Programmieren: *Gesammeltes Know-how zu Basic, Assembler, C, Modula und AmigaDos* ■ **Workshops:** *Diskey, Pixmate und MandelVroom* ■ **Hardware:** *Amiga-Tuning leichtgemacht* ■ **Top Software:** *30 Seiten Profi-Tricks*

N^o

8 COMPUTER-Flop LIVE aus dem Bundestag:



Wurden Millionen falsch investiert?



Computer-Einkauf in großem Stil... Für 40 Millionen Mark ließen die Bonner Politiker ein gigantisches Computernetz im Bundestag installieren. COMPUTER LIVE klärt, ob unsere Politiker überhaupt in der Lage sind, die teure „Parlakom-Anlage“ richtig zu nutzen. Und: Was halten Prominente wie Lothar Späth von „Politikern am Computer“?

Kaufberatung: Computersysteme für Einsteiger!

Ein großer Ratgeber klärt Sie über die wichtigsten Computersysteme auf und sagt Ihnen, welches System welche Ansprüche ideal erfüllt.

Perfekt: Reifenprofil per Computer!

Moderne Autoreifen müssen schier Unmögliches leisten. Computer im Einsatz für mehr Verkehrssicherheit – erfahren Sie, wie beim Reifengiganten Continental modernste Reifenprofile nur noch per Computer entwickelt werden können.

Pro und contra: Computer-Kauf im Versandhaus.

Billig-Angebote in Katalogen – mit Vorsicht zu genießen? Ob sie ihr Geld wirklich wert sind, mußten 6 Sonderangebote im harten Praxistext beweisen.

Erleben Sie die faszinierende Welt von COMPUTER LIVE!



Exklusiv-Interview mit Joschka Fischer: „Schädigen Computer unsere Umwelt?“

Holen Sie sich das August-Heft! Ab 26. Juli im Handel!

COMPUTER LIVE

DAS INTERNATIONALE COMPUTER-MAGAZIN

8

Markt&Technik

Wenn das Handbuch schweigt

Die meisten von Ihnen kennen die Schwierigkeiten, die einem Handbücher bereiten können. Oft sind sie zu dick, in Englisch geschrieben oder unübersichtlich strukturiert. Viele sind einfach nicht das, was sie dem Namen nach sein sollten, nämlich Handbücher. Andere wiederum bestehen aus einem schlichten Beipackzettel, bei dem sich allenfalls erahnen läßt, was das Programm leistet.

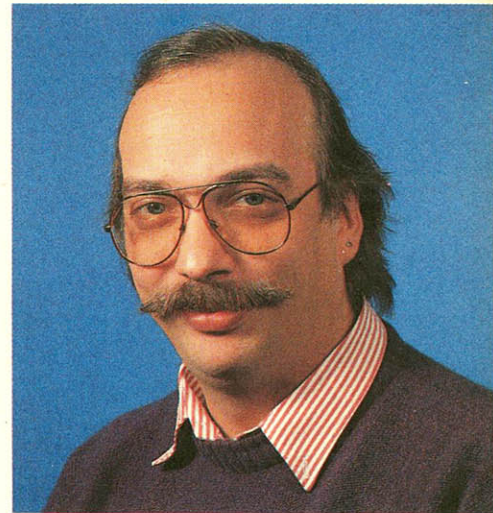
Gerade bei sehr komplexen Programmen, wie »Superbase« oder »DPaint III«, ist es schwer, sich schnell und dennoch ausführlich in das Programm einzuarbeiten. Oftmals fehlen wichtige Funktionsbeschreibungen, oder sie sind zu kompliziert dargestellt. Da findet man Aussagen, wie z.B. »Schieben Sie die Maus an den oberen, linken Rollbalken des geöffneten Fensters, ziehen diesen herunter, bis die gewünschte Darstellung auf dem Bildschirm erscheint, klicken Sie, während Sie gleichzeitig die Shift-Taste betätigen, die Icons nacheinander an und ziehen Sie das gesamte Gebilde in die ausgewählte Schublade.«

Bevor Sie nun »den Nippel durch die Lasche ziehen«, sollten Sie sich unser Heft näher betrachten. Wir haben, quer durch den Gemüsegarten, interessante Programme herausgepickt und näher beleuchtet, sprich Tips und Tricks geliefert. Auf den folgenden 114 Seiten finden Sie Hinweise zu den bekanntesten Anwendungsprogrammen wie beispielsweise »Superbase«, »Bekertext«, »Documentum« und »Wordperfect«.

Wir zeigen Ihnen, wie Sie die Druckerausgabe perfektionieren, sei es beim Etikettendruck, bei Serienbriefen oder bei Sonderzeichen.

Auch die Grafik-Freaks kommen nicht zu kurz: Mit dem Artikel über den Datentransfer zwischen »DPaint III« und »DVideo« kreieren Sie in kürzester Zeit eine Präsentation.

Wenn sich Ihr Handbuch auch weiterhin ausschweigt, verzagen Sie nicht, wir werden auch in Zukunft für Sie da sein.



Ihr Andreas Greil
(Leitender Redakteur)

»Mandelvroom« verwandelt trockene
Mathematik in interessante Bilder

SEITE 6

Workshops

6 Chaotische Fraktale

Mathematik mal ganz anders – als Grafik. Das geht! Mit dem PD-Programm »Mandelvroom« lassen sich die Funktionen der Chaos-Mathematik in Mandelbrotgrafiken verwandeln.

24 Faszinierende Bildmanipulationen

»Pixmate«, das universelle Grafiktool, wird auch Sie begeistern. Marco Vitolini-Naldini zeigt, was das Programm leistet, wenn Sie richtig damit umgehen.

29 Aufschlußreiche Innenansichten

Steigen Sie ein ins Innenleben Ihrer Disketten mit dem Shareware-Programm »Diskey«. Lernen Sie »Diskey« zu bedienen und Ihre Diskette zu verstehen.

Programmieren

44 Alles paletti mit Amiga-Basic

Die Fundgrube für Basic-Programmierer, egal ob Sie den Copper ansprechen, Speicher sparen oder Alerts ausgeben wollen.

46 Mehr Spaß mit C

Mit C geht alles, man muß nur wissen wie. Hier lernen auch Sie, dem Amiga seine Geheimnisse zu entlocken.

48 Tips zu Assembler

Ob rekursive Programmierung, Offsets oder Drucker- ausgabe in Assembler – Problem? Hier liegen Sie richtig.

50 Basteln mit Modula

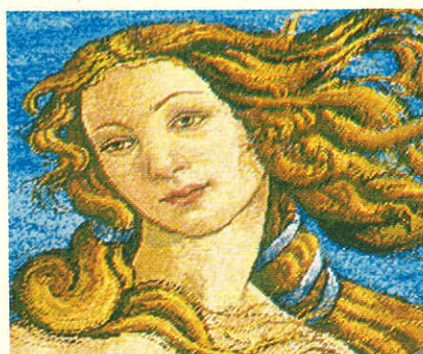
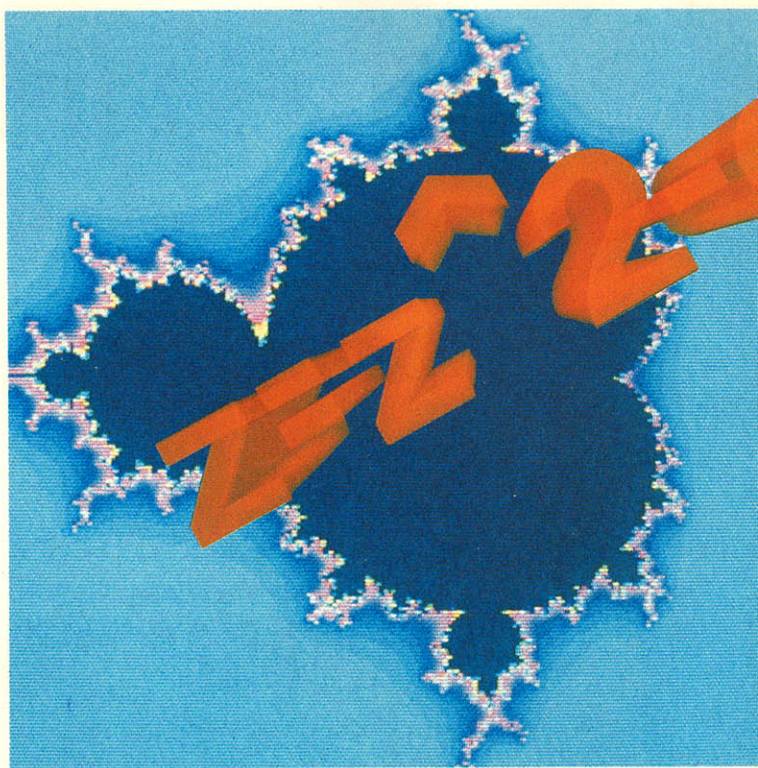
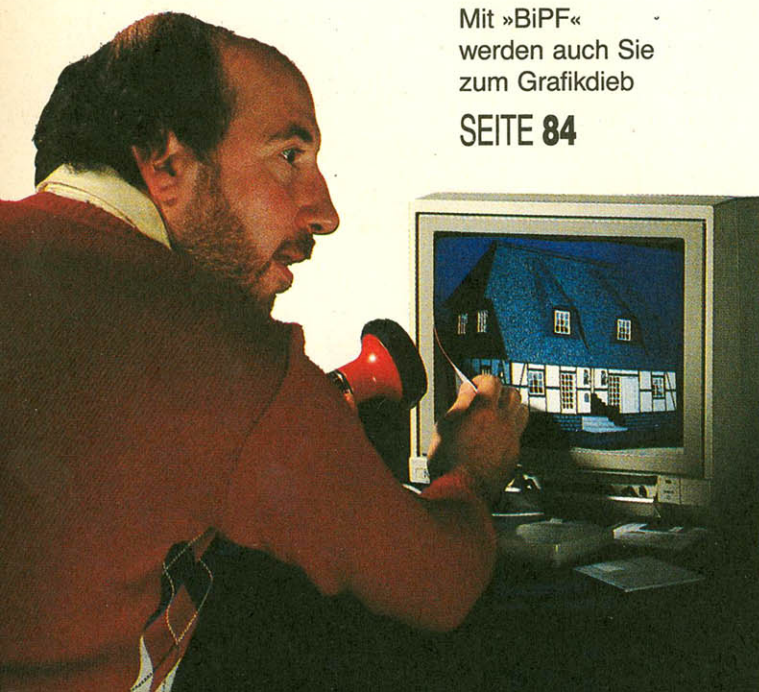
Bauen Sie sich Ihre Paßwortabfrage, oder greifen Sie direkt auf die Hardware zu. Mit den Tricks zu Modula keine Affäre.

52 Durchblick bei Amiga-DOS

Die Toolbox für den Amiga-User; hier finden Sie alles über Steuersequenzen im CLI und vieles mehr.

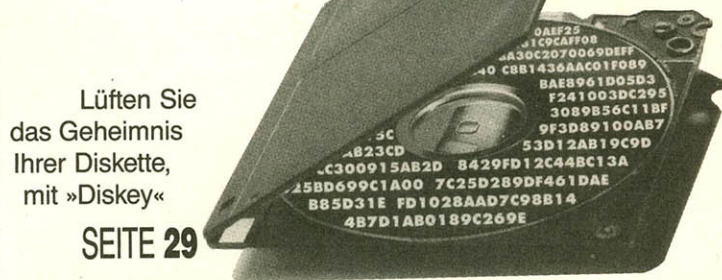
Mit »BiPF«
werden auch Sie
zum Grafikdieb

SEITE 84



Lernen Sie mit
»Pixmate« Ihre Bilder
zu retuschieren

SEITE 24



Lüften Sie
das Geheimnis
Ihrer Diskette,
mit »Diskey«

SEITE 29

Anwendungen

54 Der Servicetechniker

Sagen Sie uns, wo es hakt und unser Servicemann zeigt Ihnen den richtigen Weg. Eine komplette Übersicht der Tips.

56 Mit List & Tücke

Beckertext, das verbreitetste Textsystem für den Amiga, hat so seine Tücken. Wir kennen die Tricks, sie zu umgehen.

60 Das professionelle Textsystem

»Documentum« ist zwar eine einfache Textverarbeitung, die dennoch Features bereithält, die das Handbuch verschweigt. Wir brechen die sieben Siegel.

POWER-DISC 2

63 So geht's richtig

Wer »Wordperfect« vom PC her kennt, erlebt bei der Amiga-Version sein blaues Wunder. Damit Sie nicht auch reinfallen, zeigen wir, wo's lang geht.

65 Rock on the rocks

Lernen Sie aus den Erfahrungen des »Sonix«-Profis Martin Jobst, Ihre Kompositionen einfacher zu gestalten.

68 Trick für Trick ein Hit

DML ist eine leistungsfähige Programmiersprache für »Superbase«. Lesen Sie, wie man ihre Fähigkeiten nutzt.

72 Schweigen ist Gold, Silver macht mehr

»Turbo Silver« ist eines der wenigen Raytracing-Programme, in dem man Körper mit Texturen umspannen kann. Ein Fachmann zeigt, was Sie beachten müssen.

75 Im Rampenlicht

Strahlenverfolgung ist mit »Sculpt« ein Kinderspiel. Doch das Ergebnis ist meist gar nicht so berauschend. Damit Sie Ihre Bilder ins richtige Licht setzen, lesen Sie diesen Artikel.

78 Deluxe Paint ruft Deluxe Video...

Erstellen Sie mit uns eine komplette Computerpräsentation anhand einer Tierbibliothek. Sie werden staunen, was »Deluxe Paint« und »Deluxe Video« leisten.

Listings

84 Der Grafikräuber

Möchten Sie Grafiken aus anderen Programmen ergaunern? Mit »BiPF« werden Sie zum Meisterdieb.

95 CLI ade

Endlich ist es da, das »Disk-Utility«. Mit diesem Tool hat das CLI ausgedient. Arbeiten Sie mit Amiga-DOS per Mausclick.

105 Datenschutz für jedermann

Daten zu verschlüsseln ist zeitraubend und nicht sicher. Schützen Sie doch Ihr ganzes System mit »PassWd«.

Hardware

108 Mit Volldampf voraus

Da geht die Post ab. Betreiben Sie Ihre XT-Karte mit 8 MByte Taktfrequenz. Die High-Speed-Platine macht's möglich.

110 Ein paar Byte mehr

Eine 512-KByte-RAM-Erweiterung für den Amiga 500 unter 130 Mark. Das glauben Sie nicht? Wir zeigen Ihnen, wie.


111 Selbst ist der Mann

Der Amiga hält sich bei der Laufwerkschnittstelle leider nicht an den Industriestandard. Wie Sie trotzdem preiswerte Rohlaufwerke am Amiga betreiben können, lesen Sie hier.

Sonstiges

3 Editorial

96 Impressum

Alle Programme aus Artikeln mit einem -Symbol finden Sie auch auf der Programmservice-Diskette zu diesem Sonderheft

Das Super-Softwarepaket für den Amiga-User.

»Designers Pencil«, das Grafikstudio fürs Wohnzimmer: Über 50 Funktionen und eine komfortable Bedienoberfläche lassen Ihrer künstlerischen Ader freien Lauf.


»Penalty«, die realistische Eishockey-Simulation: Hier kommen Sie und Ihr Joystick ins Schwitzen.

»Sound Construction Set«: ein priffiges Tonstudio für den Amiga.

»Patch«: das Geschicklichkeitsspiel für Könner.



Die POWER-DISC 2 finden Sie ab 12.9. an Ihrem Kiosk

The background is a vibrant blue with a fine, woven texture. A complex, colorful fractal pattern, resembling a Mandelbrot set, is rendered in shades of pink, yellow, and blue, extending across the upper and right portions of the frame. In the lower right, a large, 3D, orange arrow points towards the right, partially overlapping the fractal. The overall composition is dynamic and mathematical.

Mit Mathematik verbindet man im allgemeinen Begriffe wie »trocken«, »staubig« und »graue Theorie«. Daß das nicht immer zutrifft, beweist uns Kevin Clague mit seinem faszinierenden PD-Programm »Mandelvroom«.

Chaotische Fraktale



von Jürgen K. Singer und
Thomas Hauser

Die Geburtsstunde der Fraktal-Mathematik und ihrer Umsetzung in die Welt der Grafik 1980 wurde eingeläutet. Damals beschäftigte sich der amerikanische Mathematiker Benoit Mandelbrot mit komplexen Zahlen in rekursiven Formeln. Nicht genug, er setzte sich auch noch in den Kopf, die Ergebnisse dieser Berechnungen in Grafiken umzusetzen. Das Resultat war sensationell. Ungeachtet der Auswirkung fraktaler Mathe-

matik auf alle Wissenschaftszweige bis hin zur Meteorologie war da etwas entstanden, was die Strenge der Mathematik mit der Schönheit der Kunst verknüpfte – die Mandelbrot-Menge.

Kaum war die Mandelbrot-Menge publiziert, stürzten sich nicht nur die Wissenschaftler der Welt darauf, sondern auch so mancher Computer-Freak. Die schnelle CPU des Amiga macht ihn zu einem potenten Partner für alle Freunde der fraktalen Grafik. Damit die Amiga-Fähigkeiten auch in Apfelmännchen umgesetzt werden können, haben begabte Programmierer die notwendige Software geschrieben. Eines der besten Programme dafür ist Kevin Clagues »Mandelvroom«.

Zunächst einmal brauchen Sie »Mandelvroom«. »AmigaLibDisk__-215« aus der Fred-Fish-PD-Serie ist voll damit. Natürlich benötigt das Programm nicht die vollen 880 KByte, die auf der Diskette Platz haben. Das Programm selbst ist 131 368 Byte groß – den Rest der Diskette füllen eine Unmenge fertiger Bilder, eine umfangreiche Doc-Datei und die Help-Files. »Mandelvroom« bietet Ihnen zu jedem einzelnen Menüpunkt eine Erklärung, die aus dem laufenden Programm aufgerufen

werden kann. Kevin Clague und Serenella Ciongoli haben eine umfangreiche Dokumentation zusammengestellt. Und wenn Sie es ganz genau wissen wollen, finden Sie auf »AmigaLibDisk_214« den Quellcode zu »Mandelvroom«. Was brauchen Sie noch? Ah ja, auf Ihrer Boot-Diskette muß im Libs-Ordner zumindest die Bibliothek »mathieedoubbas.library« vorhanden sein. Sollten Sie andere als die voreingestellten Rechenfunktionen verwenden wollen, brauchen Sie zusätzlich die »mathtrans.library«.

Starten Sie »Mandelvroom« entweder von der Workbench durch Anklicken des Icons oder dem CLI mit der Eingabe von

```
run Mandelvroom
```

»Mandelvroom« wird dann mit den Default-Parametern geladen und ruft als erstes Projekt die Datei »big-brot« auf. Wenn Sie »Mandelvroom« mit einem anderen Bild beginnen wollen, haben Sie wieder zwei Möglichkeiten: Entweder Sie öffnen den »Projects«-Ordner der Workbench und klicken dort eines der Projekt-Icons an oder Sie geben im CLI ein:

```
run Mandelvroom Projects/Dateiname1 Projects/Dateiname2
```

Wenn Sie mehr als ein File aufrufen, werden Ihnen die Grenzen der Wundermaschine Amiga bewußt gemacht: Es ist immer die Farbpalette des zuletzt aufgerufenen Files für alle vorhandenen Projekte aktiv. »Man-

Die Menüpunkte

delvroom« beherrscht zwar den Umgang mit 64 Farben, aber wenn die Farbpaletten der einzelnen Projekte zu weit auseinanderliegen, muß sich das Programm an den Fähigkeiten der Hardware orientieren.

Versuchen wir nun die Fähigkeiten von »Mandelvroom« kennenzulernen. Rufen Sie das Programm ohne Zusatzparameter auf, egal ob von der Workbench oder aus dem CLI. Wenn Sie jetzt noch die rechte Maustaste drücken und den Menüpunkt

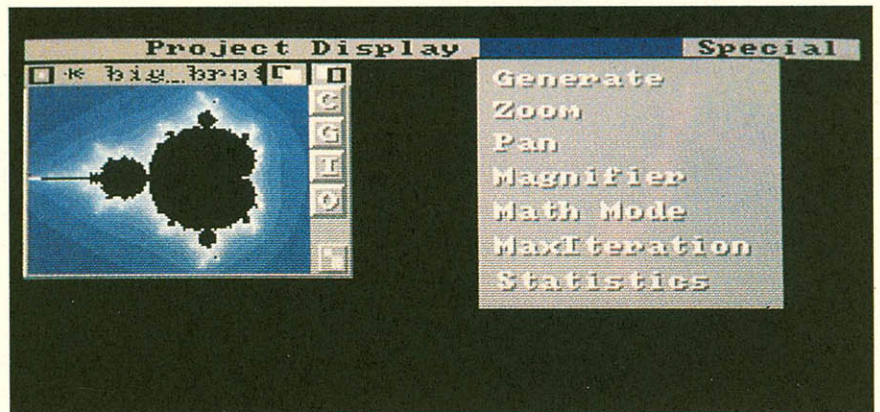


Bild 1. »Mandelvroom« – die Visitenkarte

»Calculate« anwählen, sehen Sie genau das, was wir in Bild 1 dargestellt haben.

Beginnen wollen wir mit dem ersten Menüpunkt ganz links:

Project-Menü

Als »Projekt« bezeichnet Kevin Clague jedes Mandelbrot- oder Julia-Fenster. Welches das aktuelle Projekt ist, erkennen Sie am kleinen Sternchen in der Titelzeile des Projekt-Window gleich rechts neben dem »Close«-Gadget. Da »Mandelvroom« voll multitaskingfähig ist, wird die Anzahl der gleichzeitig laufenden Projekte nur durch den freien Speicher begrenzt.

New

»Neu« enthält zwei Auswahl-Punkte: *Mand:*

In beiden Fällen werden die Grunddaten des aktuellen Projekts in ein neues Projekt übernommen, ohne das alte zu beeinflussen. Die Unterschiede zwischen Mandelbrot- und Julia-Mengen wird im zweiten Teil dieses Artikels (Seite 17) anschaulich beschrieben. Wählen Sie »New/Mand«, erscheint zunächst ein neues, inhaltsloses Fenster. Wenn Sie nun das »G«-Gadget des neuen Fensters anklicken, erhalten Sie eine exakte Kopie des Mutterprojekts. Anders sieht die Sache schon aus, wenn Sie nach dem Öffnen des Tochterfensters eines der Zoom-Gadgets anklicken. Für Ausschnittsvergrößerungen ist das Gadget »I« zuständig (»I« wie »Zoom In«), für eine »distanziertere« Betrachtung des Apfelmännchens muß das Gadget »O« herhalten (das »O« finden Sie in »Zoom Out«). Es gibt für die Anwendung der beiden Zoom-Gadgets zwei Wege:

Möglichkeit 1: Sie wollen mit einer grob gerasterten Lupe den Mutterprozeß untersuchen. Dazu klicken Sie zunächst im Tochterfenster das Gadget »I« an. Der Mauszeiger bekommt daraufhin ein Anhängsel namens »To«. Wenn Sie jetzt mit dem Mauszeiger in das Fenster des Mutterprozesses fahren und die linke Maustaste drücken, markieren Sie damit die linke obere Ecke des zu vergrößernden Bereichs. Halten Sie die Maustaste gedrückt und verschieben die Maus. Es bildet sich ein Rahmen. Und wie in Bild 2 demonstriert, sehen Sie dann im Fenster des Tochterprozesses die Pixel-Vergrößerung des eingerahmten Bereichs. Die Darstellung im Tochterfenster folgt unmittelbar den Lupen-Dimensionen im Mutterfenster. Es werden hier keine neuen Berechnungen angestellt, sondern je nach Lupengröße werden die Pixel im Mutterfenster vervielfältigt im Tochterfenster dargestellt.

Möglichkeit 2: Sie wollen einen bestimmten Bereich des aktuellen Prozesses innerhalb des aktiven Fensters vergrößern lassen. Dazu klicken Sie wieder das »I«-Gadget an, bleiben diesmal aber mit dem Mauszeiger im aktuellen Window. Die Markierung des zu vergrößernden Bereichs erfolgt wie bei Möglichkeit 1. Damit die Vergrößerung berechnet wird, klicken Sie das »G«

Gadget an (»G« wie »Generate« oder »Generieren«). Der vom Zoom-Rahmen umschlossene Bereich wird dann formatfüllend im Prozeßfenster abgebildet. Wenn Sie »G« anklicken und damit den Berechnungsvorgang starten, werden Sie sehen, daß sich das »G«-Gadget in ein »S«-Gadget verwandelt hat. »S« steht für »Stop« und unterbricht die Berechnung. Nach einer solchen Unterbrechung läßt sich die Berechnung nur

sion des Rahmens zusammengestaucht wird. Natürlich muß auch bei »Zoom-Out« der Berechnungsvorgang durch Anklicken von »G« gestartet werden. Sollten Sie es sich nach dem Markieren anders überlegt haben und kein Zooming wünschen, so klicken Sie in die linke obere Ecke des Markierungsrahmens. Wie bei einem normalen Window sitzt dort ein »Close«-Gadget, das den Rahmen schließt und damit die

Gadget im Julia-Prozeßfenster an. Der Mauszeiger erhält wieder das Anhängsel »To«. Klicken Sie nun in den gewünschten Bereich der Mandelbrot-Menge, auf die sich der Julia-Prozeß bezieht, und starten Sie Berechnung und Darstellung mit einem Klick auf das »G«-Gadget.

Current«

Wie das »C«-Gadget eines Projekt-Window macht »Current« einen Prozeß zum aktuellen Prozeß. Besonders wichtig wird dieser Menüpunkt bei rahmenlosen (borderless) Windows. Um mit »Current« einen Prozeß zu aktivieren, muß zuerst der Mauszeiger über dem gewünschten (evtl. rahmenlosen) Projekt positioniert werden. Dann drücken Sie die rechte Maustaste, halten sie gedrückt und wählen mit dem Mauszeiger den Menüpunkt »Current« an. Lassen Sie die rechte Maustaste los, während »Current« selektiert ist. Der gewünschte Prozeß wurde zum aktuellen Prozeß erhoben (erkennlich am Sternchen in der Window-Kopfleiste).

Load und Save

Wenn hier von Laden oder Speichern die Rede ist, sind nicht nur Grafiken gemeint, sondern es werden auch die jeweiligen Basisparameter »mitgenommen«. Das geht so weit, daß Berechnungen und Grafi-

Basis sichern

ken, die mittels »S« gestoppt wurden, abgespeichert und später wieder in das RAM geladen werden können und die Berechnungen durch Anwählen des Menüpunkts »Calculate-Generate-Resume« an der Abbruchposition fortgesetzt werden. Wenn Sie »Load« oder »Save« befehlen, öffnet das Programm einen komfortablen File-Requester.

Close

wirkt wie das Close-Gadget eines Windows, bezieht sich aber immer auf das aktuelle Projektfenster. Wollen Sie rahmenlose Projekte beenden, müssen Sie diese erst mit »Current« aktualisieren, ehe Sie sie mit »Close« schließen können.

Save ILBM

Mit diesem Punkt sichern Sie den gesamten Bildschirminhalt als IFF-

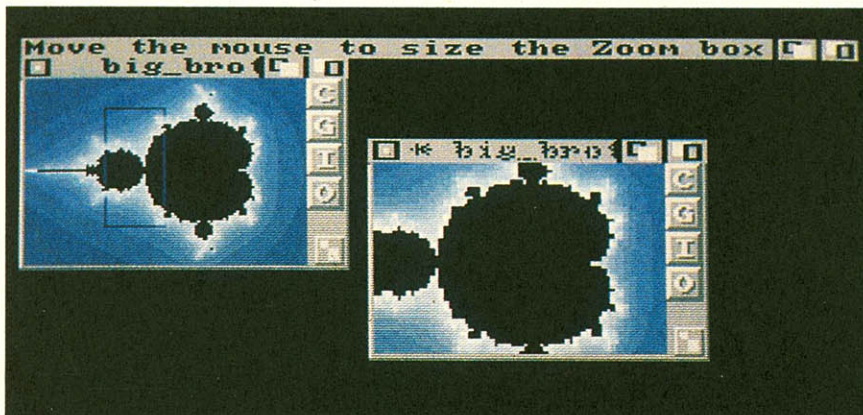


Bild 2. »Mandelvroom« – die Zoom-Funktion mit »Magnifier«

durch einen Punkt im Hauptmenü fortsetzen. Vom Projektfenster aus können Berechnungen nur gestartet und gestoppt werden. Ein nach einer Unterbrechung angeklicktes »G« startet die Berechnung bei Null.

Möglichkeit 3: Klicken Sie das Gadget »O« an. Der Inhalt des Prozeßfensters wird von einem Rahmen umspannt, dessen Lage und Dimensionen Sie mit der Maus beeinflussen können. In der rechten unteren Ecke des Begrenzungsrahmens ist ein Gadget angedeutet. Wenn Sie diesen Bereich mit dem Mauszeiger anfahren und durch Drücken der linken Maustaste »einfangen«, können Sie die rechte untere Ecke des Zoom-Bereichs verschieben. Die linke obere Ecke des Rahmens bleibt dabei fixiert. Um das Zentrum des Prozeßfensters mehr oder weniger stark »wegzurücken«, benutzen Sie bitte das Pseudo-Gadget im Zoom-Rahmen links unten. Wenn Sie diesen Bereich mit Mauszeiger und linker Maustaste aktivieren, folgen alle vier Ecken des Zoom-Rahmens den Mausbewegungen. Im Gegensatz zu »Zoom-In« symbolisiert der Rahmen bei »Zoom-Out«, daß der Inhalt des Prozeßfensters auf die Dimen-

Zoom-Funktion abbricht. Mit dem »C«-Gadget machen Sie den dazugehörigen Prozeß zum aktuellen Prozeß (Markierung mit Sternchen). Ganz rechts unten sitzt das übliche Size-Gadget. Sie sollten allerdings nicht annehmen, daß Sie damit sofort eine vergrößerte Abbildung des Fensterinhalts bekämen. Jede Vergrößerung der Projekt-Window-Dimensionen bedarf einer Neuberechnung (»G«-Gadget).

Julia

»New/Julia« öffnet wie »Mand« ein eigenes Fenster und bezieht sich auf die Grunddaten des aktuellen Prozesses. Da sich die Julia-Menge aber auf einen Bereich innerhalb der Mandelbrot-Menge bezieht, muß dem Programm ein Punkt innerhalb des korrespondierenden Apfelmännchens bekanntgegeben werden. Dazu klicken Sie das »J«-

ILBM-File. Auch hier steht Ihnen der von »Load« und »Save« bekannte File-Requester hilfreich zur Seite.

Help

Zu jedem Menüpunkt, zu jedem Gadget gibt es ein eigenes Help-File. Zuerst wählen Sie aus dem Hauptmenü »Help« an. Der Mauszeiger erhält das Anhängsel »With«. Jetzt klicken Sie einfach das Gadget an, zu dem Sie Information wünschen, bzw. wählen in der Menüleiste jenen Punkt an, dessen Funktion Ihnen unklar ist. Dieses Anwählen geht genau so vor sich, als wollten Sie den Menüpunkt aktivieren. Eine kleine Macke hat die Hilfefunktion allerdings: Als Relikt der NTSC-Programmierung öffnet der Ausgabeteil des »Help«-Programms nur ein kleines, NTSC-gemäßes Window, in dem der Hilfstext ausgegeben wird. Wenn Sie dieses Window nicht durch das Größen-Gadget auf die volle PAL-Größe bringen, gehen Ihnen bei längeren Hilfstexten die untersten Zeilen verloren. Der Rollbalken auf der linken Seite rechnet richtig im PAL-Format und »glaubt« beim Umblättern auf die nächste Seite, daß die »unteren« Zeilen auf dem vorhergehenden Bildschirm ausgegeben worden seien.

Cancel

Diese Funktion bricht die aktuelle Parametereingabe ab. Wenn Sie z.B. das »Zoom-In«-Gadget angeklickt haben, wartet das Programm auf Ihre Definition des zu vergrößernden Bereichs. Um aus dieser Definition aussteigen zu können, wählen Sie einfach »Cancel« an. Dies hat einen ordnungsgemäßen Abbruch der gewählten Funktion zur Folge.

Quit

ist allen bekannt. Eine Sicherheitsabfrage »Really quit ?« gibt Ihnen eine letzte Chance - dann sind Sie draußen, und »Mandelvroom« räumt das Feld und den Speicher.

»About« gibt die übliche Copyright-Meldung aus.

Display

In diesem zweiten wichtigen Hauptmenüpunkt werden alle Einstellungen vorgenommen, die mit der Ausgabe auf dem Bildschirm zu tun haben.

Colors

Nach dem Anwählen dieses Menüpunkts öffnet »Mandelvroom« ein kleines Fenster, in dem sich für jede Farbe der R-G-B-Anteil einstellen läßt (Bild 3). Die einzelnen Farben des aktuellen Projekts werden durch kleine Gadgets dargestellt. Klicken Sie das Gadget jener Farbe an, die Sie zu ändern wünschen und ver-

»Color«-Windows ersetzen Sie beispielsweise Farbe A durch Farbe B, so daß Farbe B zweimal vertreten ist. Klicken Sie zuerst Farbgadget B an, klicken Sie danach auf das »Copy«-Gadget und dann auf das Farbgadget A - die Farbe A ist jetzt identisch mit Farbe B. Der gleiche Ablauf bei Verwendung des »Exchg«-Gadgets (austauschen) wechselt Farbe A mit Farbe B, ohne den Farbwert von A oder B zu beeinflussen. Blicke noch das »Spread-Gadget«. Diese Funktion erleichtert die Einstellung von Farbverläufen (to spread - verteilen). Um z.B. einen gleichmäßig abgestuften Farbverlauf zwischen Far-

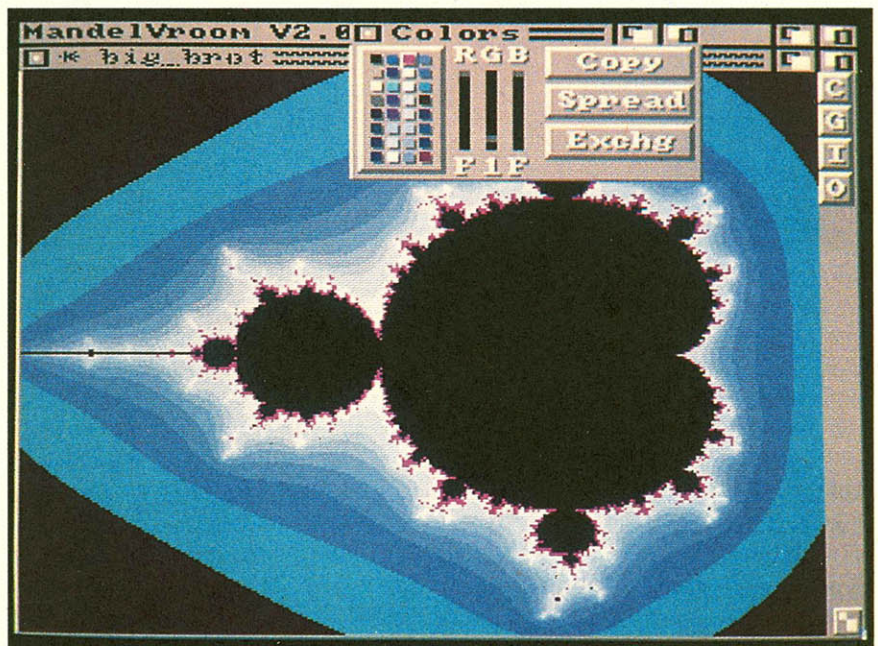


Bild 3. Mit dem »Colors«-Menü haben Sie die Farben Ihrer Mandelbrötchen im Griff

schieben Sie die Rot-, Grün- und Blau-Regler nach Belieben. Beachten Sie bitte, daß alle Farben des Bildschirms in diesen Gadgets repräsentiert sind - also auch Border, Schrift usw. Jede im »Color«-Window angewählte Farbe, die auf dem Bildschirm vertreten ist, wird analog zur Stellung der Schieberegler mit geändert. Sie sehen sofort, wie sich Ihre Manipulation an den Schieberegler auf die Gesamtgrafik auswirkt. Die Zahlen und Buchstaben unter den Schieberegler stellen die 16 Farbstufen je Grundfarbe dar. Der Einfachheit halber hat Kevin Clague die Numerierung in »Hex« vorgenommen. Mit der »Copy«-Funktion des

be A in Farbgadget 2 und Farbe B in Farbgadget 8 zu erhalten, klicken Sie zuerst Farbgadget 2 an, danach das »Exchg«-Gadget und anschließend Farbgadget 8. Bei all diesen Farbspielereien sollten Sie vorsichtig sein. Da alle Farbmanipulationen sich nicht nur auf die Grafik, sondern auch auf die Menüleiste auswirken, kann es schon mal passieren, daß Sie die Farbgadgets und Schieberegler nicht mehr sehen können. Und dann hilft eigentlich nur noch ein Neustart des Programms.

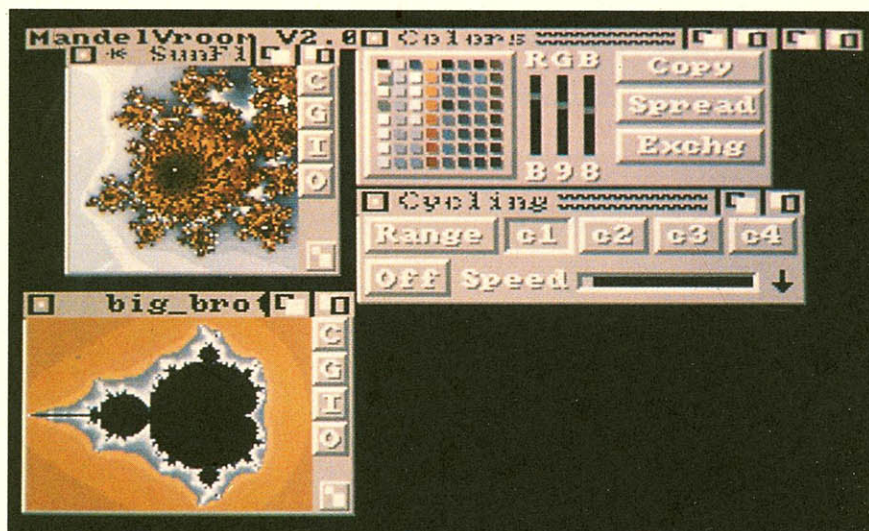


Bild 4. »Color-Cycling« de Luxe mit »Mandelvroom«

Cycling

Color-Cycling ist Ihnen sicherlich ein Begriff. Mit »Cycling« bringen Sie diese Pseudo-Animation in Ihre Apfelmännchen. »Cycling« allein ist allerdings zu wenig — Sie brauchen zeitgleich auch die »Color«-Palette. Das hat folgenden Grund: Sie sehen in Bild 4 nicht nur zwei »Mandelbrötchen« und das »Color«-Window, sondern auch das Kontrollfeld für »Cycling«. Die Gadgets »c1« bis »c4« stehen für vier unabhängige Cycling-Bereichsdefinitionen. Um einen solchen Bereich zu definieren, brauchen Sie das »Color«-Kontrollfenster. Ein Cycling-Bereich wird folgendermaßen definiert: Zuerst wählen Sie die Nummer der Bereichsdefinition, z. B. »c2«. Dann klicken Sie im »Color«-Window die Startfarbe an. Jetzt ein Klick auf »Range« — der Mauszeiger bekommt wieder sein »To«-Anhängsel. Nun muß noch der Endbereich definiert werden. Wieder ein Klick auf ein Farbgadget im »Color«-Fenster. So können Sie vier Bereiche angeben, die aber alle gleichzeitig aktiv sind. Abschalten kann man einen solchen Bereich nur, indem man dasselbe Farbgadget als Start- und als Endfarbe definiert. Die Geschwindigkeit, mit der die einzelnen Cycling-Bereiche durchlaufen werden, stellen Sie mit dem Schieberegler »Speed« ein. Auch diese Einstellung kann für jede Bereichsdefinition separat vorgenommen werden. Ob sich der Farblauf von Farbe A nach Farbe B vollzieht oder umgekehrt, bestimmen Sie mit dem

schwarzen Pfeilchen neben dem »Speed«-Regler — anklicken ändert die Richtung. Gestartet wird das Color-Cycling durch einen Klick auf das »On«-Gadget oder durch das Anklicken des »Speed«-Balkens. Wenn Sie dem »Speed«-Balken den Vorzug geben, läuft das Cycling solange, wie Sie die linke Maustaste drücken, bei Verwendung des »On«-Gadgets bis zum Anklicken des »Off«-Gadgets oder durch Anwahl eines der vier Bereichsgadgets.

Contours

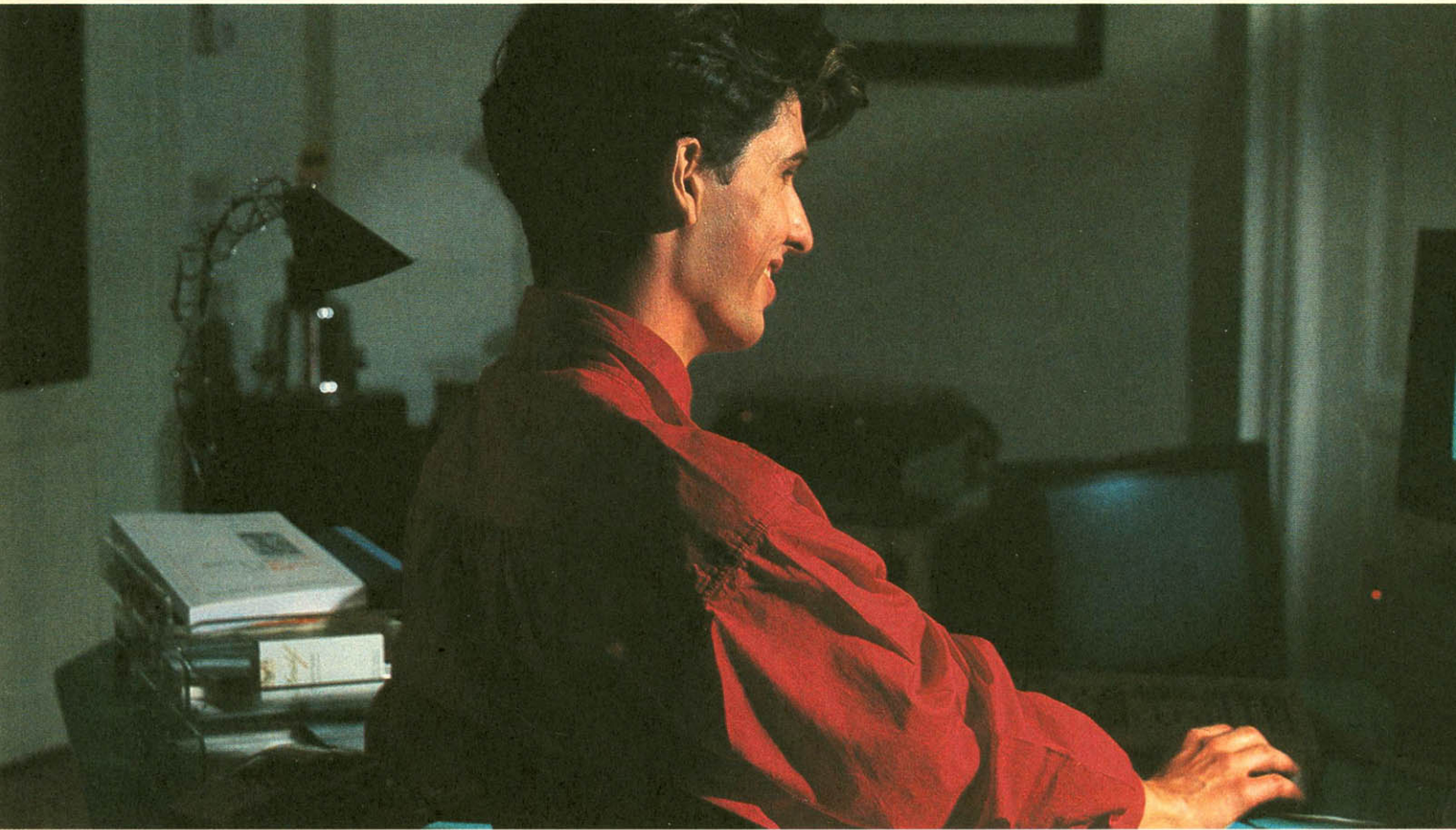
Mit dem »Contours«-Menü machen Sie mit Ihren Mandelbrot-Grafiken, was Kartographen mit ihren Landkarten anstellen: Sie ordnen dem Höhenrelief Ihrer Apfelmännchen Farben zu, die die Z-Achse repräsentieren (auf der Z-Achse wird die Anzahl der durchgeführten Iterationen aufgetragen). Die »Höhe« eines Punktes wird vom Programm in der Titelleiste angezeigt, wenn Sie den gewünschten Punkt mit dem Mauszeiger anfahren und die linke Maustaste drücken. Zusätzlich erhalten Sie die Lokation des Punktes innerhalb der Menge durch die Angabe des Real- (»r«) und des Imaginäranteils (»i«). Im Gegensatz zur

Farbpalette, bei der sich Änderungen auf den gesamten Bildschirm auswirken, beschränkt sich der Wirkungsbereich von »Contours« auf das aktuelle Projekt. Wenn Sie »Contours« anwählen, öffnet sich ein Window, wie Sie es in Bild 5 sehen. Zuerst einmal fallen die equalizerähnlichen Schieberegler auf. Sie stellen den »Wasserspiegel« dar, ab dem eine Farbe zur Repräsentation eines gewissen Niveaubereichs herangezogen wird. Wenn Sie beispielsweise den Schieberegler ganz links nach unten ziehen, ist das, als wenn Sie den Stöpsel aus der Badewanne ziehen. Zuerst waren nur Kopf und Schultern sichtbar, jetzt sieht man auch noch die Knie. Diese Einstellung läßt sich für jede verwendete Farbe (=Höhenbereich) durchführen. Um die neue Einstellung in die Grafik zu übertragen, ist es notwendig, das Gadget »Paint« im Contours-Window anzuklicken. Für alle, denen das etwas zu schnell und zu oberflächlich war, noch eine kleine Ergänzung: Schieberegler, die einen Höhen- oder Iterationsbereich definieren, der von einem anderen Schieberegler mit höherer Priorität bereits abgedeckt wird, werden mit dunklen »Schiebeknöpfen« dargestellt. Die Prioritätsreihenfolge ist von links nach rechts fallend. Das heißt also, daß die Einstellung des Schie-

Die dritte Dimension

bereglers 3 für alle Regler mit Ordnungszahl größer drei das Limit darstellt. Aktive Schieberegler, deren repräsentierter Iterationsbereich nicht von anderen Reglern höherer Priorität beansprucht wird, haben helle Schiebeknöpfe. Am deutlichsten wird das wohl, wenn Sie den Regler 1 (ganz links außen) nach unten stellen und danach »Paint« anklicken. Damit ergibt sich Iterationstiefe »Null« für alle Farben — das Bild wird monochrom, und alle anderen Regler sind inaktiv (dunkler Schiebeknopf). Welche Farbe welchen Höhenbereich darstellt, entnehmen Sie der Kopfleiste des »Contours«-Windows. Dort sehen Sie drei Buchstaben: »C«, »P« und »H«. »C« steht für »Current Contour«, für die Reglernummer, die ak-

Ich schau Dir in den



W. Häring
**Schnellübersicht
 Amiga-DOS 1.3**
 Alles schnell im Griff:
 Grundlagen und Peripherie, Massenspeicher, Verzeichnisse und Laufwerke, Dateien und ihre Sicherung, Ein- und Ausgabe – sowie Antworten auf alle Fragen, die bei der täglichen Arbeit auftreten.
 1989, 292 Seiten,
 ISBN 3-89090-730-X
DM 39,-



P. Wollschlaeger
**Schnellübersicht
 Amiga-Basic**
 Die Beschreibungen sind problemorientiert aufgebaut, und die Informatikern werden so vermittelt, wie sie bei der täglichen Arbeit auftreten. Eine ausklappbare Themenübersicht und Querverweise erleichtern die Arbeit zusätzlich.
 1989, 336 Seiten,
 ISBN 3-89090-736-9
DM 39,-



T. Kaltenbach/H. Woerrlein
**Schnellübersicht
 GFA-Basic 3.0**
 Das rasante Nachschlagewerk für den schnellen GFA-Interpreter. In problemorientierter Aufbereitung erhalten Sie eine detaillierte Übersicht aller Befehle. An zahlreichen Beispielen werden Anwendung und Einsatz erläutert.
 1989, 431 Seiten,
 ISBN 3-89090-101-8
DM 39,-



■ **NEU**
 A. Grote
**Desktop Video
 auf dem Amiga**
 Die Verbindung von Computer und Video wird auch für Amateure immer reizvoller und – einfacher. Hier kommt die grundlegende Übersicht mit genauer Beschreibung der vielfältigen Hardware-Erweiterungen und Software-Programme für Videofilmer. Auf der beiliegenden Diskette sind Programme, um interessante Effekte selbst zu gestalten.
 1990, 192 Seiten,
 inkl. 2 Disketten
 ISBN 3-89090-312-6
DM 59,-



■ **NEU**
 M. Breuer
Amiga-2000-Buch
 Vollständig überarbeitete und aktualisierte Neuauflage dieser fundierten Einführung in die Bedienung des Amiga 2000 und 2500. Mit leichtverständlicher Beschreibung der Hardware und der verschiedenen Ausbaumöglichkeiten. Dazu ausführliche Darstellung von Workbench 1.3, CLI und der neuen Shell.
 2., überarb. Auflage 1990,
 672 Seiten
 ISBN 3-89090-287-1
DM 59,-

Bildschirm, Kleines!



■ **NEU**

G. Glaeser/T. Grohser
Amiga 3-D-Sprinter
 Profi-Software zum Buchpreis: Das Programm ergänzt den Amiga in seinen Grafikmöglichkeiten und reizt sie vollständig aus. Grafiken, Schatten, Animationen und Spiegelungen werden je nach Rechnerkonfiguration in Echtzeit berechnet. Die Bedienung ist so einfach und effizient, daß Sie in kurzer Zeit komplizierteste Objekte erzeugen können. 1990, ca. 250 Seiten, inkl. 2 Disketten
ISBN 3-89090-109-3
DM 98,-*

■ **NEU**

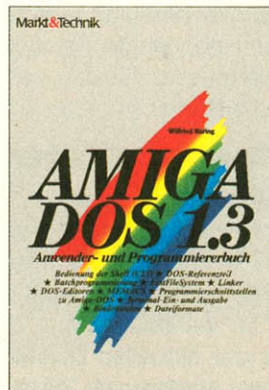
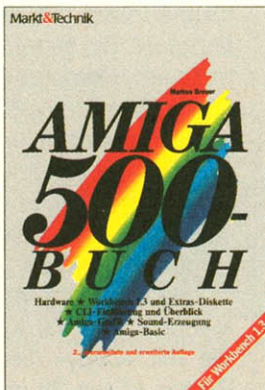
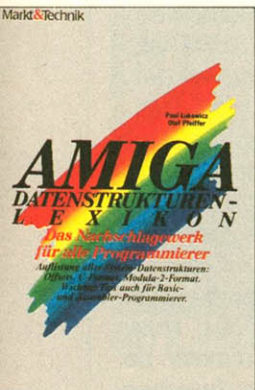
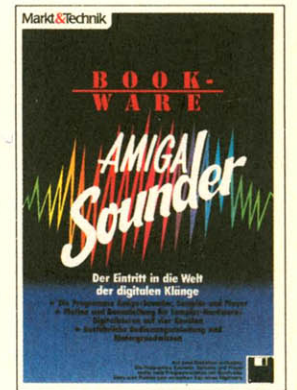
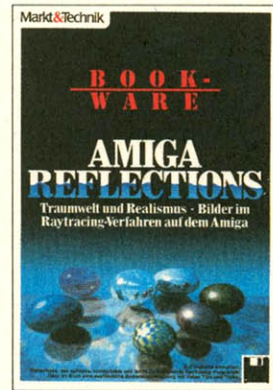
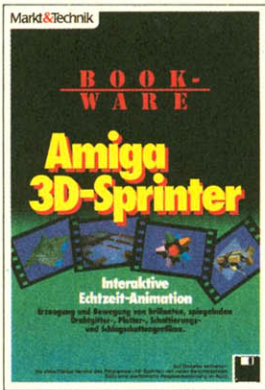
Atlantis
AMIGA TRICKSTUDIO A, Version 2
 Profi-Software zum Buchpreis: die Traumfabrik für den Amiga-Anwender. Vom Stummfilm-Slapstick bis zum Werbespot - Trickstudio A unterstützt Sie beim Aufbau des Films, bei der Umsetzung von Einzelbildern in Abläufe und hilft Ihnen, Bild und Ton punktgenau zu synchronisieren. 1990, 86 Seiten, inkl. Diskette
ISBN 3-89090-886-1
DM 99,-*

C. Fuchs

AMIGA REFLECTIONS
 Profi-Software zum Buchpreis: Jetzt produzieren Sie auf Ihrem Amiga Bilder für Ihre private Diashow oder den Vorspann für Ihren Videofilm. Reflections erzeugt IFF-Grafiken und unterstützt den HAM-Modus. Das Begleitbuch erklärt, wie Raytracing funktioniert, und liefert zahlreiche Tips und Tricks für den Anwender. 1989, 156 Seiten, inkl. Diskette
ISBN 3-89090-727-X
DM 98,-*

H. Knappe

Amiga Sounder
 Profi-Software zum Buchpreis: ein Komplettpaket für den Einstieg in die Welt der digitalen Klänge. Zur Software gibt es eine Platine, die mit wenig Lötlötarbeit zu einem 4-Kanal-Digitizer ausgebaut werden kann. Die Beschreibung ist so ausführlich, daß auch Anfänger kaum etwas falsch machen können. 1989, 336 Seiten, inkl. 2 Disketten und Platine
ISBN 3-89090-709-1
DM 98,-*



* unverbindliche Preisempfehlung

■ **NEU**

P. Lukowitz/O. Pfeiffer
Amiga Datenstrukturen-Lexikon
 Alle Systemdatenstrukturen werden unter Angabe des Offsets aufgelistet und ausführlich beschrieben. Zu den vier wichtigsten Programmiersprachen - C und MODULA-2, Basic und Assembler - finden Sie detaillierte Benutzeranweisungen und Anwendungsbeispiele. Referenzlisten der Systemroutinen machen die Vorgänge endgültig transparent. 1990, ca. 250 Seiten, ISBN 3-89090-250-2
DM 69,-

M. Breuer

Amiga-500-Buch
 Eines der erfolgreichsten Commodore-Bücher in aktueller Überarbeitung. Alles über Hardware, Software, Zubehör und eine ausführliche Beschreibung der Workbench 1.3. Durch viele Abbildungen und Beispiele werden Sie mit der Bedienung des Amiga 500 schnell vertraut. In einem übersichtlichen Nachschlageteil werden die Shell-Befehle erläutert. 2., überarb. Auflage 1989, 541 Seiten
ISBN 3-89090-300-2
DM 49,-

W. Häring

Amiga-DOS 1.3
 Im ersten Teil werden dem Anwender die Shell- und die Amiga-DOS-Befehle, Startup-Sequenzen und die Arbeit mit dem Editor erklärt. In einem speziellen Anhang finden Sie viele nützliche Tips zur Konfiguration Ihres Druckers. Der zweite Teil wendet sich an künftige Programmierer und macht sie mit dem Betriebssystem und der Hardware vertraut. 1989, 392 Seiten, ISBN 3-89090-802-0
DM 69,-

Markt&Technik-Bücher und -Software erhalten Sie bei Ihrem Buchhändler, in Computerfachgeschäften und in den Fachabteilungen der Warenhäuser.



tuell zur Bearbeitung ansteht. »P« steht für »Pen-Number« und ist die Ordnungszahl jener Farbe, die dem momentan aktuellen Höhenbereich zugeordnet wurde (die Farbnummer ergibt sich aus der Reihenfolge der Farben im »Colors«-Menü). Bleibt noch »H« für »Height«, also »Höhe«. Hier wird die zugeordnete Iterations-tiefe angezeigt, die entweder nur eine Stufe groß ist oder mehrere Iterationsstufen »weit« sein kann – je nach Reglerstellung. Wenn Sie jetzt versuchen, mit den Schieberegler exakte Werte einzustellen, werden Sie schnell feststellen, daß das sehr schwierig ist. Die Auflösung macht einem schon ganz schön zu schaffen, und das ewige »Paint« anzuklicken nervt ein wenig. Besser und bequemer geht es mit dem »Set«-Gadget. Um einem definierten Bereich Ihrer Mandelbrot-Menge eine bestimmte Farbe zuzuordnen, gehen Sie bitte folgendermaßen vor: Zuerst klicken Sie im »Contours«-Window den gewünschten Regler an. Nach einem Klick auf das »Set«-Gadget erhält der Mauspointer den Zusatz »To«. Wenn Sie nun in Ihrer Mandelbrot-Menge einen Punkt anklicken, wird der aktive Schieberegler auf die korrespondierende Höhe gesetzt. Öffnen Sie das »Colors«-Window. Klicken Sie im »Contours«-Window auf »Set« und im »Colors«-Window auf die gewünschte neue Farbe. Der aktive Schieberegler hat nun eine neue Farbdefinition erhalten, die nach Anklicken von »Paint« in die aktuelle Grafik übertragen wird. Das nächste interessante Gadget ist jenes mit der Bezeichnung »Smooth«. Damit errechnet »Mandelvroom« einen sanft abgestuften Farbverlauf zwischen zwei Höhenbereichen. Nehmen wir an, Sie wollen zwischen den Höhenreglern 3 und 11 und damit zwischen den zusammenhängenden Iterationsstufen eine sanfte Farbtreppe aufbauen. Dazu aktivieren Sie zunächst Regler 3 und geben dann mit dem »Set«-Kommando wie oben beschrieben

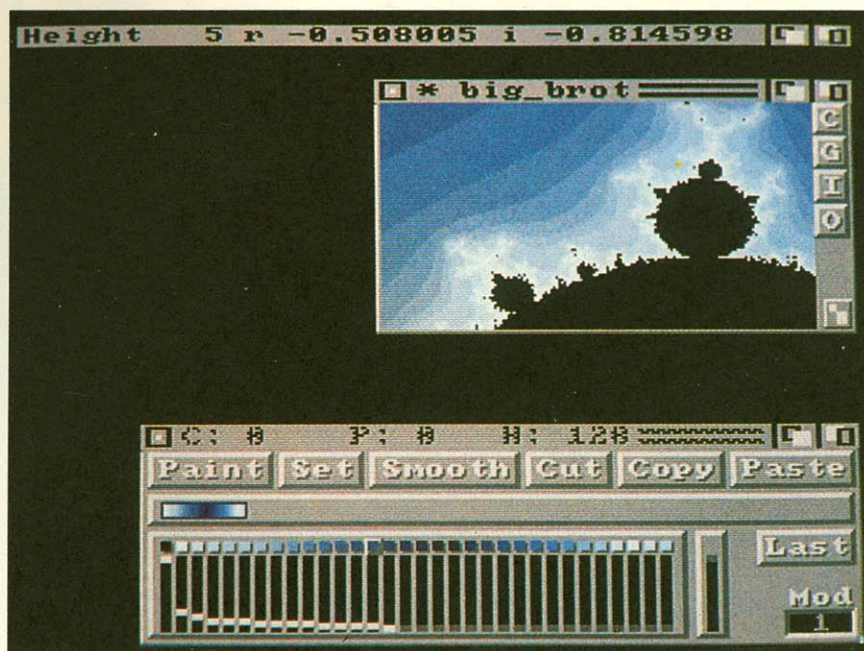


Bild 5. Farben als dritte Dimension – das »Contours«-Menü

eine Iterationsstufe als Anfangsbereich ein. Dann schalten Sie Regler 11 auf »aktiv« und übermitteln durch »Set« und Anklicken des Bereichs im Mandelbrötchen die gewünschte zweite »Höhe«. Klicken Sie »Smooth« und danach Regler 3 an. »Mandelvroom« errechnet anhand der Grenzwerte eine mehr oder weniger zart abgestufte, gleichmäßig zwischen den Grenzen abgestufte Iterationsverteilung. Vergessen Sie dabei aber nicht, daß die Regler Prioritäten ha-

Wichtige Prioritäten

ben und Sie daher für Regler 3 eine höhere Iterationsstufe wählen müssen als für Regler 11. Mit »Cut« trennen Sie einen beliebigen Teil der Farbregler im wahrsten Sinne des Wortes aus der aktuellen Definition heraus. Dieser abgetrennte Teil ist allerdings nicht verloren, sondern wird für die Laufzeit des Programms zwischengespeichert und kann mit »Paste« an beliebiger Stelle im »Contours«-Menü wieder eingefügt werden. Dieses Einfügen geht so vor sich, daß Sie zuerst »Paste« und dann den Regler mit der niedrigsten Priorität anklicken, von dem an aufsteigend die gespeicherten Einstellungen eingetragen werden. Ähnlich wie »Cut« funktioniert »Copy«. Wie aus der Namensgebung schon ableitbar ist, wird bei »Copy« der markierte Teil nicht aus der aktuellen De-

finition herausgetrennt, sondern nur in den Zwischenspeicher geschrieben, um bei Bedarf an beliebiger Stelle mit dem »Paste«-Kommando zusätzlich eingetragen zu werden. Ein auf den ersten Blick etwas mysteriöses Gadget ist »Last« – man klickt es an, und es tut sich (manchmal) nichts. Das kommt daher, daß »Last« für die letzte, die 256. Höhendefinition steht. Sie sehen nur 32 Schieberegler im »Contours«-Window? Dann schnappen Sie sich mit dem Mauszeiger und der linken Maustaste mal den kleinen Rahmen, der oberhalb der höchstwertigen Schieberegler steht, und ziehen Sie ihn nach links. Insgesamt sind es 256 Schieberegler, die Ihnen zur Verfügung stehen. Machen Sie mal folgendes: Holen Sie mit »Copy« die Parameter der Regler 12 bis 23 in den Zwischenspeicher. Aktualisieren Sie Regler 24 durch Anklicken. Wählen Sie das Gadget »Paste« an. Wenn der Mauszeiger den Zusatz »To« trägt, klicken Sie keinen Reglerknopf an, sondern das Gadget »Last«. Jetzt sind alle Regler von 24 bis 256 mit einer sich wiederholenden Sequenz der mit »Copy« selektierten Parameter ver-

sehen. Soweit zu »Last«. Jetzt bleibt nur noch eine Funktion: »Mod«. »Mod« steht für »Modulo« und ist eine Zusatzfunktion zu »Paste«. Die Wirkung läßt sich wieder an einem Beispiel am besten erklären: Angenommen, Sie haben 27 Reglerparameter mit dem »Copy«-Kommando in den temporären Speicher geholt. Wenn Sie jetzt im »Mod«-Gadget die Zahl »7« eintragen und dann ein »Paste«-Kommando ausführen, werden nur jene Parameter ein- bzw. angefügt, die an 7., 14. und 21. Position im Zwischenspeicher stehen. Der Abstand zwischen den an- bzw. eingefügten Parametern entspricht der Position innerhalb der Original-Sequenz.

AutoContour

Diese Funktion beeinflußt direkt die Einstellungen der Schieberegler

Herstellungen erfolgt so, daß die zur Verfügung stehenden Abstufungen möglichst gleichmäßig verteilt sind, wobei als »Verteilschlüssel« die Anzahl der Bildpunkte je Niveau herangezogen wird.

Histogram

Im »Histogram« können Sie sehen, wie oft ein Ereignis mit einer bestimmten Ordnungszahl aufgetreten ist. In unserem Fall ist das »Ereignis« das Auftreten eines Pixels, und mit Ordnungszahl ist hier der korrespondierende Wert der Z-Achse gemeint. Im »Mandelvroom-Histogram« (Bild 6) sehen Sie, welche Höhe wie oft aufgetreten ist.

Borderless

Durch Anwahl dieses Menüpunkts, der eigentlich nur ein Ein-Aus-Schalter ist, wird der Rahmen

zahl bezieht sich immer nur auf das aktuelle Projekt. Haben Sie jedoch mehrere Projekte gleichzeitig laufen, kann zwar für jedes Projekt die Farbzahl separat eingegeben werden, es ist jedoch auf dem ganzen Schirm immer die Farbzahl gültig, die für das aktuelle Projekt gewählt wurde. Haben Sie beispielsweise Projekt »A« mit acht Farben definiert und

Die Farbzahl

Projekt »B« mit zwei Farben, so wird Projekt »A« mit nur zwei Farben dargestellt, wenn Sie Projekt »B« durch Anklicken des »C«-Gadgets aktualisieren. Umgekehrt wird Projekt »B« mit acht Farben dargestellt, wenn Projekt »A« aktuell ist. Wählen können Sie zwischen 2, 4, 8, 16, 32 und 64 Farben. Bei genug freiem Speicher haben Sie die freie Farbwahl. Lediglich die 64-Farben-Darstellung ist davon abhängig, welchen »View Mode« Sie gewählt haben.

View Modes

Sie haben die Wahl zwischen »High Resolution« (640 Pixel horizontal), »Low Resolution« (320 Pixel horizontal), »Interlace« (512 Pixel vertikal), »Non-Interlace« (256 Pixel vertikal) und allen zwischen den genannten möglichen Kombinationen. Überdies ist der Half-Brite-Modus anwählbar, der die Anzahl der darstellbaren Farben auf 64 erhöht (siehe »Num Colors«). Der aktivierte Mode wird in bester Amiga-Intuition-Manier durch ein Häkchen gekennzeichnet (Bild 7). Eine nochmalige Anwahl schaltet den jetzt aktiven Mode wieder aus. Hier scheint übrigens eine Warnung angebracht: Der Multitasking-Betrieb bereitete im Test einige Schwierigkeiten. Dabei geschah es mehrmals, daß die gleichzeitige Anwahl von »Interlace«, »Hires« und »Half-Brite« von »Mandelvroom« akzeptiert wurde, ein Abschalten dieser Funktionen aber ebenso nicht mehr möglich war wie die Aktualisierung eines anderen Projekts oder auch nur das Schließen eines Projekts durch Anklicken des »Close«-Gadgets oder des äquivalenten Menüpunkts.

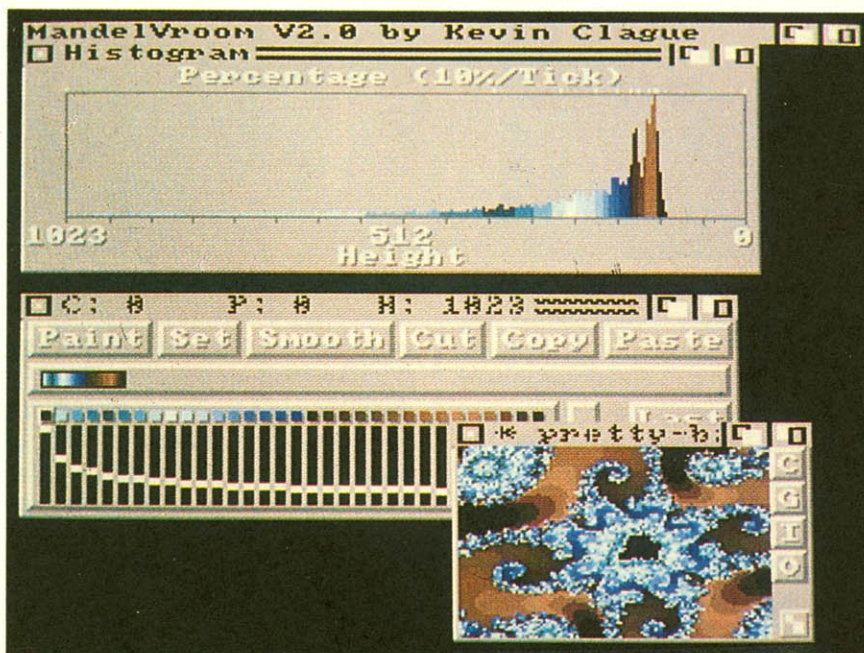


Bild 6. »Histogram« — die Statistik gibt Auskunft

im »Contours«-Menü. Passen Sie also auf, wenn Sie sich mühsam eine tolle Parameter-Zusammenstellung gebastelt haben. »AutoContour« hat zwei Stufen (»I« und »II«). Die Funktionsweise läßt sich folgendermaßen beschreiben: Auswahl, Anzahl und Reihenfolge der Farben bleibt Ihnen überlassen, da richtet sich »AutoContour« ganz nach Ihrem Geschmack. Die Stellung der Regler hingegen, also die Zuordnung des darzustellenden Höhenbereichs, behält sich »AutoContours« vor. Die Berechnung der verschiedenen Reg-

mit den Gadgets um das aktuelle Projekt ein- und ausgeschaltet.

Num Colors

Hier teilen Sie »Mandelvroom« mit, aus wie vielen Farben sich Ihr Projekt maximal zusammensetzt. Die Anzahl wird einfach durch Anfahren des entsprechenden Submenüs teils ausgewählt. Die gewählte Farb-

Screen Size

Dieser Punkt unterteilt sich in die anwählbaren Modi *Standard* und *WorkBench*. Ist »Standard« angewählt, orientiert sich »Mandelvroom« an den von Commodore vorgegebenen Bildschirmdimensionen. Bei »WorkBench« wird die Auflösung des Workbench-Screens als Maßstab genommen. Dies mag auf den ersten Blick witzlos erscheinen, doch bedenkt man, daß es Programme wie »MoreRows« gibt, die die Auflösung der Workbench auf mehr als 700 Pixel horizontal steigern, wird der Sinn dieses Menüpunkts klar.

Calculate

Der erste Unterpunkt ist **Generate**, eng verwandt mit dem Gadget »G« an jedem Projektrahmen. »Generate« unterteilt sich in *Start*, mit dem Sie einen Berechnungsvorgang starten, *Stop*, bei dem eine Berechnung unterbrochen wird, und in *Resume*, mit dem ein unterbrochener Rechenprozeß fortgesetzt wird. »Start« und »Stop« wirken wie das »G/S«-Gadget am Projektrahmen, »Resume« steht Ihnen nur im »Calculate«-Menü zur Verfügung.

Zoom

Die Zoom-Funktionen von »Mandelvroom« wurden schon ausführlich besprochen – dieser Menüpunkt ist eine Alternative zu den Gadgets am Projektrahmen inklusive »Close«-Gadget und »Julia« bei einem »Julia-Project«.

Pan

In »Mandelvroom« steht »Pan« für »Panorama«. Angenommen, Sie haben ein Projekt fertig und wollen wissen, wie es denn links oben jenseits des Projektrahmens weitergeht. Mit »Zoom-Out« könnten Sie das zwar erfahren, aber eine Verkleinerung paßt Ihnen jetzt nicht in den Kram. Also wählen Sie *Start*. Plazieren Sie den Maus-Zeiger irgendwo innerhalb des Projektrahmens und drücken die linke Maustaste. Der gesamte Rahmeninhalt läßt sich verschieben. Sie wollten wissen, wie es links oben weitergeht: Also machen

Sie Platz und schieben das »ausgeschnittene« Bildchen nach rechts unten. Jetzt wählen Sie *Generate* und schon wird der bisher freie Bereich innerhalb des Projektrahmens mit dem Rest der Mandelbrot-Grafik gefüllt. Falls Ihnen die Neuberechnung zu lange dauert, läßt sie sich mit *Cancel* abbrechen.

Magnifier

Dieser Punkt ist eng mit der Zoom-Funktion verbunden. Sie erinnern

FFP

Deutlich genauer, aber leider ebenso deutlich langsamer werden die Berechnungen, wenn Sie den Prozessor in die Abgründe der Fließkomma-Arithmetik jagen (*Fast Floating Point*).

IEEE

Das höchste der Gefühle für den »Normalsterblichen« ist die Verwendung der IEEE-Routinen. Über den Zeitbedarf möchten wir hier nur eine

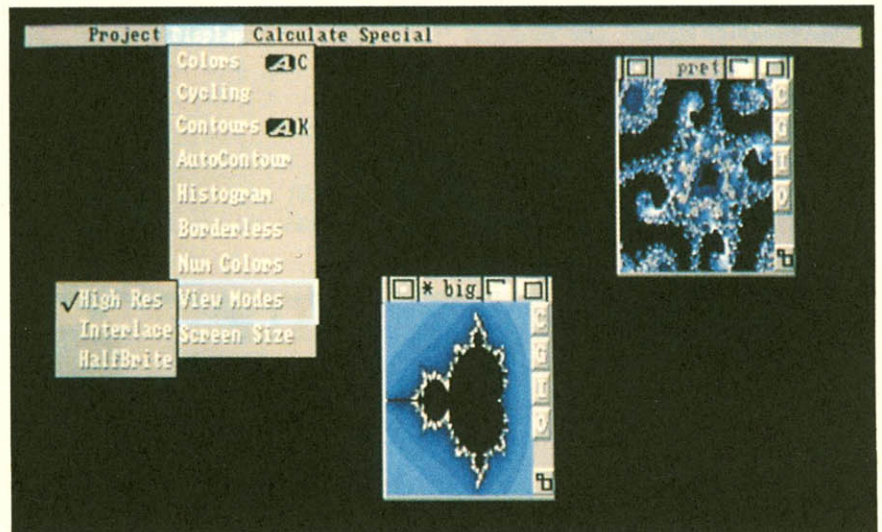


Bild 7. Mit »View Modes« haben Sie alle Bildschirm-Modi im Griff

sich: Wenn Projekt »A« geöffnet ist, als neues Projekt »B« gestartet wird und Sie dann die Zoom-Funktion von »B« auf »A« ausrichten, so sehen Sie in »B« als Vorausinformation grob gerastert die Vergrößerung des eingerahmten Teils. So praktisch diese Vorschau auch sein mag, sie kostet Rechenzeit und verlängert die Reaktionszeit auf Mausbewegungen etc. Mit *Off* schalten Sie diese Funktion ab, mit *On* wieder ein.

Math Mode

»Mandelvroom« läßt Ihnen die Wahl, wie genau und wie schnell Ihre Grafiken berechnet werden sollen. Dieser Punkt bietet Ihnen insgesamt fünf Variationen.

Int68000

Hier wird mit dem Befehlssatz des MC68000 gearbeitet, und es werden alle Rechenoperationen auf der Integer-Ebene abgewickelt. Dies ist die schnellste Art, mit einem Standard-Amiga (ohne Umrüstung) zu Apfelmännchen zu kommen, wobei allerdings naturgemäß die Genauigkeit etwas nachläßt.

Indikation abgeben: »big brot«, das Start-Apfelmännchen, ist in der Default-Ausführung mit »Int68000« nach 30 s fertig berechnet. Mit der IEEE-Variante war die Berechnung nach 7 min und 23 s fertig. Bei solchen Rechenzeiten schleicht sich verständlicherweise der Wunsch nach einer schnelleren CPU und einem Arithmetik-Coprozessor in die Gedanken des Users.

Int68020 und 020/881

Für den 68020 und den 68881-Coprozessor ist »Mandelvroom« vorbereitet. Wenn Sie einen oder beide Prozessoren in Ihrem Amiga eingebaut haben, steht einer schnellen Berechnung des Mandelbrots nichts mehr im Wege.

Maxiteration

Wenn Sie diesen Menüpunkt auswählen, öffnet »Mandelvroom« ein Eingabe-Window, in dem Sie die von Ihnen gewünschte höchste Anzahl

von Iterationen je Pixel eintragen. Die Eingabe wird automatisch auf einen Wert von 1023 reduziert. Daß für jedes Projekt dieser Wert separat eingegeben werden kann, sei als Selbstverständlichkeit nur am Rande erwähnt.

Statistics

Der letzte Menüpunkt innerhalb »Calculate« gibt Ihnen sachdienliche Daten über Real- und Imaginärwerte zum aktuellen Projekt.

Special

Zunächst wollen wir uns das Submenü **Preset** näher ansehen. Kevin Clague hat hier zehn Definitionen fix und fertig abgelegt, die nur darauf warten, daß Sie sie anwählen.

Orbit

Nach der Definition von Benoit Mandelbrot ist »Orbit« die Folge jener Punkte innerhalb einer Mandelbrot- oder Julia-Menge, die im selben Rechenvorgang ermittelt wurden. Der Aufruf der »Orbit«-Funktion ist denkbar einfach. Wählen Sie im »Special«-Menü den Punkt »Orbit« an. Das sich darauf öffnende Window ist zunächst inhaltslos. Wenn Sie jetzt mit dem Mauszeiger in ein Projektfenster gehen und einen beliebigen Punkt anklicken, werden im »Orbit«-Window die relevanten Punkte abgebildet. Wenn Sie die linke Maustaste gedrückt halten, können Sie durch Hin- und Herfahren mit der Maus jeden beliebigen Bereich erkunden und erhalten das Ergebnis postwen-

Theorie der Fraktale

dend im »Orbit«-Window ausgegeben. Welche Rechengenauigkeit dabei angewandt wird, bestimmen Sie mit dem nächsten Menüpunkt:

Orbit Mode

Sie haben die Wahl zwischen normaler 68000-Integer-Arithmetik, FFP- und IEEE-Arithmetik. Im Gegensatz zu »Math Mode« werden schnellere CPUs und Coprozessoren nicht unterstützt. Die Iterationsgrenze für die »Orbit«-Darstellung bestimmen Sie mit »Max Orbit«, dem letzten Punkt im »Special«-Menü.

Soviel zur Praxis im Umgang mit »Mandelvroom«. Zur Einstimmung auf die Rolle von Fraktalen in der Beschreibung chaotischer physikalischer Systeme werden wir zuerst den Grundstein für unsere Betrachtungen besprechen. Dazu greifen

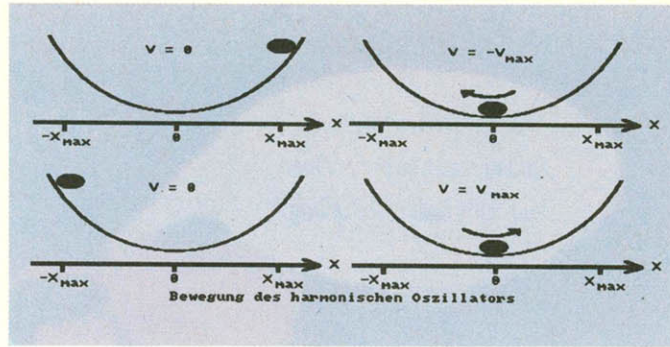


Bild 8.
Anschauliche Darstellung des harmonischen Oszillators

wir auf eines der bekanntesten Modelle aus der Physik zurück: die Bewegung eines idealen Pendels, auch harmonischer Oszillator genannt. Wir können uns statt eines Pendels eine Situation wie in Bild 8 vorstellen: Eine Kugel befindet sich auf einer parabolisch geformten Oberfläche an einem bestimmten Ort (X-Koordinate). Zugleich hat diese Kugel eine gewisse Geschwindigkeit. Beobachten wir das Verhalten dieser Kugel im Verlauf der Zeit, so werden wir feststellen, daß sie unter dem Einfluß der Erdanziehung den Abhang hinab rollt, wobei ihre Geschwindigkeit zunimmt. Nachdem sie den tiefsten Punkt passiert hat, rollt sie mit abnehmender Geschwindigkeit den gegenüberliegenden Abhang hinauf, bis ihre Geschwindigkeit Null geworden ist. Nun hat die Kugel ihren höchsten Punkt erreicht, und sie beginnt wieder den Abhang hinabzurollen, bis sie erneut den höchsten Punkt auf der anderen Seite erlangt. Dieses Spiel setzt sich in unserem Modell in alle Ewigkeit fort. Der Grund für die Beliebtheit dieses Modells ist ziemlich trivial: Es ist eines der wenigen Modelle, die sich exakt berechnen lassen. Ist erst einmal der Ort und die Geschwindigkeit der Kugel zu einem bestimmten Zeitpunkt bekannt, lassen sich diese Größen für jeden anderen Zeitpunkt berechnen. Der harmonische Oszillator hat noch eine besondere Eigenschaft: Er ist periodisch, daher hat die Kugel nach einer ganz bestimmten, festen Zeit wieder dieselbe Position und dieselbe Geschwindigkeit. Um fest-

zustellen, ob ein System periodisch ist, gibt es eine elegante grafische Methode: Wir fertigen ein Diagramm, dessen X-Achse den Ort der Kugel darstellt, während die Y-Achse die Geschwindigkeit der Kugel zeigt. Tragen wir in dieses Diagramm nun den Ort und die Geschwindigkeit der Kugel für jeden Zeitpunkt ein, so erhalten wir bei periodischen Systemen eine geschlossene Kurve.

Oszillatoren

Um dies besser zu verstehen, sollten Sie Listing 1 abtippen. Das dort dokumentierte Basic-Programm gibt jeweils einen zufällig gewählten Wert für Ort und Geschwindigkeit der Kugel vor und berechnet dann deren Werte für spätere Zeiten. Für jedes Paar von Anfangswerten ergibt sich dabei eine Ellipse. Im Programm wird nicht die exakte Lösungsformel des harmonischen Oszillators benutzt, sondern ein numerisches Näherungsverfahren, das jeweils neue Orte und Geschwindigkeiten aus den vorhergehenden berechnet. Diese Vorgehensweise spielt bei unseren späteren Betrachtungen eine große Rolle. Die Programmiersprache Basic wurde gewählt, da jeder Amiga-Besitzer über diese Sprache verfügt. Sollten Sie einen Compiler haben, so ist es angesichts der hohen erforderlichen Rechenzeiten zu empfehlen, die Basic-Listings zu compilieren.

Reibung ist attraktiv

Doch nun zurück zu unserem harmonischen Oszillator. Wie Sie sicherlich schon festgestellt haben, widerspricht dieses Modell unserer alltäglichen Erfahrung. Denn schließlich wird niemand glauben,

Die neue POWER PLAY ist da!

Der Traum eines jeden SpieleFreaks geht für einen von Euch bald in Erfüllung.

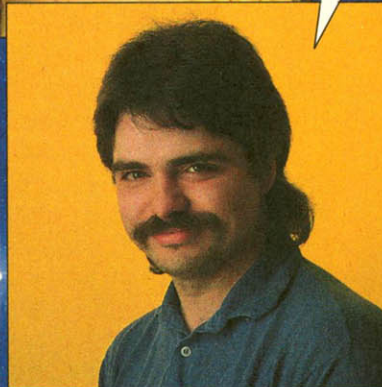


Spielautomat zu gewinnen!

POWER PLAY verlost diesen prächtigen Arcade-Automaten mit dem Spiel "Gradius III" von Konami. Jeder kann mitmachen. Zehn knifflige Fragen warten auf Euch.



Bei Rollenspielen nehmen wir kein Blatt vor den Mund. Wir zeigen Euch wie Ihr Eure Akteure aufpäppelt.



So bringen's Rollenspiele!

POWER PLAY Nr. 9 informiert Euch ausgiebigst über Rollenspiele. Ihr erfahrt, wie ein Rollenspiel entsteht und lernt die wichtigsten Computer-Rollenspiele kennen. Sie dürfen in keiner Sammlung fehlen.

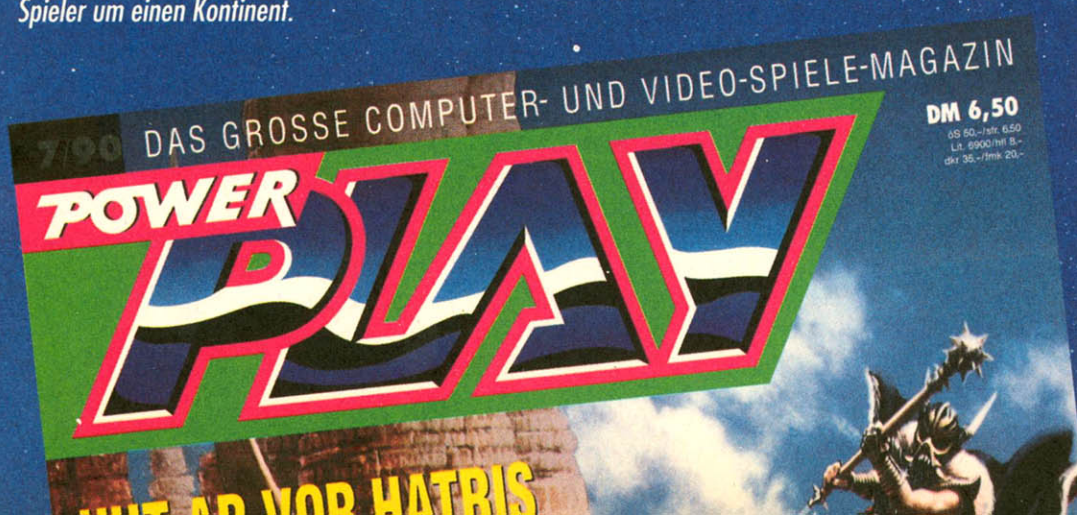
Wir nehmen die neuen Spielelemente von "Powermonger" unter die Lupe und verraten Euch alles über dieses Programm.

Der Populous-Nachfolger kommt!

POWER PLAY zeigt Euch exklusiv die ersten Bilder Populous-Nachfolgers "Powermonger". In diesem potentiellen Superhit balgen sich gleich vier Spieler um einen Kontinent.



Holt Euch POWER PLAY jetzt bei Eurem Händler!



DM 6,50

ISSN 0930-1111 Nr. 650
Lfd. 0900/1111 8-
dkt 95,-/Jah 20,-

daß die Kugel auf alle Ewigkeit in der parabelförmigen Mulde hin und her »surft«. Was wir vernachlässigt haben, ist die bei allen Systemen auftretende Reibung. Diese bewirkt, daß die Kugel von Mal zu Mal weniger hoch hinaus kommt, bis sie schließlich am tiefsten Punkt der Mulde zur Ruhe kommt ($x=0, v=0$). Um die Reibung bei unserer numerischen Simulation zu berücksichtigen, entfernen Sie einfach das obere der beiden »REM«-Schlüsselwörter, und schon sind die Kurven nicht mehr geschlossene Ellipsen, sondern sie laufen alle auf den »Attraktor« $(0,0)$ zu. Daß ein derartiger Attraktor nicht immer aus nur einem Punkt bestehen muß, können Sie feststellen, indem Sie den Reibungsteil des harmonischen Oszillators wieder auskommentieren und statt dessen das zweite REM-Schlüsselwort entfernen.

van der Pols Herz

Der »van der Polsche Oszillator« ist periodisch. Seine wesentliche Eigenschaft ist, daß alle Bahnen im Orts-Geschwindigkeitsdiagramm auf dieselbe geschlossene Kurve zu laufen. Dies entspricht einer »Selbsterregung« des Oszillators. Wenn wir annehmen, daß sowohl der Ort als auch die Geschwindigkeit am Anfang null sind, so wird sich unser System trotzdem in Bewegung setzen und beginnen zu oszillieren. Der »van der Polsche Oszillator« ist ein häufig benutztes Modell der Elektrotechnik (Kippschwingungen) und der Biologie (Kontraktion des Herzmuskels). Der bei diesem Oszillator auftretende Attraktor ist ein sog. Grenzzyklus. Die Form dieses Grenzzyklus läßt sich durch die Wahl der Variablen »k« in Listing 1 wählen: Falls $k = 0$ wird, erhalten wir wieder den harmonischen Oszillator, während wir für große Werte von k eine Rechteckschwingung bekommen. Dabei sollte beachtet werden, daß das Programm in seiner gegenwärtigen Form nicht für große k -Werte ausgelegt ist.

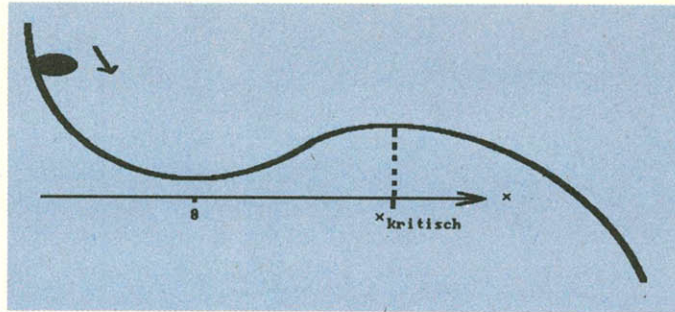


Bild 9. Anharmonischer Oszillator mit kritischer Energie

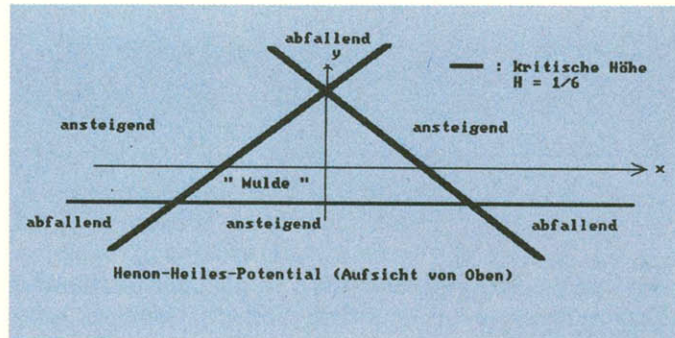


Bild 10. Höhenlinien des »Hénon-Heiles-Potentials«

Wir erlauben unserer Kugel nicht mehr nur in eine Richtung zu schwingen, sondern in zwei. Diese Richtungen sollen in Zukunft mit x und y bezeichnet werden. Doch ein weiterer Oszillator allein bringt noch nicht das rechte Hochgefühl. Daher beschließen wir, die beiden Oszillatoren mit Hilfe einiger zusätzlicher mathematischer Terme zu koppeln. Diese Terme verändern die Fläche, auf der unsere Kugel rollt. Setzt man sich etwa auf die durch $x = 0$ bestimmte Achse, so sieht die Fläche in y -Richtung besehen ungefähr wie in Bild 9 aus.

Sternenhimmel

Daraus können wir sofort folgern, daß es nun eine kritische Höhe gibt, die unsere Kugel nicht überschreiten darf. Tut sie dies dennoch, so wird sie den rechts der Mulde gelegenen Abhang hinabrollen und auf nimmer Wiedersehen verschwinden. Blickt man genau von oben auf unsere Rollfläche, so bietet sich ein entsprechender Anblick (Bild 10). Die unserer Wahl der Kopplungsterme entsprechende kritische Höhe ist übrigens $1/6$ (Listing 2). Mit Hilfe dieses Systems versuchten die beiden französischen Astronomen Hénon und Heiles die Bewegungen von Sternsystemen in Galaxien zu beschreiben. Bei der grafischen Darstellung unseres Systems stehen wir allerdings vor einem gewissen Problem: Es treten insgesamt vier

Variablen auf (die Orte x, y und die Geschwindigkeiten V_x, V_y). Daher ist es üblich, nur eines der beiden

Orts-Geschwindigkeitsdiagramme (in unserem Fall die y -Richtung) darzustellen. Genau dann, wenn unsere Kugel von der rechten oder linken Hälfte in die andere Hälfte wechselt ($x = 0$), werden die y -Position und die y -Geschwindigkeit in das Diagramm eingetragen. Dies kann man sich so vorstellen, daß sich in der durch $x = 0$ bestimmten Ebene eine Lichtschranke befindet, die einen Fotoapparat auslöst, der das Orts-Geschwindigkeitsdiagramm der Kugel aufnimmt und dessen Film nicht weitertransportiert wird. Wie oben schon erwähnt, darf die kritische »Höhe« (in unserem Fall handelt es sich um die Energie) von $1/6$ nicht überschritten werden. Berechnet man die Hénon-Heiles-Abbildung für die Energien $1/24, 1/12, 1/9, 1/8$ und $1/6$, so erkennt man bei den niedrigen Energien im $Y-V_y$ -Diagramm noch geordnete Strukturen, die jedoch mit steigender Energie immer geringer werden und bei $E = 1/6$ schließlich vollkommenes Chaos zeigen. Beachten Sie, daß die Berechnungen für jede dieser Energien jeweils mehrere Stunden dauern. Sie sollten daher eine gehörige Portion Zeit mitbringen.

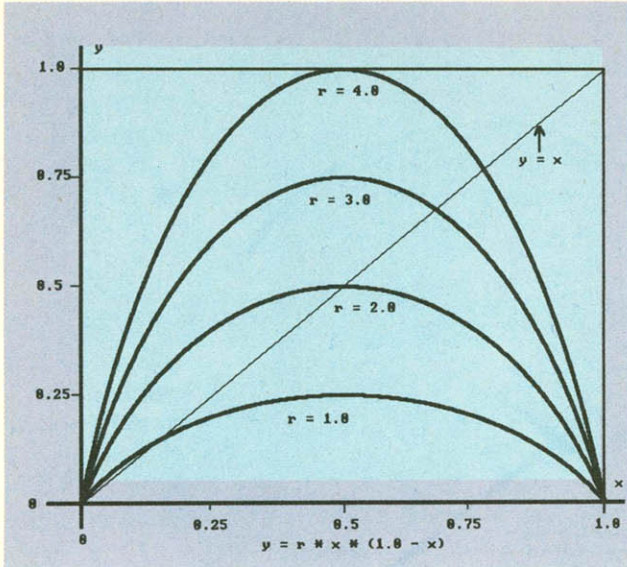


Bild 11. Funktion der logistischen Abbildung für unterschiedliche Parameterwerte

Aus der Hénon-Heiles-Abbildung können wir den Schluß ziehen, daß der Übergang zu chaotischem Verhalten von einem Kontrollparameter des Systems (in diesem Fall der Energie) geregelt wird. Eine weitere Eigenschaft, die in allen von uns bis jetzt betrachteten Systemen auftrat, ist die Iteration: Die neuen Werte der Orte und Geschwindigkeiten werden nach einer vorgegebenen Regel aus den alten Werten berechnet. Um die Orte und Geschwindigkeiten zur Zeit t_1 zu berechnen, wobei wir die Orte und Geschwindigkeiten zur Zeit t_0 kennen, berechnen wir diese Werte zur Zeit $t_0 + dt$. Aus diesen Werten ermitteln wir die Werte $t_0 + 2 * dt$ und so fort, bis wir die Zeit t_1 erreicht haben. Betrachten wir die Abbildung $f(x) = r * x * (1 - x)$ und ihr Verhalten unter Iterationen. In Bild 11 ist der Graph dieser Abbildung für verschiedene Werte des Parameters r wiedergegeben. Da wir den Funktionswert als neue Eingabe in die Funktion verwenden wollen, darf auch das Ergebnis $f(x)$ nicht größer als 1 und nicht kleiner als 0 sein (x liegt zwischen 0 und 1). Dies hat unmittelbar zur Folge, daß der Parameter r nur Werte zwischen 0 und 4 annehmen darf. Wie eine derartige Iteration abläuft, ist in Bild 12 dargestellt: Wir starten mit einem beliebigen Wert x_0 und bilden $y = f(x_0)$. Dann wird $x_1 =$

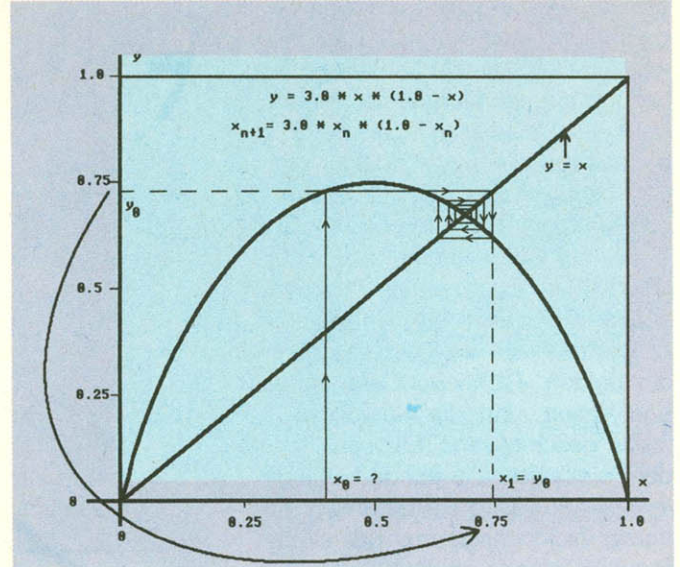


Bild 12. Iteration der logistischen Abbildung

y gesetzt, und das Spiel beginnt von neuem. Grafisch läßt sich dieser Vorgang noch einfacher darstellen. Starte mit x_0 und ziehe eine Linie bis zur Parabel $r * x * (1 - x)$, von der Parabel ziehe eine waagrechte Linie zur Geraden $y = x$, von der Geraden ziehe eine senkrechte Linie zur Parabel etc. Normalerweise würde man erwarten, daß der Attraktor dieses Systems der Schnittpunkt der Geraden $y = x$ mit der Parabel $y = r * x * (1 - x)$ ist. Dies ist jedoch nur für r kleiner 3 der Fall. Für Werte von r größer als 3 oszilliert das System plötzlich zwischen zwei Werten, bei noch größerem r zwischen vier.

Periodenverdopplung

In immer kürzeren Abständen verdoppelt sich die Zahl der Punkte, aus denen der Attraktor besteht, bis der Attraktor bei einem bestimmten Wert von r ($r = 3,5699...$) aus unendlich vielen Punkten besteht. Was bedeutet, daß aus unserem Attraktor ein etwas ungewöhnlicher Attraktor geworden ist. Zugleich haben wir etwas über den Weg vom regulären zum chaotischen Verhalten gelernt: Dieser Übergang geschieht durch eine »Periodenverdopplung« in immer kürzeren Abständen. Dieses Verhalten ist auch anhand von Listing 3 nachzuvollziehen. Im Fenster »Zeitverhalten« werden dabei die ersten Iterationen angezeigt, die ausgeführt werden, um dem System Gelegen-

heit zu geben, von einem beliebigen Anfangswert aus in die Nähe seines Attraktors zu gelangen. Dann wird dieser in das r - x -Diagramm im Fenster darüber eingezeichnet. Die Abbildung $r * x * (1 - x)$ ist unter dem Namen »logistische Abbildung« bekannt. Sie spielt eine Rolle in der Ökologie: Bezeichnet $x(n)$ die Anzahl der Individuen der n -ten Generation einer Population, so ist die Anzahl der Nachkommen abhängig von der Anzahl $x(n)$. Ist der den Individuen zur Verfügung stehende Platz beschränkt, so nimmt er zugleich abhängig von der Anzahl $x(n)$ ab. Die restlichen Faktoren, wie Nahrungsangebot, Vermehrungsrate usw. gehen in den Parameter r ein, so daß sich für die Anzahl der Individuen $x(n+1)$ in der nächsten Generation ergibt:

$$x(n+1) = r * x(n) * (1 - x(n))$$

Eine wichtige Eigenschaft chaotischer Systeme ist ihre Selbstähnlichkeit. Vergrößert man bestimmte Ausschnitte der logistischen Abbildung, so erhält man ein Gebilde, welches der logistischen Abbildung verblüffend ähnlich sieht. Ganz analog verhält es sich mit den Figuren der Hénon-Heiles-Abbildung, jedoch sind dazu so viele Iterationen nötig, daß die in unserem Beispiel benutzten Näherungsverfahren zu unge-

nau werden, von den benötigten Rechenzeiten ganz zu schweigen. Die Selbstähnlichkeit schafft jedoch die Verbindung von chaotischen Systemen zur fraktalen Geometrie. Die Geometrie, die uns normalerweise beigebracht wird, ist die der alten Griechen. Die beherrschenden Elemente sind Punkte, gerade Linien und Kreise. Diese Einfachheit der Beschreibung hat jedoch auch ihren Preis: Mit den uns in der Natur umgebenden Objekten haben die Elemente wenig gemein, die mit dieser Geometrie beschrieben werden können. Fraktale dagegen werden nicht dadurch beschrieben, daß man etwa zwei vorgegebene Punkte mit einer geraden Linie verbindet, sondern indem auf ein Ausgangsobjekt (das durchaus eine Linie sein kann) eine bestimmte Regel anwendet. Diese Regel erzeugt neue Objekte, auf welche die Regel erneut in Anwendung gebracht werden kann. Wie Sie se-

Dynamische Systeme

hen, sind wir wieder bei der Iteration angelangt. Als Beispiel kann die Kochsche Kurve dienen (Listing 4). Wir gehen von einer geraden Linie aus und ersetzen diese durch das dargestellte Objekt. Nun verkleinern wir dieses Objekt entsprechend und ersetzen nun jede Linie im ursprünglichen Objekt durch das verkleinerte Objekt. Wiederholt man diesen Vorgang unendlich oft, so entsteht ein fraktales Gebilde. Dieses Gebilde besitzt einige seltsame Eigenschaften, wie Sie beim Versuch die Länge der Linie zu berechnen, durch die dieses Objekt gebildet wird, erkennen können: Bei jeder Ersetzung nimmt die Länge des neuen Objektes um ein Drittel der Länge des ersetzten Objektes zu. Daher ist die Linielänge des Fraktals unendlich, obwohl er sich nur in einem endlich großen Gebiet aufhält.

Da die oben besprochenen physikalischen Systeme eine Bewegung beschreiben, werden sie häufig als dynamische Systeme bezeichnet. Allen diesen Systemen ist gemein, daß ihr Zeitverhalten durch eine Iteration berechnet wird. Ausgehend von diesem Sachverhalt, bezeichnet man nun die Iteration einer mathematischen Funktion, also $x(n+1) = f(x(n))$ als dynamisches System. Die

Menge der bei der Iteration entstehenden Werte $x(0), x(1), x(2)$ etc. heißt »Orbit« des Systems. Eine interessante Frage ist: Kann man das Verhalten des Orbits vorhersagen? Betrachten wir die Funktion $f(x) = x^2$. Wählen wir als $x(0)$ eine Zahl vom Betrag kleiner als 1, so wissen wir, daß $x(n)$ mit wachsendem n immer näher an die 0 rücken wird. Andererseits wird für Anfangswerte mit Betrag größer 1 unser Funktionswert mit

Es wird komplex

wachsendem n immer größer. Daher besitzt dieses System die beiden Attraktoren 0 und Unendlich. Orbits, bei denen kleine Abweichungen in den Anfangswerten zu keiner dramatischen Änderung des Orbits führen, heißen stabil. Wählen wir als konkreteres Beispiel unseren harmonischen Oszillator. Eine leicht verschiedene Anfangsgeschwindigkeit führt nur dazu, daß die Kurven im X-V-Diagramm leicht nebeneinander verlaufen. Vollkommen anders sieht der Fall bei unserer Funktion $f(x) = x^2$ aus, wenn wir als Anfangswert $x(0)$ die Zahl 1 wählen. Unter der Iteration bleibt diese Zahl erhalten, denn $1 \cdot 1 = 1$ - es liegt ein Fixpunkt des Systems vor. Dieser Fixpunkt ist nicht stabil, denn wählen wir einerseits $x(0)$ als 0,999999, so wird der Orbit auf die Null zulaufen. Wählen wir andererseits $x(0)$ als 1,000001; damit wird der Orbit ins Unendliche laufen. Die Menge der Punkte, deren Orbit nicht stabil ist, heißt die »chaotische Menge« des Systems.

Die wirklich interessanten dynamischen Systeme sind jedoch nicht eindimensional, sondern wie das Hénon-Heiles-System zweidimensional. Aus Gründen der mathematischen Einfachheit betrachtet man bei solchen Systemen mit Vorliebe Funktionen von komplexen Variablen. Wandeln wir unsere obige Funktion daher um in $f(z) = z^2$, so ändert sich an den stabilen Orbits nichts. Ist z betragsmäßig kleiner als 1, laufen die Orbits auf die 0 zu. Ist z vom Betrag größer als 1, läuft der zugehörige Orbit ins Unendliche. Die Werte z vom Betrag 1 bilden den Kreis vom Radius 1 um den Ursprung des Koordinatensystems. Ein Kreis hat wenig Chaotisches an sich. Woher rechtfertigt sich der Name

»chaotische Menge« für die nicht stabilen Orbits? Die Antwort ist nicht offensichtlich, jedoch läßt sich mathematisch zeigen, daß sich die Orbits auf diesem Kreis exakt so verhalten wie die logistische Abbildung für den Wert $r = 4$. Dies heißt aber, daß über zwei Startpunkte $z_1(0)$ und $z_2(0)$, die sich anfänglich nur wenig unterscheiden, keine Aussagen darüber gemacht werden können, wie weit sie sich nach sehr vielen Iterationen voneinander entfernt haben. Diese Unfähigkeit, den Abstand zweier anfänglich benachbarter Punkte berechnen zu können, rechtfertigt den Namen »Chaos«. In komplexen dynamischen Systemen wird die chaotische Menge oft auch nach ihrem Entdecker benannt, dem französischen Mathematiker Julia.

Julia und Mandelbrot

Julia-Mengen sehen nur selten so einfach aus wie in unserem Beispiel. Meist handelt es sich bei ihnen um Fraktale. Doch auch diese Fraktale können noch unterschiedliches Verhalten zeigen. Stellen wir uns etwa vor, wir wählen einen beliebigen Punkt, der zur Julia-Menge eines Systems gehört. Könnten wir dann von diesem Punkt aus eine ununterbrochene Linie zu jedem anderen Punkt aus der Julia-Menge ziehen, ohne dabei die Julia-Menge jemals verlassen zu müssen? Mengen mit dieser Eigenschaft heißen »zusammenhängend«. Die Antwort auf diese Frage ist manchmal ja und manchmal nein. Daher lassen sich Julia-Mengen danach unterscheiden, ob sie zusammenhängend sind oder nicht. Die bekanntesten Bilder von Julia-Mengen entstammen der Abbildung:

$$f(z) = z^2 + c$$

Dabei ergeben sich für jede Wahl der Konstanten c unterschiedliche Julia-Mengen. Die Menge aller Konstanten c , die zu einer zusammenhängenden Julia-Menge führen,

Literaturhinweise

Allgemeinverständlich:

1. B. Mandelbrot, Die fraktale Geometrie der Natur, Birkhäuser Verlag 2. H.-O. Peitgen, P.-H. Richter, The Beauty of Fractals, Springer Verlag 3. A. K. Dewdney, Computer-Kurzweil, Spektrum der Wissenschaft, Heft 10/85 4. Chaos und Fraktale, Spektrum der Wissenschaft: Verständliche Forschung

Mathematisch:

5. H.-O. Peitgen, D. Saupe, The Science of Fractal Images, Springer Verlag 6. H.-G. Schuster, Deterministic Chaos, Physik-Verlag Weinheim 7. A. Kunick, W.-H. Steeb, Chaos in dynamischen Systemen, BI Wissenschaftsverlag

Software (Amiga):

K. Clague, Mandelbroom, auf Fish-Disk 215 (Public Domain)

wird nach ihrem Entdecker Benoit Mandelbrot benannt, im deutschen Sprachraum ist diese Menge auch als »Apfelmännchen« bekannt. Die Mandelbrot-Menge ist das komplizierteste mathematische Objekt, das zur Zeit bekannt ist.

Somit können wir Ihnen zum Abschluß nur viel Spaß bei Ihren Entdeckungsreisen in die Welt der Fraktale wünschen. ag

Programmname: Logistics.bas

Computer: A500, A1000, A2000 mit Kickstart 1.2 & 1.3

Sprache: Amiga-Basic

Bemerkung: siehe Text

Programmautor: Jürgen K. Singer

```
1 CLO RANDOMIZE TIMER
2 pT a$ = ""
3 RB WHILE a$ <> "e" ' Taste >>e<< beendet Programm
4 rV2 a$ = ""
5 6X c = SQR(50)
6 z6 x = 2*c * RND - c ' -c < x < c (Ort)
7 6g v = 2*c * RND - c ' -c < v < c (Geschwindigkeit)
8 Dy k = .8 ' Dämpfung (harm. Osz.), Kopplung (Pol
1)
9 vu dt = .005 ' Zeitschritt
```

```
10 Hm WHILE a$ = "" ' Beliebige Taste (ausser >>e<<) w
aehlt
11 Au4 a$ = INKEY$ ' neue Startwerte x und v
12 SF E = .5 * (v*v + x*x)
13 fD LOCATE 1, 1
14 nc PRINT USING "E = ###.##"; E ' Energie
15 Ps PRINT USING "v = ###.##"; v ' Geschwindigkeit
16 yO PRINT USING "x = ###.##"; x ' Ort
17 kx valt = v ' Berechnung der Bewegung
:
18 hc v = valt - x * dt ' harmonischer Oszillator
19 zP0 REM v = v - k*valt * dt ' harm. Osz. mit Reibung
20 w7 REM v = v + k*(1-x*x)*valt * dt ' van der Pol's Oszillato
r
21 Wu4 x = x + v * dt
22 iR PSET (20*x+300, -10*v+100)
23 oo2 WEND
24 gm IF a$ = "c" THEN CLS
25 qe0 WEND
(C) 1990 M&T
```

Listing 1. Numerische Simulation des harmonischen Oszillators mit und ohne Reibung

Programmname: Logistics.bas

Computer: A500, A1000, A2000 mit Kickstart 1.2 & 1.3

Sprache: Amiga-Basic

Bemerkung: siehe Text

Programmautor: Jürgen K. Singer

```
1 CLO RANDOMIZE TIMER
2 kv DEF FNv(x,y) = .5*(x*x+y*y)+x*x*y-y*y*y/3! ' Henon-Heiles-Po
tential
3 Se SCREEN 2, 640, 256, 1, 2 ' Graphische Dar
stellung
4 4C WINDOW 3, "X-Y-Ebene", (0,0)-(310,190),20,2 ' auf Screen mit
mehreren
5 Xv WINDOW 4, "Y-Vy-Ebene", (310,0)-(630,190),20,2 ' Fenstern
6 Ft WINDOW 2, "Daten", (0,200)-(630,240),20,2
7 uY a$ = ""
8 hQ maxcount% = 100
9 3E lauf% = 0
10 tH y0 = 150 : vy0 = 100 ' Nullpunkt innerhalb der Fenster
11 ar ysc = 150 : vsc = 100 ' Skalierung auf Fenster-Koordinaten
12 iN dymin = .001 ' Kleinster Schritt in y-Richtung
13 i5 dt = .05 ' Zeitschritt
14 8j WHILE a$ <> "e" ' falls a$ = "e", Programm beenden
15 k22 WINDOW OUTPUT 2
16 G4 again:
17 l74 LOCATE 1,1 ' Erfrage Wert der Energie
18 AN PRINT SPACE$(70);
19 r2 LOCATE 1,1
20 iF INPUT "Energie (0 < E < 0.16666): ",Ein
21 JS2 IF Ein < 0 OR Ein > 1/6 GOTO again ' Falsche Energie?
22 o8 WHILE a$ <> "e"
23 OG4 lauf% = lauf% + 1 ' Zaehlen Laeufe mit unterschied
lichen
24 sO E = Ein : a$ = "" ' Anfangsbedingungen
25 pT x = 0 : t = 0
```

```
26 sv Ymax = 0! : dy = .125 ' Berechne fuer diese Energie, d
en
27 gP WHILE dy > dymin ' maximal und den minimal moegl
ichen
28 V16 IF FNv(x,Ymax) < E THEN ' y-Wert
29 R98 Ymax = Ymax + dy
30 5o6 ELSE
31 7N8 dy = .5 * dy
32 SL6 END IF
33 ym4 WEND
34 hR Ymin = 0 : dy = .125
35 oF WHILE dy > dymin
36 jb6 IF FNv(x,Ymin) < E THEN
37 rt8 Ymin = Ymin - dy
38 Dw6 ELSE
39 FV8 dy = .5 * dy
40 aT6 END IF
41 6u4 WEND
42 iI y = (Ymax-Ymin) * RND + Ymin ' waehle y mit Ymin < y
< Ymax
43 Mg Vx = 2*( E - FNv(x,y) )
44 7W Vy = SQR(Vx) * RND ' waehle Y-Geschwindigkeit
45 x5 Vy = Vy * .9 ' Sicherheitsfaktor
46 dm Vx = SQR(Vx - Vy*Vy) ' berechne X-Geschwindigkeit
47 df count% = 0 ' Zähle, wie oft Teilchen x=0 durchlauft
48 nF WHILE a$ <> "e" AND a$ <> "n" ' weder beendet noch n
eue Energie?
49 Ia6 WINDOW OUTPUT 2
50 O3 a$ = INKEY$
51 pr LOCATE 2,1 ' Ausgabe von:
52 rW PRINT USING "Zeit: ###.##";t; ' Zeit
53 Mw PRINT USING "E = ###.##";E ' Energie
54 Io PRINT USING "X = ###.##";x; ' X-Koordinate
55 lo PRINT USING "Y = ###.##";y ' Y-Koordinate
56 WR PRINT USING "Vx = ###.##";Vx; ' X-Geschwindig
keit
57 V2 PRINT USING "Vy = ###.##";Vy ' Y-Geschwindig
keit
58 AS PRINT USING "N = ###.";count%; ' Anzahl x=0 Durc
hlaeufe
```



```

59 ot PRINT USING " Lauf: ###"; lauf%; ' Anzahl Anfangsb
ed.
60 bz Xold = x
61 i8 Yold = y
62 20 t = t + dt ' naechster Zeitsch
ritt
63 VM E = .5*(Vx*Vx + Vy*Vy) + FNV(x,y) ' aktuelle Energie
64 HY x = x + Vx * dt ' neue Orte
65 8Z y = y + Vy * dt
66 XU Vx = Vx - (x+2!x*y) * dt ' und neue Geschwin
digkeit
67 1K Vy = Vy - (y+x*x-y*y) * dt
68 M5 IF (SGN(x)*SGN(Xold) <= 0) THEN ' Falls x=0 Durchg
ang,
69 628 WINDOW OUTPUT 4 ' dann zeichne Punkt
70 rl PSET (ysc*y+y0, vsc*Vy+vy0) ' in Y-Vy-Diagramm
71 5R count% = count% + 1 ' und erhoehc Zaehle
r
72 oo IF count% = maxcount% THEN a$ = "n" ' neue Anfangswe
rte?
73 706 END IF
74 8o WINDOW OUTPUT 3 ' Zeichne Position
im
75 6q PSET(ysc*x+y0,vsc*y+vy0) ' X-Y-Diagramm
76 bk IF a$ = "c" OR a$ = "n" THEN CLS ' X-Y-Diagramm loes
chen?
77 XI IF a$ = "c" THEN a$ = ""
78 hV4 WEND
79 iW2 WEND
80 jX0 WEND
81 Fk WINDOW CLOSE 2 ' Bildschirm aufraeumen
82 bI WINDOW CLOSE 3
83 hP WINDOW CLOSE 4
84 DJ SCREEN CLOSE 2
(C) 1990 M&T

```

Listing 2. Numerische Simulation der Hénon-Heiles-Abbildung

Programmname: Logistics.bas
Computer: A500, A1000, A2000 mit Kickstart 1.2 & 1.3
Sprache: Amiga-Basic
Bemerkung: siehe Text

Programmautor: Jürgen K. Singer

```

1 JX0 xmax = 630
2 BN ymax = 240
3 PC st = xmax/100
4 dW SCREEN 2,640,512,1,4
5 ej WINDOW 2,"Logistische Abbildung",(0,0)-(xmax,ymax),20,2
6 CR WINDOW 3,"Zeitverhalten",(0,ymax+12)-(xmax,2*ymax),20,2
7 dI FOR j = 1 TO xmax
8 jx3 WINDOW OUTPUT 3
9 dx r = 1.5 * j/xmax + 2.5
10 wv x = RND
11 En PSET (1,ymax*x)
12 pE FOR i = st TO xmax STEP st
13 L47 x = r * x * (1! - x)
14 bc LINE -(1,ymax*x)
15 gw3 NEXT i
16 13 WINDOW OUTPUT 2
17 Qb FOR k = 1 TO 100
18 Q96 x = r * x * (1! - x)
19 7b PSET (j,ymax*x)
20 p73 NEXT k
21 wA WINDOW OUTPUT 3
22 u0 CLS
23 q70 NEXT j
24 wu WHILE INKEY$ = "" : WEND
25 bh WINDOW CLOSE 2
26 h0 WINDOW CLOSE 3
27 IO SCREEN CLOSE 2
(C) 1990 M&T

```

Listing 3. Logistische Abbildung

Programmname: Logistics.bas
Computer: A500, A1000, A2000 mit Kickstart 1.2 & 1.3
Sprache: Amiga-Basic
Bemerkung: siehe Text

Programmautor: Jürgen K. Singer

```

1 A10 ' CLEAR ,100000& ' Bei planes > 2 wird mehr Speicher be
noetigt
2 qn2 planes = 3
3 0j SCREEN 2,640,256,planes,2 ' Eigener Bildschirm und eigen
es Fenster
4 aS WINDOW 2, "Koch - Kurven", (0,0)-(620,240),20,2
5 A2 farben = 2^planes ' Anzahl der moeglichen Farben
6 SH punkte = farben^4 ' Anzahl von Eckpunkten der Ku
rve
7 KO DIM x(punkte),y(punkte) ' Merken der Eckpunkte
8 fP DIM SHARED c ' Hoehe eines gleichseitigen D
reiecks ist
9 dI c = SQR(3)/2 ' sqr(3)/2 * Seitenlaenge
10 JMO main:
11 xB2 x(0) = 50 : y(0) = 100 ' Aeusserste Eckpunkte
12 Lt x(punkte) = 290 : y(punkte) = 100
13 3J k = punkte
14 zn FOR i = 1 TO farben-1 ' Iterationstiefe wird durch Farbenz
ahl bestimmt
15 We4 s = 0 ' Starte am linken Rand
16 tN WHILE s < punkte ' Solange rechter Rand nicht erreic
ht
17 816 e = s + k ' Ersetze Linie zwischen s und e dur
ch die in
18 pF l = s + k/4 : m = s + k/2 : r = s + 3*k/4 ' Regel de
finierte Figur
19 o2 Regel x(s), y(s), x(1), y(1), x(m), y(m), x(r), y(r),
x(e), y(e)
20 QI LINE (2*x(s),y(s))-(2*x(1),y(1)),1
21 wn LINE -(2*x(m),y(m)),1
22 RS LINE -(2*x(r),y(r)),1
23 Cn LINE -(2*x(e),y(e)),1
24 3o s = e ' Alle Linien
25 qe4 WEND
26 G7 k = k / 4 ' Figuren werden immer kleiner
27 s82 NEXT i
28 Oy WHILE INKEY$ = "" : WEND
29 fL WINDOW CLOSE 2
30 LR SCREEN CLOSE 2
31 FAO END
32 Yw SUB Regel( xs, ys, xl, yl, xm, ym, xr, yr, xe, ye ) STATIC
33 rl '-----
34 6b ' Unterprogramm Regel berechnet Eckpunkte von:
35 Yd ' m Eingabe: Punkte s und e
36 T3 ' ^ Rueckgabe: Punkte s, l, m, r,
e
37 FQ ' s / e globale Variable: c
38 iN ' 1 r
39 xr '-----
40 DP3 xc = (xe-xs)/3
41 PZ yc = (ye-ys)/3
42 Mu xl = xc + xs
43 T4 yl = yc + ys
44 Ls xr = xc + xl
45 S2 yr = yc + yl
46 a2 xp = (xl + xr)/2
47 jE yp = (yl + yr)/2
48 4e xm = xp - c*(yr-yl)
49 rP ym = yp + c*(xr-xl)
50 qs0 END SUB
(C) 1990 M&T

```

Listing 4. Demonstration von Fraktalen - die Kochsche Kurve

Faszinierende Bilder

Was helfen einem wunderschöne digitalisierte Bilder, wenn sie nicht mehr weiterverarbeitet werden können. Gott sei Dank gibt es Programme, mit denen sich umfangreiche Manipulationen anstellen lassen.

von Marco Vitolini-Naldini

Auf dem kommerziellen Markt sind es bereits vier solcher Bildweiterverarbeiter zu haben: »Butcher«, »Pixmate«, »Colors« (Zusatzprogramm von »Deluxe Photolab«) und »Transfer24« (Zusatzprogramm von »Digi-View« und »Digi-Paint 3«). »Pixmate« wollen wir nun herauspicken und näher betrachten.

Angenommen, man will ein Hold-And-Modify-(HAM-)Bild in »Deluxe Paint« weiterverarbeiten. Sehr schnell wird man die Flinte ins Korn werfen - es geht nämlich nicht. Höchstens lassen sich mit einigen HAM-Programmen Lores-Bilder einladen und weiterverarbeiten; damit wären die Möglichkeiten aber schon erschöpft. In einem Softwareladen sollte man sich nach Programmen umsehen, die Bilder in allen nur erdenklichen Möglichkeiten bearbeiten können.

Bildverarbeitung - Image Processing nennt sich die Aufbereitung digitaler Bilder. Was sich so hochgestochen anhört, ist in Wirklichkeit nichts anderes als Änderungen der Auflösung oder Farbpalette eines bereits existierenden Bildes sowie Farbveränderungen einzelner Bildpunkte. Daß dies aber auch sehr umfangreich werden kann, wollen wir anhand dieses Workshops zu »Pixmate« demonstrieren.

»Pixmate« liegt mittlerweile in der Version 1.1 und in Deutsch vor. Das Handbuch ist ebenfalls in Deutsch

und erklärt ausführlich alle einzelnen Funktionen. Die Hauptanwendung dieses Programms wird sicherlich in der Konversion der verschiedenen Auflösungen des Amiga liegen. Zusätzlich stehen eine fast unüberschaubare Anzahl weiterer Manipulationen bereit, die sich hauptsächlich auf die Veränderung der Farbpalette beziehen. Durch Kombinieren verschiedener Funktionen ergeben sich weit über 3000 Möglichkeiten, sein Bild nach eigenen Vorstellungen umzuarbeiten.

Umwandlung à la carte

Fangen wir also an und laden ein HAM-Bild, beispielsweise eine durch einen Ray-Tracer errechnete Grafik, in den Speicher (Bild 1). Nach abgeschlossenem Ladevorgang (das Load- und Save-Requester ist eines der vorbildlichsten auf dem Amiga), befindet sich das Bild sichtbar auf dem »Pixmate«-Screen. Jetzt geht's an die Umwandlung dieses Bildes vom HAM-Modus in Lores, da wir es für irgendwelche Zwecke in Deluxe-Paint-III weiterbearbeiten wollen. Dazu wählt man den entsprechenden Menüpunkt »Effekte/HAM —> 32« an, worauf sich innerhalb weniger Sekunden ein beeindruckendes Schauspiel auf dem Bildschirm zeigt. Ist die Umrechnung schließlich abgeschlossen, wird das Bild in nunmehr 32 Farben auf dem Screen sichtbar. Spätestens jetzt, also im direkten Vergleich, erkennt man die Vorzüge des HAM-Modus. Nach einem kurzen Igit und einem Druck auf die U-Taste (UNDO) glänzt die Grafik wieder im Farbenrausch. Was also tun, um die Qualität des Bildes

nicht ganz zu verlieren? Der Extra-Halfbrite-Modus mit seinen 64 Farben könnte einen brauchbaren Kompromiß ergeben. Schließlich verfügt Deluxe-Paint-III auch über diese Auflösung, weswegen eine 64-Farben-Konversion ebenfalls möglich ist. Also wählen wir den Menüpunkt »Effekte/HAM —> 64« an. Einige Sekunden und Bildschirmblitze später sitzt man einer 64-Farben-Version gegenüber, die zwar auch nicht mehr so hübsch anzusehen ist, aber doch einen gesunden Kompromiß darstellt (Bild 2). Zu solchen Umwandlungen ist sicherlich noch hinzuzufügen, daß die Qualität der farbreduzierten Grafik stark vom HAM-Bild abhängig ist. Verfügt das Original über zahlreiche Schattierungen der unterschiedlichsten Farben, so kann nach der Umwandlung natürlich nicht mehr viel davon übrigbleiben. Ebenso gibt es HAM-Grafiken, die sich von der 32-Farb-Version auf den ersten Blick gar nicht unterscheiden.

Information ist alles

Nun wollen wir aber noch einen Schritt weitergehen. Sehen wir uns zunächst die Werte unseres umgewandelten Bildes an. Dazu wählen wir den Menüpunkt »Info/Format«. In einem Fenster erfahren wir, daß es sich um das Halfbrite- und Interlace-Format handelt. Aha, deshalb flimmert es auch so. Was liegt näher, den Interlace-Modus abzuschalten. Auch hierzu existiert ein Menüpunkt: »Effekte/Formatiere.../320 x 256«. Einige Augenblicke später liegt die Grafik in halbiert vertikaler Auflösung vor. Die Kontrolle im »Info/Format«-Menü beweist es. Doch bevor wir das Bild letztendlich auf der Diskette abspeichern, wollen wir uns noch ein paar statistische Informationen darüber einholen. Durch den Menüaufruf »Info/Zählen« werden nun alle Pixel der Grafik gezählt und einer der 64 Farben zugeordnet. Nach abgeschlossenem Zählvorgang erscheint ein Fenster, in dem sich alle Farben aneinandergereiht befinden. Klickt

dmmanipulationen

man nun mit der Maus auf solch ein Farbfeld, gibt ein Text das prozentuale Vorkommen dieser Farbe im Bild an sowie deren Pixelzahl. Damit aber noch nicht genug. Durch Drücken der Taste »H« wird eine komplette Balkengrafik ausgegeben, die auf einen Blick die prozentuale Farbenverteilung im Bild vermittelt. Schließen wir nun das Fenster (Schließ-Gadget oben links) und speichern das konvertierte Bild auf Diskette ab. Hierbei bestehen mehrere Möglichkeiten: Man kann das Bild als normale IFF-Grafik (gepackt oder ungepackt), lediglich die Palette oder ein Rohdatenformat (ohne IFF-Header) speichern. Die Auswahl nehmen Sie im Projekt-Menü vor, das Sie durch

Druck auf die rechte Maustaste erreichen. Im Normalfall steht unter dem Punkt »IFF Komp...« der Schalter auf »An«. Das bedeutet, daß alle abgespeicherten Grafiken komprimiert abgelegt werden und um 20 bis 30 Prozent verkürzt sind. Wenn Sie diesen Schalter auf »Aus« stellen, findet keine Codierung bzw. Decodierung statt, was wiederum schnellere Lade- und Speicherzeiten bewirkt. Je nachdem, wie Sie die Schalterstellung wählen, wird das Bild mit dem Punkt »Speichern« auf dem gewünschten Datenträger gesichert. Wollen Sie das Rohdatenformat oder Palette speichern, benutzen Sie den Punkt »Schreiben...« im Projekt-Menü.

Solche Umwandlungen in Auflösung oder Farbmodi stellen sicherlich die am meisten benutzten Funktionen von »Pixmate« dar. Trotzdem sind solche Arbeiten für das Programm die leichteste Übung. Wesentlich interessanter wird da schon eine weitere Funktion, die Farbab-

Einfache Farbabstimmung

stimmung. Sicherlich kennen Sie das Problem. Man hat sich ein Bild digitalisiert, von einem Ray-Tracer berechnen lassen oder sogar selbst gemalt. Im nachhinein findet man aber die Farben zu verwaschen, den Kontrast zu klein oder das ganze Bild zu dunkel. Hier schafft »Pixmate« endlich Abhilfe. Laden wir uns dazu ein solches Bild beliebiger Auflösung in den Speicher (Bild 3). Mit dem Menü »Farbe/Farbbestimmung« erscheint ein Fenster, in dem sich mehrere Gadgets und Schieberegler befinden. Über den bekannten RGB-Reglern sind drei weitere interessante Gadgets angebracht:

1. Mit Hilfe des Kontrast-Reglers (KON-Gadget) läßt sich, wie der Name schon sagt, der Kontrast des Bildes einstellen. Das heißt, man kann helle Farben heller und dunkle Farben dunkler oder umgekehrt erscheinen lassen (Bild 4). Dabei rechnet sich »Pixmate« vorher einen Helligkeitsdurchschnitt aller Farben aus.
2. Eine weitere Manipulationsmöglichkeit besteht in der Veränderung der Farbsättigung (SAT-Gadget). Hiermit kann man das Bild brillanter und klarer in seiner Farbgebung erscheinen oder bei Reduzierung der Sättigung Pastelltöne entstehen lassen. Nimmt man die Sättigung vollends weg, bleibt nur noch ein in Graustufen erscheinendes Schwarzweiß-Bild übrig.
3. Der Helligkeitsregler (INT-Gadget) verändert die Farbpalette des Bildes dahingehend, daß alle Farben gleichmäßig in ihrer Farbe und Intensität Richtung Schwarz oder Weiß hin verändert werden.

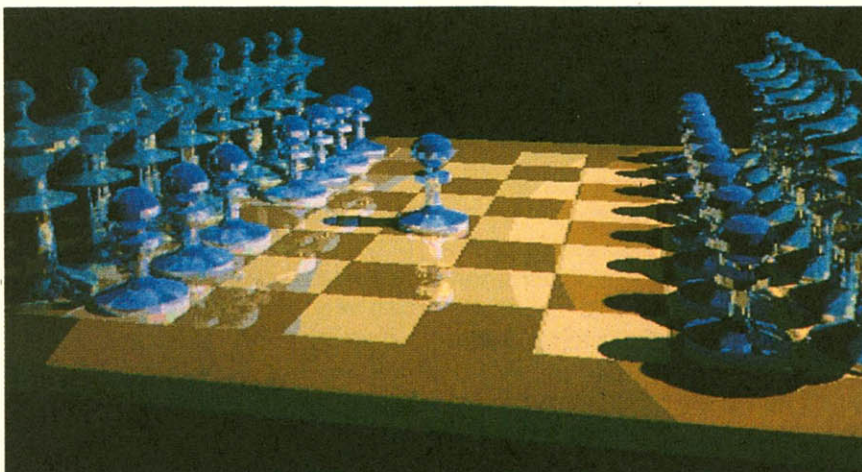
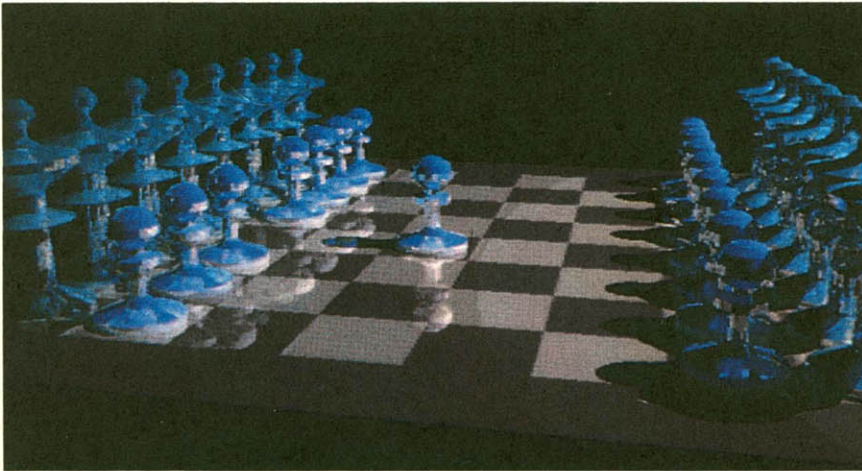


Bild 1+2. Umwandlung eines HAM-Bildes in den Extra-Halfbrite-Modus

Weiterhin existieren drei Schieberegler, mit denen die Rot-, Grün- und Blauanteile eines Bildes einzeln beeinflusst werden können. Schiebt man z.B. den Rot-Regler nach links, wird der Rot-Anteil des Bildes verkleinert und die grünen und blauen Bildkomponenten in ihrer Intensität belassen. Mit diesen Möglichkeiten kann man ein Bild dahingehend bearbeiten, als hätte man es mit einem bestimmten Farbfilter fotografiert (Bild 5). Um die gewünschten Einstellungen dauerhaft vorliegen zu haben, brauchen Sie nur auf das JUST-Gadget zu klicken und das Fenster zu schließen. Bei einem nochmaligen Aufruf der Farbbestimmung werden Sie das Bild nicht mehr mit dem RESET- oder RÜCK-Gadget in seinen Urzustand zurückversetzen können.

Effekthascherei

Kommen wir nun zu einem Menüpunkt, mit dem sich das auf dem Screen befindliche Bild regelrecht in seinen Bestandteilen manipulieren läßt. Wählen wir dazu »Effekte/Bildschirm« mit der Maus an, worauf sich wieder ein Fenster mit einer Unmenge an Gadgets öffnet. Hier können alle Einstellungen des Screens vorgenommen werden. Es lassen sich beispielsweise einzelne Bitplanes an- und abschalten oder vertauschen. Weiterhin kann man von Lores auf Hires hin- und herschalten, HAM-Modus oder Extra-Halfbrite aktivieren sowie das Zeilensprungverfahren des Interlace-Modus beeinflussen. Hierbei wird nicht nur das Bild selbst, sondern auch der dafür zuständige Screen beeinflusst. Wechselt man beispielsweise von Lores nach Hires, so zeigt sich der Screen nur noch in halber Breite, was das Bild natürlich dementsprechend staucht. Wer allerdings nur das Bild selbst umformatieren möchte, kann dies auch tun. Hierfür existieren Gadgets wie »Kleiner«, »Breiter«, »Länger« oder »Kürzer«. Mit diesen Schaltern kann man das Bild wahlweise um die Faktoren »2« oder »3« in X- oder Y-Richtung stauchen oder dehnen. Wird auf diese Weise der

Bildinhalt verkleinert, läßt sich zusätzlich festlegen, ob jeder gerade oder ungerade Pixel ausgelassen werden soll. Es kann der Durchschnitt zweier benachbarter Pixel berechnet werden - für das ganze Bild oder jede Grundfarbe einzeln. Letztlich gibt es noch zwei weitere Gadgets, mit denen man das Bild an der X- oder Y-Achse spiegeln kann.



Nach all diesen Features, die wir bisher kennengelernt haben, ist »Pixmate« noch lange nicht ausgereizt. Mit der Menüwahl »Effekte/Bildverarb«, was soviel wie Bildverarbeitung heißt, gelangen wir direkt ins Herz des Programms. Es öff-

Bildverarbeitung leichtgemacht

net sich wiederum ein Fenster, was mit seinen zahlreichen Gadgets und Schieberegler eine Schaltzentrale gleichkommt. Hier lassen sich, laut Handbuch, »tausende« verschiedener Spezialeffekte erstellen. Die Operationen dieses »Bild-Prozessors« kann man in drei Kategorien einteilen: logische, Pixel- und Matrix-Operationen.

1. Logische Operationen wirken auf die gesamte Darstellung und behandeln alle Pixel gleich.

2. Pixeloperationen durchforsten das Bild Pixel für Pixel, führen ihre Berechnungen aus und ersetzen, basierend auf den Rechnungen, bestimmte Pixel.

3. Matrixoperationen entsprechen den Pixeloperationen und erlauben eine differenziertere Darstellung einiger Details. Zusätzlich kann der »Bild-Prozessor« Bilddaten aus zwei verschiedenen Quellen miteinander

kombinieren. Dafür kann man sich vorher ein zweites Bild auf einen extra Screen laden oder eine Kopie von dem ersten anlegen. Das zweite Bild erstellen Sie sich im Menüpunkt »Edit/Kopie —> Alt« oder mit der C-Taste. Wenn Sie nun mit dem Mauszeiger den »Pixmate«-Bildschirm herunterziehen, sehen Sie die Workbench. Ziehen Sie nun auch diesen

Screen herunter, erscheint die Kopie des Bildes. »Kopie z.« steht auch in der Menüleiste. Daraufhin haben Sie zwei »Pixmate«-Bildschirme, die Sie unabhängig voneinander bearbeiten können. Um eine der Logik-Operationen durchzuführen, benötigt man ein solches zweites Bild, da diese dann, gemäß der Logik-Gesetze, miteinander verbunden werden. Diese Verbindungen lassen sich nach verschiedenen Gesichtspunkten auswählen. Für jeden Punkt steht ein eigenes Gadget zur Verfügung. Leuchtet das Feld »NOP« auf, wird keine Operation (No Operation) durchgeführt - das Bild bleibt unverändert. Aber nun zu den anderen Möglichkeiten.

Zum ersten wäre da das »logische AND« (und). Hier werden alle Bildpunkte, die in den beiden Bildern gleich sind, erhalten, alle anderen gelöscht. Im logischen OR-Modus (oder) kann eine Auswahl zwischen den verarbeiteten Pixeln des Quell-



Bild 3 bis 5. Original und Fälschung: nachträgliche Farbabstimmung mit »Pixmate«

bildes (Bild 6) und denen des aktuellen Bildes getroffen werden. Anschließend vermischt man diese miteinander. Die SUB-Funktion führt eine logische Subtraktion zwischen schon verarbeiteten Daten des Quellbildes und dem Originalbild durch. Mit XOR wird ein »Exklusiv Oder« zwischen den beiden Bildern durchgeführt. Bei einem einzelnen Bild läßt sich hier auf einfache Weise ein Negativ-Bild erstellen. Bleibt abschließend noch das NOT-Symbol, mit dem alle eben gezeigten Operationen ins Gegenteil umgekehrt werden können.

Weitaus interessanter sind sicherlich die Pixeloperationen. Hier können in Kombination mit den Logikfunktionen, den Schieberegeln oder eines zweiten Bildes, unzählige Manipulationen durchgeführt werden. Haben Sie sich schon einmal über die vielen »Brösel« eines digitalisier-

ten Bildes geärgert? Mit der DST-Funktion (Durchschnitt) gehören diese Probleme der Vergangenheit an. DST errechnet den durchschnittlichen Farbwert eines 3 x 3 Pixel-Bereichs. Diese neun Farbwerte werden addiert und durch neun geteilt. Der errechnete Wert wird auf den mittleren Pixel dieses Bereichs abgegeben. Anschließend geht »Pixmate« zum nächsten Pixel über und wiederholt die Prozedur, bis alle Pixel abgearbeitet sind. Das Ergebnis ist ein klares, unverbröseltes Bild. Falls es in einigen Fällen etwas verwaschen wirken sollte, kann man dies mit Hilfe der Matrix-Operationen sowie den Schieberegeln exakter einstellen (Bild 7).

Mit »ZUF« (Zufall) werden, wie bei DST auch, 3 x 3 Pixel-Bereiche herangezogen. Anstatt einen Durchschnittswert zu berechnen, wird hier ein zufälliger Farbwert gesetzt. Mit

dieser Methode behandelte Bilder erscheinen wie durch eine geriffelte Glasscheibe (Bild 8).

Eine andere Art und Weise, Ungeheimheiten oder kleinere Fehler aus einem Bild zu bekommen, findet sich in den Funktionen »MW1« und »MW2« (Mittelwert) wieder. Statt der Durchschnittsberechnung wird hier eine Mittelwertberechnung durchgeführt, was zwar etwas mehr Zeit benötigt, aber auch für bessere Ergebnisse als DST sorgt. MW1 verwendet hierzu einen 3 x 3-Pixelbereich, während MW2 ein 5 x 5-Pixelfeld zur Berechnung heranzieht.

Eine ganz andere Möglichkeit eröffnet die USM-Funktion. Man wird es kaum glauben, aber es geht tatsächlich: Oftmals ärgert man sich über unscharfe Fotos seiner Urlaubsreise. Und nach den Gesetzen der Optik sind diese auch nie mehr scharf zu bekommen. Im Computer ist dies aber allemal möglich. Durch eine ausgeklügelte Rechnerei (sie nennt sich »Laplacesche Korrelation«) werden unter Zuhilfenahme einer 5 x 5-Matrix die unscharfen Bildkomponenten einfach herausgerechnet. Das Ergebnis ist immer wieder verblüffend.

Noch sind wir aber nicht am Ende. Die LKA-Funktion untersucht ein 25 x 25-Feld auf seine durchschnittliche Helligkeit. Sie ähnelt der USM-Funktion und hat die Aufgabe, kleine Details eines Bildes besser hervorzuheben. Ganz anders ist die BIN-Funktion. Hier wird jedes Pixel einem binären Entscheidungsprozeß unterworfen. Nach Einstellung eines Grenzwertes werden alle Pixel der ersten oder letzten Farbe der Farbpalette zugeordnet. Auf diese Weise lassen sich zweifarbige Bilder produzieren. Der letzte Operand, UNT, verhält sich wie BIN. Der Unterschied liegt darin, daß UNT alle Pixel, die einen höheren Farbwert als die Grenzwerteinstellung haben, bei der Operation unverändert bleiben, während Pixel, die unter den Grenzwert fallen, auf den Farbwert 0 (= schwarz) gesetzt werden. Diese Funktion ist sinnvoll, um unerwünschte Hintergrunddetails zu eliminieren.

Als letztes sind die Matrixoperationen an der Reihe. Hier stehen zwei verschiedene Arten zur Verfügung. Mit »LAP« werden die Umriss- und

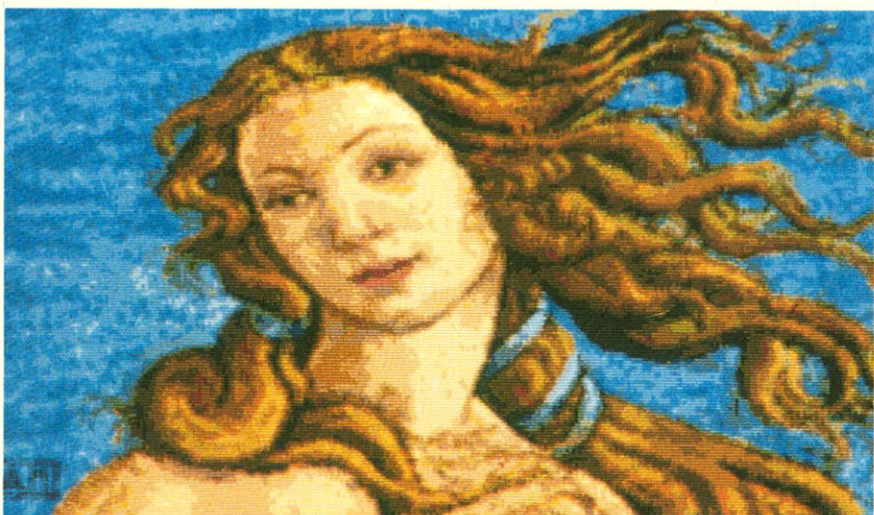


Bild 6 bis 8. Mit Hilfe des Bild-Prozessors eröffnen sich ungeahnte Nachbearbeitungsmöglichkeiten

mit »BGZ« die Ecken eines Bildes hervorgehoben oder aufgedeckt. Zur Anwendung kommt das aus der USM-Funktion benutzte Bildverschärfungsverfahren. Damit lassen sich bestimmte Details aus dem Bild selektiv hervorheben. Die in einer länglichen Gadgetbox angeordneten sieben Kernel-Funktionen

(*,+X,—,/,) sind dafür verantwortlich, welche Art von Umrissen dem Prozeß am stärksten unterworfen werden. Während einer solchen Matrix-Operation wird übrigens die aktuelle Einstellung bei den Logik-Operationen mitverwendet.

Nun aber zu den Schiebereglern, die alle Pixel- und Matrixfunktionen des Bild-Prozessors beeinflussen. Der »Min«-Regler dient der Einstellung der Empfindlichkeit, mit der die Operation durchgeführt wird. Bei einer Einstellung von »0« werden alle Bildschirmpunkte zur Berechnung verwendet. Bei höheren Werten schließt das Programm immer mehr Pixel aus der Berechnung aus. Interessant ist der »Max«-Regler. Bei maximaler Einstellung von »16« werden alle Neuberechneten Bildpunkte zur Darstellung herangezogen. Durch Verminderung dieses Wertes setzt »Pixmate« nur neue Pixel, wenn der Farbunterschied des alten Pixels zum neu berechneten einen gewissen Wert überschreitet. Der Dritte im Bunde gibt an, wieviel Prozent des Bildes bearbeitet werden sollen. Bei einer Einstellung von beispielsweise 50 Prozent berechnet das Grafikprogramm nur jeden zweiten Pixel neu. Dadurch können die Funktionen des Bildprozessors in ihrer Auswirkung stufenlos abgeschwächt werden.

Nach so viel verschiedenen Möglichkeiten der Bildverarbeitung kommt man sich sicher richtig erschlagen vor. Trotzdem hat »Pixmate« immer noch weitere Funktionen zu bieten. Beispielsweise lassen sich die Rot-, Grün- und Blauanteile eines Bildes separat auf Diskette abspeichern. Auch Cyan, Magenta oder Gelb lassen sich extrahieren. Oder wollen Sie die Komplementärfarben eines Bildes anzeigen? Auch das ist kein Problem.

Trotzdem wollen wir nun lieber die Bremse ziehen, da man über die Bildverarbeitung das ganze Sonderheft füllen könnte. Wer trotzdem noch nicht genug hat, der sei auf einschlägige Literatur verwiesen: Dawson, Benjamin; »Introduction to Image Processing«, Peterborough, NH BYTE (McGraw-Hill), März 1987. ag

Aufschlußreiche Ansichten



Steigen Sie ein in das Innenleben Ihrer Disketten und inspizieren oder manipulieren Sie einzelne Bytes. Unser Workshop und das hilfreiche Shareware-Programm »DisKey« helfen Ihnen, ein Disketten-Profi zu werden.

von Angela Schmidt und
Dr. Rudolf Egg

Der Diskettenmonitor »DisKey« ist als Shareware-Programm erhältlich. Die aktuelle Version 2.0 befindet sich z.B. auf der Diskette »Share 5«. Wer eine ältere Version besitzt, muß allerdings damit rechnen, daß nicht alle in diesem Workshop erklärten Funktionen vorhanden sind. Die Bezugsadressen finden Sie am Ende des Beitrags.

Zu Beginn erläutern wir das Grundprinzip und die Arbeitsvorgänge von »DisKey«, ein zweiter, systematischer Teil stellt alle Einzelfunktionen Schritt für Schritt vor. Danach

folgt eine kurze Übersicht des Filesystems des Amiga, damit Sie bei Ihrer Arbeit mit dem Diskettenmonitor über die einzelnen Blocktypen Bescheid wissen. Den Abschluß bilden Hinweise auf die einzelnen Optionen der Konfigurationsdatei von »DisKey«, mit der Sie wichtige Voreinstellungen definieren.

Der Diskettenmonitor ist in der Lage, alle Bytes einer Diskette sektorweise anzuzeigen und zu verändern. Im Gegensatz dazu stehen Filemonitore wie »NewZAP«, die innerhalb eines Programms spezifische Änderungen vornehmen können. Mit einem Filemonitor lassen sich beispielsweise englische Programmtext-

te ins Deutsche übertragen, die vorgegebenen Fenstergrößen einer Anwendung ändern oder sonstige Bytefolgen manipulieren. »DisKey« ist dazu nicht gedacht. Es ist zwar auch mit »DisKey« möglich, Programme zu modifizieren, dies geht aber längst nicht so komfortabel vorstatten wie mit einem File-Monitor. Die Stärken eines Diskettenmonitors liegen eher auf anderen Gebieten.

Eine Amiga-Diskette ist in 1760 einzelne Einheiten aufgeteilt, die »Block« genannt werden und von 0 bis 1759 durchnummeriert sind. Mit »DisKey« kann man sich jeden dieser Blöcke ansehen, ihn verändern, verlorengegangene Daten und Files aufspüren, Schreib-Lese-Fehler beseitigen und defekte Tracks reparieren – zumindest teilweise, manchmal sogar ganz. Dies funktioniert selbst dann, wenn sich die Diskette wegen eines Fehlers auf der Magnetoberfläche mit den üblichen DOS-Routinen nicht mehr lesen läßt.

Um beispielsweise eine Kickstart-Diskette zu patchen (verändern), braucht man einen Diskettenmonitor, denn File-Monitore sind dazu nicht in der Lage.

Der Programmstart – es geht los

Sie starten »DisKey« von der Workbench mit einem Doppelklick auf das Programm-Icon. Die beiden folgenden Info-Fenster werden mit einem Mausklick geschlossen. Anschließend erscheint ein Fenster, das in mehrere Abschnitte unterteilt ist:

In der oberen Hälfte befinden sich acht lange, schmale Ausschnitte, das HEX-Display. Weiter unten folgen zwei größere Fenster, in denen die entsprechenden ASCII-Werte dargestellt werden. Diese auf den ersten Blick etwas unübersichtliche Darstellung ermöglicht das gleichzeitige Anzeigen der ASCII- und HEX-Werte eines ganzen Blocks. Die Darstellung wird verständlicher, wenn man sich das rechte ASCII-Display direkt unter dem linken vorstellt. Zwischen diesen beiden ASCII-Fenstern findet man ein Fenster mit zwölf Befehlswörtern. Diese Funktionen werden durch Anklicken aktiviert, sind aber auch durch Kurzbefehle (Shortcuts) über Tastatur oder Pull-Down-Menüs anwählbar. Links un-

ten auf dem Bildschirm gibt es schließlich noch Einstellmöglichkeiten für den zu bearbeitenden Block, das Laufwerk etc. Im gegenüberliegenden rechten Kasten erhalten Sie Informationen über die momentan angewählte Diskette. Bevor wir uns mit diesem Fenster rechts unten näher befassen, noch ein kleiner Hinweis:

Da das Programm über sehr umfangreiche Tastenkombinationen verfügt, werden wir diese wie folgt darstellen:
LAMIGA = linke AMIGA-Taste
RAMIGA = rechte AMIGA-Taste
LSHIFT = linke SHIFT-Taste
RSHIFT = rechte SHIFT-Taste
So bedeutet beispielsweise <LAMIGA RSHIFT w>, daß Sie die linke AMIGA-Taste, die rechte SHIFT-Taste und die W-Taste gleichzeitig drücken müssen.

Aber nun zurück zu unserem Bildschirm und dem Informationsfenster (rechts unten). In der obersten Zeile erscheinen Meldungen des Programms, die den Anwender über aktuelle Vorgänge informieren. Das sind beispielsweise Fehlermeldungen, Aufforderungen zu einer Eingabetätigkeit und ähnliches.

Die unterste Zeile enthält den Namen der Diskette, die gerade bearbeitet wird. Alle Befehle haben nur auf diese Diskette Wirkung. In der mittleren Zeile findet man mehrere Informationen:

Die ersten drei oder vier Buchstaben weisen auf den Diskettentyp hin, wobei vier Möglichkeiten bestehen:

DOS: Eine normale DOS-Diskette, z.B. die Workbench-Diskette, eine PD-Diskette, eine Datendiskette usw.

FFS: Eine Diskette mit Fastfilessystem. In der Shareware-Version ist das aber die Ausnahme.

KICK: Es wurde eine Kickstart-Diskette eingelegt.

????: »DisKey« kennt diesen Diskettentyp nicht.

Der nächste Eintrag gibt an, ob der aktuell bearbeitete Block frei oder benutzt ist:

FREI: Der Block ist frei. Änderungen in diesem Block ziehen keine Schreib-Lese-Fehler nach sich, sind aber sinnlos, da das DOS diese Blöcke nach Belieben beschreiben darf.

USED: Es wurde ein benutzter Block geladen. Änderungen in diesem Block ziehen meist irgendwelche Folgen nach sich. Es heißt also aufpassen.

KICK: Bei Kickstart-Disketten sind Block 1 bis 512 benutzt. Das Programm macht hierüber aber keine speziellen Aussagen. Die aktuelle »DisKey«-Version 2.0 berücksichtigt Kickstart-Disketten bis zur Versionsnummer 1.3. Ab Kickstart 1.4 könnte es möglicherweise zu Problemen kommen, wenn dort mehr oder andere Blöcke benützt werden sollten. Außerdem gilt, daß »DisKey« nur mit Devices (Disketten, Platten) mit 512 Byte großen Blöcken arbeitet. Auch das könnte bei späteren Kickstart-Versionen eine gewisse Einschränkung bedeuten.

????: Es handelt sich um einen unbekanntem Diskettentyp.

Die folgenden Angaben dieser Anzeige informieren Sie über den gerade geladenen Blocktyp des einfachen Filesystems oder des Fastfile-systems:

ROOT: Es handelt sich um den Rootblock (das Stammverzeichnis) der Diskette, normalerweise Block 880. Dort findet man neben dem Diskettenamen auch Informationen darüber, wo sich bestimmte Dateien und Unterverzeichnisse befinden.

BOOT: Einer der Bootblöcke der Diskette, normalerweise Block 0 oder 1, ist gerade geladen. Bei bootfähigen Disketten steht dort ein kurzes Assemblerprogramm, das sowohl Install-Bootblock als auch Bootintros, aber auch ein Virus sein kann.

HEADER: Jede Datei, die sich auf der Diskette befindet, hat einen Namen, ein Erstellungsdatum, Protect-Flags und evtl. einen Kommentar. Mit dem List-Befehl von Amiga-DOS werden diese Einträge angezeigt. Das Filesystem speichert die Informationen sowie Zeiger auf weitere zum File gehörenden Blöcke im Header-Block ab.

Stoff für Ihren Amiga



HiSoft-Basic-Compiler

Mit HiSoft Basic gibt es endlich auch einen Basic-Compiler für den Amiga. Der interaktive Editier-, Kompilier- und Laufzeitzyklus entspricht dem eines Interpreters.

Der integrierte Editor erlaubt eine komfortable Eingabe. HiSoft Basic unterstützt die Eigenschaften des Amiga mit Fenstern, Grafik-Kommandos, Sprite-Handling und Maschinenzugriffen während des Gebrauchs von Bibliotheken. Es ist voll kompatibel mit Amiga-Basic, andere Standarddialekte für den PC, wie z.B. Microsoft QuickBasic, und den Atari ST können mit geringen Modifikationen kompiliert werden. Rekursive Unterprogramme und Funktionen sind möglich. Eine Anzahl strukturierter Ausdrücke wie z.B.

WHILE...WEND, DO...LOOP UNTIL und SELECT...CASE lassen Sie jede Programmierhürde meistern. Die Größe von Variablen ist nicht beschränkt. Aussagekräftige Fehlermeldungen und Korrekturmöglichkeiten tragen zur komfortablen Handhabung bei.

Hardware-Anforderungen:
Amiga 500, 1000 oder 2000 mit Kickstart 1.2 oder höher. 3,5"-Diskette, Bestell-Nr. 54127
DM 179,-*
(sFr 161,- /öS 1790,-*)

Devpac-Assembler 2.0
Devpac 2.0 ist ein Entwicklungspaket für den Amiga mit komfortablem Editor/Assembler, symbolischem Debugger und Linker zum Einbinden von Hochsprache-Modulen.
GenAm ist ein 68000er-Makroassembler mit integriertem Bildschirmditor, der bis zu 75.000 Zeilen pro Minute

assemblieren kann. Der 2-Paß-Assembler erzeugt sowohl linkbaren als auch direkt ausführbaren Code. Er unterstützt lokale Labels, die Signifikanz beträgt bis zu 127 Zeichen. Makros können bis zu 36 Parameter beinhalten und – Rekursion inbegriffen – so tief verschachtelt werden, wie Speicherplatz vorhanden ist. MonAm, der Debugger, erlaubt das Setzen von Breakpoints, das Disassemblieren auf Diskette und noch weitere Features, die das Debuggen zum Vergnügen machen.

Hardware-Anforderungen:
Amiga 500, 1000 oder 2000 mit mindestens 512 Kbyte, ein Diskettenlaufwerk. 3,5"-Diskette, Bestell-Nr. 54131
DM 149,-*
(sFr 135,-*/öS 1490,-*)

Devpac-Assembler 2.0 im Test! Amiga-Magazin 4/89: Eines der besten Programme seiner Art für den Amiga.

SuperED C

Multitasking-fähiger Editor als Programmierumgebung für den Aztec-C-Compiler (V 3.6). Bestell-Nr. 54139
DM 39,-*
(sFr 35,-*/öS 390,-*)

* Unverbindliche Preisempfehlung

Markt & Technik-Bücher und -Software erhalten Sie bei Ihrem Buchhändler, in Computerfachgeschäften oder in den Fachabteilungen der Warenhäuser.


Markt & Technik
Zeitschriften · Bücher
Software · Schulung

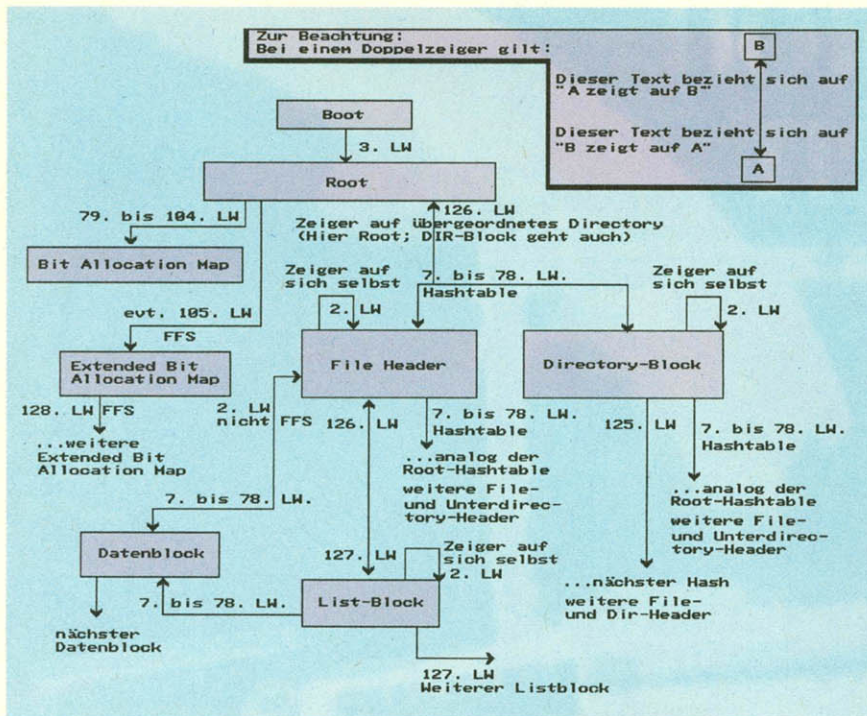


Bild 1. So ist das File-System der Amiga-Disketten aufgebaut

FILE-LIST: Wenn eine Datei so groß ist, daß nicht alle Datenblöcke im Header-Block vermerkt werden können, legt das Filesystem einen oder mehrere File-List-Blöcke an, die die restlichen Datenblöcke archivieren.

DIR: Ein Directory-Block, der – wie der Header-Block – Informationen über den Verzeichnisnamen, den Kommentar, das Datum und die Protect-Flags enthält. Zusätzlich ist hier unter anderem eingetragen, welche Files und Unterverzeichnisse dem Verzeichnis angehören.

DATA: Im Datenblock findet man die eigentlichen Daten eines Files. Will man ein Programm ändern, so ist das nur dann möglich, wenn man sich in einem Datenblock befindet.

BAM: Der Bit-Allocation-Map-Block einer Diskette gibt Aufschluß darüber, welche Blöcke der Diskette frei und welche belegt sind. Meist findet man den BAM-Block in der Nähe des Rootblocks. Seine Position ändert sich häufig.

BEXT: Dieser Block tritt nicht bei Disketten auf, sondern ist nur unter dem Fastfilessystem möglich. Wenn die normalen BAM-Blöcke nicht ausrei-

chen, um die freien und belegten Blöcke einer Festplatte zu archivieren, muß das Fastfilessystem noch erweiterte Bit-Allocation-Map-Blöcke erstellen. Diese haben einen etwas anderen Aufbau als die normalen Bit-Allocation-Map-Blöcke.

Köpfe, Spuren und Blöcke

Bisher wissen wir nur, daß eine Diskette in einzelne Blöcke unterteilt ist, aber nicht, wie diese auf der Diskette angeordnet sind. In dem Bildschirm-Feld links unten lassen sich verschiedene Sachen einstellen. Neben dem zu bearbeitenden Laufwerk, das man aber komfortabler über die Menüleiste anwählt, werden dort Block, Track, Zylinder, Head und Sektor angegeben. Dies erscheint auf den ersten Blick vielleicht verwirrend, jedoch sind die Zusammenhänge leicht zu verstehen.

Eine Diskette besteht aus 160 Tracks, die von 0 bis 159 durchnummeriert sind. Die Tracks findet man auf der Diskette immer abwechselnd auf der Ober- und Unterseite. Zwei derartige Tracks, die sich gewissermaßen gegenüberliegen, definieren einen Zylinder. Es gibt also 80 Zylinder auf der Diskette, die von 0 bis 79 durchnummeriert sind. Head bezeichnet die Seite der Diskette, auf der man sich gerade befindet. Beispiele:

Zylinder 0, Head 0 = Track 0
 Zylinder 0, Head 1 = Track 1
 Zylinder 1, Head 0 = Track 2

Werden Daten von einer Diskette gelesen werden, wird aus technischen Gründen immer mindestens ein ganzer Track eingelesen. Intern wird der Track in elf Abschnitte, die Sektoren, aufgeteilt (Nummer 0 bis 10). Wenn man zusammenrechnet, kommt man auf:

$160 \text{ Tracks} \times 11 \text{ Sektoren} = 1760 \text{ Blöcke}$

oder

$2 \text{ Heads} \times 80 \text{ Zylinder} \times 11 \text{ Sektoren} = 1760 \text{ Blöcke.}$

Die Speicherkapazität einer Diskette läßt sich folgendermaßen berechnen:

$1760 \text{ Blöcke} \times 512 \text{ Byte} = 901120 \text{ Byte}$

oder

880 KByte

Die Blöcke werden so durchnummeriert, daß Track »n«, Sektor »m« dem Block »n*11+m« entspricht.

Klickt man auf »BLOCK:«, so verschwindet die Zahl dahinter, und es wird die Eingabe einer Blocknummer erwartet, die mit <RETURN> abgeschlossen wird. <ESC> veranlaßt »DisKey«, die Eingabe abzubrechen. Bereits geänderte Daten, die nach der Eingabe eines neuen Blocks unwiederbringlich verloren wären, bleiben dadurch erhalten.

Eine andere Möglichkeit, den Block zu verändern, ist ein Klick auf »+« oder »-«. Aber auch mit der Tastatur lassen sich diese Funktionen starten:

<LAMIGA b> = BLOCK:

<LAMIGA LSHIFT b> = BLOCK -

<LAMIGA RSHIFT b> = BLOCK +

Gibt man einen neuen Block ein, ändern sich die anderen Werte im Fenster links unten, der bezeichnete Block wird geladen und im ASCII- und HEX-Display dargestellt. Das gleiche können Sie auch mit »TRACK:«, »HEAD:« usw. anstellen. Dann würde der Tastenbefehl für »TRACK« z.B. <LAMIGA LSHIFT t>

AMIGA
SONDERHEFT

PROGRAMM-SERVICE

Direkt bestellen statt abtippen!
Die aktuelle Diskette zum Sonderheft 13:

»Disk-Utility«

Ein nützliches Utility für alle, die mit CLI-Befehlen auf Kriegsfuß stehen. Strukturieren, Formatieren und Kopieren ganzer Disketten und einzelner Files wird zum Kinderspiel. Selbst das Starten eines Programms von dieser Oberfläche ist möglich. Beschreibung auf Seite 95.

»BiPF«

Der Grafikräuber, mit dem Sie alle Bilder aus Programmen stehlen können. »BiPF« findet die Bitplanes

eines Programms und speichert die Bitmap als IFF-Bild. Beschreibung auf Seite 84.

»PassWd«

Es ist manchmal etwas schwer, wichtige Disketten oder die Festplatte und damit die darauf befindlichen Daten vor unbefugten Zugriff zu schützen.

Mit »PassWd« erhalten Sie ein Werkzeug, mit dem Sie spielend Ihre Geheimnisse behalten. Die Beschreibung steht ab Seite 105.

»DPaint III - DVideo«

Begleitend zu unseren Tips und Tricks zum Datentransfer zwischen DPaint III und DVideo, finden Sie auf der Programm-Service-Diskette alle Bilder, die in diesem Beitrag erarbeitet werden. Zusätzlich geben wir Ihnen ein Animations-Beispiel, das ebenfalls in dem Artikel erstellt wird. Der Datentransfer beginnt ab Seite 78.

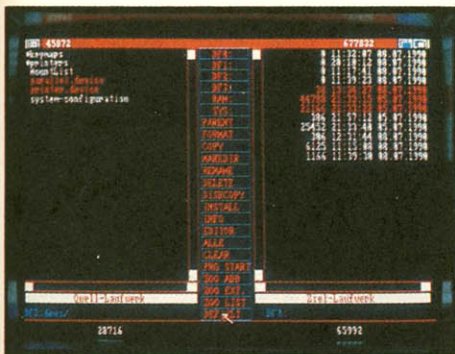
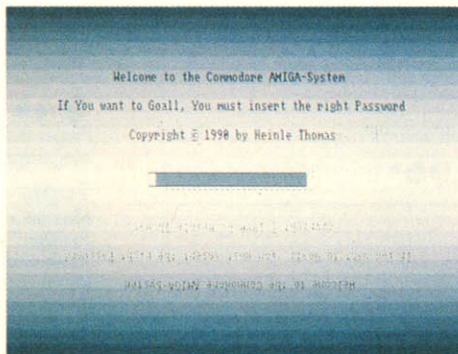
Zwei Disketten !

Bestell-Nr. 45013

DM 29,90*

sFr 26,90*/öS 299,-*

* Unverbindliche Preisempfehlung




Markt & Technik
Zeitschriften · Bücher
Software · Schulung

**Weitere Angebote
auf der Rückseite!**

AMIGA PROGRAMMSERVICE

Bit für Bit nur Hits... ...das Beste aus zwei Jahren Amiga-Magazin!

Wir haben auf zwei Disketten die interessantesten Programmangebote der Jahre 1988 und 1989 für die Bereiche Anwendungen/Tools und Spiele gesammelt. Mit dieser Sammlung legen Sie einen leistungsfähigen Grundstein oder bilden eine sinnvolle Erweiterung für Ihre Programmbibliothek. Und dies alles zum besonders günstigen Paketpreis.

DISK I: Anwendungen/Tools

AmigaDat - Die Dateiverwaltung für alle Zwecke, von der Schallplattensammlung über Adressen bis hin zur Videosammlung.

Manager - Das komfortable Haushaltsbuch.

Disketi - Drucken von Diskettenlabels.

MasterCruncher - Leistungsfähige Daten- und Programmkomprimierung.

Recover - Retten von gelöschten Daten.

Resi - Macht Programme resistent.

MouseCreator und PointerMaker - Generieren Sie Ihre eigenen Mauszeiger.

DiskSpy - Problemloses Ändern von Daten direkt auf Diskette.

AmigaSort - Bringt Ordnung in Ihre Diskettensammlung.

Fade - Einfaches Ein- und Ausblenden von Bildschirmen ermöglicht tolle Effekte.

VirusControl - Der komfortable Virenschutz.

TrapHandle - Die Möglichkeit bei auftretenden Fehlern Ihr C-Programm sauber zu beenden.

DISK II: Die Spielesammlung, die man haben muß

Action

Troof - Das spannende Motorradrennen der Zukunft.

Quadron - Geschicklichkeit bei höchsten Geschwindigkeiten sind gefragt.

Spieleumsetzungen

Poker - Wann bekommen Sie den Royal-Flush.

Billard - Tolle Grafik erwartet Sie bei dieser fantastischen Umsetzung.

Domino - Verblüffende Umsetzung des bekannten Spiels.

Kniffel - Eine grafisch verblüffende Würfelspielumsetzung für bis zu vier Spieler.

3D-Tic-Tac-Toe - Dreidimensionales Spiel zum Kombinieren und Denken.

Best of the Rest

Eliza - Der Amiga als Psychotherapeut.

Arriba - Die Tastatur lernt sprechen.

Die Beschreibungen der Programme sind als Readme-Datei auf den jeweiligen Disketten.

Bestell-Nr. 47901

Zwei Disketten randvoll mit tollen Programmen zum Paketpreis von:

DM 39,90* sFr 35,90*/ öS 390,-*

Amiga-Sonderheft 10

Golden Paint Pot: Dieses Malprogramm der Superlative braucht den Vergleich zu professionellen Programmen nicht zu scheuen. Lassen Sie sich von der Funktionsvielfalt und der durchdachten Benutzerführung überraschen und beeindrucken. **Kalaha:** Diese 3000 Jahre alte Spielidee hat bis heute nichts von ihrem Reiz verloren. Mit seinen einfachen Regeln und trotzdem komplexen Spielbau kommt keine Langeweile auf. Ein Muß für jede Spielesammlung. **Der File-Requester:** Dieser File-Requester, für höchste Ansprüche, gibt Ihren Basic-Programmen eine noch professionellere Note. **Basic-Kurs:** Tippen Sie nicht gerne ab? Wollen Sie trotzdem nicht auf die lehrreichen Beispiel-Listings des Basic-Kurses aus diesem Heft verzichten? Auf der Programm-Service-Diskette haben Sie alle 30 Listings sofort zur Hand. Weiterhin befinden sich auf der Diskette alle Programme, die im Inhaltsverzeichnis des Amiga-Sonderhefts 10 mit einem Diskettensymbol gekennzeichnet sind.

Bestell-Nr. 45010 **DM 29,90*** sFr 26,90*/öS 299,-*

Amiga-Sonderheft 2 Grafik, Anwendung

Object-Editor: Animierte Figuren, beispielsweise für eigene Spiele, entwickeln Sie mit diesem Editor auf komfortable Weise. Sogar mit Deluxe Paint erstellte Pinsel lassen sich einlesen. **Haushaltsbuch:** Mit diesem hervorragenden Anwendungsprogramm verwalten Sie alle Einnahmen und Ausgaben auf übersichtliche Weise. Eine Monats- oder Jahresstatistik zeigt, in welchen Bereichen Sie zukünftig sparen können. Jetzt haben Sie Ihre Finanzen im Griff.

Keyboard-Master: Lernen Sie im Zehn-Finger-System zu tippen. Mit diesem didaktisch ausgereiften Programm ist dies kein Problem. Für Programmierer sind sogar Spezial-Lektionen mit wichtigen Sonderzeichen vorhanden.

FastLoadCopy: Dieses Tool bringt den DIR-Befehl auf Trab. Nach der »Operation« wird das Inhaltsverzeichnis einer Diskette im D-Zug-Tempo eingelesen. Zusätzlich kopiert das Programm Disketten und versieht diese mit dem schnellen Directory.

Weiterhin befinden sich auf der Diskette alle Programme, die im Inhaltsverzeichnis des Amiga-Sonderhefts 2 mit einem Diskettensymbol gekennzeichnet sind.

3,5"-Diskette für Amiga

Bestell-Nr. 45802 **DM 29,90*** sFr 26,90*/öS 299,-*

Amiga-Sonderheft 11

IconMaster: Ein nützliches Utility zum Erstellen eigener Piktogramme für die Workbench. Vorhandene Icons lassen sich verändern und IFF-Grafiken als Bildvorlagen nutzen. Ihrer Kreativität sind keine Grenzen gesetzt. **Mauser:** Der Einheits-Mauspfeil gehört der Vergangenheit an. Mit diesem Programm installieren Sie einen Pointer nach Ihren Wünschen. Dabei stehen Ihnen 15 Farben zur Verfügung. Damit noch etwas Bewegung ins Spiel kommt, läßt sich der neue Mauszeiger sogar noch animieren. **DPaint-III-Workshop:** Begleitend zu unserem großen DPaint-III-Workshop finden Sie auf der Programm-Service-Diskette alle Bilder, die in diesem Kurs erarbeitet werden. Zusätzlich geben wir Ihnen ein Animations-Beispiel, das ebenfalls in dem Artikel erstellt wird. **Deluxe Video 3.0:** Eine Ergänzung zu unserem Workshop über DPaint III stellt dieser Kurs dar. Auch zu diesem Artikel finden Sie alle erarbeiteten Beispiele auf der Programm-Service-Diskette. **Create-a-Shape:** Im Heft stellen wir Ihnen diesen Animationskünstler vor. Dabei wird Schritt für Schritt eine kleine Animation erstellt. Damit Sie auch gleich die einzelnen Schritte nachvollziehen können, finden Sie auf der Diskette alle Programme, die im Inhaltsverzeichnis des Amiga-Sonderhefts 11 mit einem Diskettensymbol gekennzeichnet sind.

Bestell-Nr. 45011 **DM 29,90*** sFr 26,90*/öS 299,-*

Drei Disketten!

Amiga-Sonderheft 9 Grafik

Amiga-Objekt-Editor: Überdimensionale Sprites spielend leicht entwickelt. **AOE-Convert:** bauen Sie die mit dem Amiga-Objekt-Editor entwickelten Super-Sprites in Ihre Basic-Programme ein. **IFF-Filter:** Farbbilder auf faszinierende Weise modifizieren. **Ray Tracer:** Arbeiten mit dem Spiel von Licht und Schatten. **3DAnim:** Gittergrafiken - bewegt, perspektivisch, dreidimensional.

Spiele

Flugsimulator - Ping-Pong - Krieg der Sterne - Denkspiel Trainingslager - Strategiespiel »Buh Au CS« - Börsenspiel

Tools

MasterCruncher: Packt alles. **Virus Killer:** Unentbehrlich. **TestDev:** Welche Devices sind vorhanden? **Weiches Ein- und Ausblenden** von Bildern. **Shuffle:** Der schnelle Weg zum hintersten Fenster. **Felder;** Beschleunigtes Abarbeiten von numerischen Feldern.

Anwendungen

Keyboard-Master: Lernprogramm für das 10-Finger-System. **Sound-Effekt-Editor:** Geräusche selbst erzeugen. **Etiketten:** Das Programm für Ihre Diskettenetiketten. **hscroll:** Laufschriften erzeugen. **Uni-Datei:** Universelle Dateiverwaltung mit komfortabler Benutzerführung.

Außerdem finden Sie alle Programme aus den Artikeln, die mit dem D-Symbol gekennzeichnet sind, auf den Programmservice-Disketten.

Bestell-Nr. 45009 **DM 29,90*** sFr 26,90*/öS 299,-*

Zwei Disketten!

* Unverbindliche Preisempfehlung

Wichtig: Mit den Gutscheinen aus dem »Super-Software-Scheckheft« zu DM 149,- können Sie Software-Disketten Ihrer Wahl aus dem Programmservice-Angebot im Wert von DM 180,- bestellen - egal, ob diese DM 19,90, DM 29,90 oder DM 89,- kosten. Sie sparen DM 30,-!

Das Super-Software-Angebot finden Sie in den Zeitschriften

Computer Persönlich, PC Magazin Plus, Amiga-Magazin, Amiga-Sonderheft, 64'er-Magazin, 64'er-Sonderheft, ST-Magazin, PC Magazin, Computer live.

Übrigens: Die Gutscheine können Sie auch übertragen oder verschenken!

Das Scheckheft können Sie per Verrechnungsscheck oder mit der eingeklebten Zahlkarte direkt beim Verlag bestellen.

Kennwort: »Super-Software-Scheckheft«, Bestell-Nr. W156

Sie suchen hilfreiche Utilities und professionelle Anwendungen für Ihren Computer? Sie wünschen sich gute Software zu vernünftigen Preisen? Hier finden Sie beides! Unser stetig wachsendes Sortiment enthält interessante Listing-Software für alle gängigen Computertypen. Jeden Monat erweitert sich unser aktuelles Angebot um eine weitere interessante Programmsammlung für jeweils einen Computertyp.

Bestellungen bitte nur gegen Vorauskasse an: Markt&Technik Verlag AG, Buch- und Software-Verlag, Hans-Pinsel-Straße 2, D-8013 Haar, Telefon (089) 4613-0. Schweiz: Markt&Technik Vertriebs AG, Kollerstrasse 37, CH-6300 Zug, Telefon (042) 440.550. Österreich: Markt&Technik Verlag Gesellschaft m.b.H., Große Neugasse 28, A-1040 Wien, Telefon (0222) 5871393-0; Microcomput-ique, E. Schiller, Göglstraße 17, A-3500 Krems, Telefon (02732) 74193; MES-Versand, Postfach 15, A-3485 Haitzendorf; Bücherzentrum Meidling, Schönbrunner Straße 261, A-1120 Wien, Telefon (0222) 833196. Bestellungen aus anderen Ländern bitte nur schriftlich an: Markt&Technik Verlag AG, Abt. Buchvertrieb, Hans-Pinsel-Straße 2, D-8013 Haar. Nur gegen Bezahlung der Rechnung im voraus.

Bitte kein Bargeld einschicken!

Bitte verwenden Sie für Ihre Bestellung und Überweisung die beigeheftete Postgiro-Zahlkarte, oder senden Sie uns einen Verrechnungsscheck mit Ihrer Bestellung. Sie erleichtern uns die Auftragsabwicklung, und dafür berechnen wir Ihnen keine Versandkosten.

lauten. Aber meist benötigt man nur die Blockangabe.

Um einzelne Bytes auf der Diskette zu ändern, bewegt man den Cursor im ASCII- oder HEX-Display und modifiziert die angewählten Werte

Der Editor

nach Wunsch. Der Cursor bewegt sich stets in den beiden Fenstern synchron. Für die Eingabe klickt man einfach das gewünschte Fenster an, das dann weiß umrahmt wird. Selbstverständlich ist eine Wahl zwischen ASCII- oder HEX-Änderungen per Tastatur ebenfalls möglich. Für ASCII-Änderungen ist die Taste <RAMIGA SPACE> zu drücken, für HEX entsprechend <LAMIGA SPACE>. Probieren Sie das einfach ein wenig aus und seien Sie beruhigt: Die Daten werden zunächst noch nicht auf Diskette übertragen. Überschriebene Bytes verfärben sich rot. So hat man immer einen guten Überblick, was an dem Block geändert wurde. Natürlich lassen sich diese Bytes wieder zurückholen. Dies geschieht durch das Drücken der SHIFT-Taste, während man mit den Cursortasten über das entsprechende Byte fährt.

Bevor Sie nun aber begeistert anfangen, Daten zu ändern und diese auf die Diskette schreiben, sei darauf hingewiesen, daß fast jeder Block über eine Prüfsumme verfügt, und wenn diese nicht angepaßt wird, erzeugt das AmigaDOS lästige Schreib-Lese-Fehler. »DisKey« ist jedoch in der Lage, diese Anpassung vorzunehmen. Mehr darüber erfahren Sie in der Beschreibung der einzelnen Funktionen.

Schreib-Lese-Fehler

Wenn Amiga-DOS einen Schreib-Lese-Fehler meldet, muß es sich nicht unbedingt um einen Fehler auf der Diskettenoberfläche handeln. Dies bedeutet, daß Diskettenmonitore – im Gegensatz zu File-Monitoren – in der Regel die entsprechenden Daten ohne Schwierigkeiten lesen können. In diesem Fall ist nur die Diskettenstruktur beeinträchtigt, die Art, wie die Dateien verwaltet werden.

Andere Fehler sind auf der Magnetschicht zu suchen und entstehen beispielsweise durch Schmutz. Solche Daten können von den meisten Diskettenmonitoren nicht mehr gelesen werden. In diesem Fall gibt »DisKey« im Statusfenster rechts unten eine rote Fehlermeldung aus. Wenn der Fehler beim Laden eines neuen Blocks eingetreten ist, versucht »DisKey« anschließend sein Glück mit einer eigenen Routine.

Die Daten, die dann auf dem Bildschirm erscheinen, können allerdings etwas verändert sein. Die Erfahrung hat gezeigt, daß manchmal nur ein einziges Bit unterschiedlich ist, oft auch gar keines. Da aus technischen Gründen stets ein gesamter Track gelesen wird, können unter AmigaDOS oftmals einige völlig fehlerfreie Blöcke nicht geladen werden, da die Betriebssystem-Routinen nur korrekte Tracks zulassen.

»DisKey« versucht dann auf eigene Faust, alle Blöcke des Tracks zu decodieren. Im Normalfall sind zwischen ein und drei Blöcke eines Tracks zerstört. Die restlichen acht

Keine Angst vor Viren

bis zehn Blöcke können jedoch korrekt decodiert werden. Wer nun aber versucht, diese Daten wieder auf dieselbe Stelle der Diskette zurückzuschreiben, wird Pech haben, denn auch hier weigert sich das DOS. Im Laufe unseres Workshops werden daher noch Möglichkeiten vorgestellt, diese Daten in eine Datei zu schreiben oder einen gesamten Track mit Hilfe der Funktion »REFORMAT« zu restaurieren.

Viren sind jedem Anfänger, aber auch Fortgeschrittenen, ein großer Schrecken. Der Lamer-Virus beispielsweise läßt sich nicht so einfach von einem Diskettenmonitor entlarven, selbst wenn man sich den Bootblock ansieht. Ist der Lamer einmal im Speicher aktiv, täuscht er einen normalen Install-Bootblock vor. In Wirklichkeit aber ist ein gefährlicher Lamer-Bootblock auf der Diskette, der diese von Zeit zu Zeit mit unerwünschten Daten füllt, oder sogar die Festplatte formatiert.

Um diesen Virus zu entlarven, wird der Bootblock einer Diskette von »DisKey« beim Einlegen zweimal gelesen: Das erste Mal ganz normal,

das heißt, »DisKey« erhält bei einem im Speicher aktiven Lamer-Virus einen Install-Bootblock. Das zweite Mal liest »DisKey« den Bootblock mit seiner eigenen »RAWREAD«-Routine und entschlüsselt diesen. Unterscheidet sich das Ergebnis vom ersten Leseversuch, so warnt »DisKey« mit der Meldung »TD_READ <> RAWREAD«, die auf einen Lamer hinweisen kann.

Durch einen Fehler, der aber zum Glück nur gelegentlich vorkommt und eventuell im Betriebssystem zu suchen ist, erscheint diese Meldung in einigen Fällen selbst dann, wenn gar kein Virus aktiv ist. Man sollte sich aber in jedem Fall einmal den Bootblock dieser Diskette ansehen, den »DisKey« von nun an nur noch mit »RAWREAD« liest.

Auch andere Viren werden beim Einlegen einer neuen Diskette in das aktive Laufwerk durch das Suchen einiger typischer Zeichenfolgen im Bootblock entlarvt. Entdeckt »DisKey« einen Virus oder virusverdächtigen Bootblock, wird automatisch anstatt des Rootblocks der Bootblock geladen und angezeigt.

Allerdings gibt es einige Bootblöcke, bei denen »DisKey« nicht genau feststellen kann, ob es sich dabei um einen Virus oder nur um einen harmlosen Lade-Bootblock handelt. Sicherheitshalber werden diese Bootblöcke – wie Viren – sofort angezeigt. Deswegen sollte man nicht sofort den Bootblock zerstören, sonst kann es passieren, daß dabei wichtige Daten überschrieben werden.

»DisKey« verfügt über eine große Anzahl verschiedener Funktionen, die, wie schon erwähnt, in dem Befehlsfenster in der unteren Bildschirmhälfte angezeigt und durch Anklicken aktiviert werden. Zunächst sieht man jedoch dort nur zwölf Befehlswörter. Drückt man die rechte Maustaste, erscheinen nochmals zwölf Befehle, und ein dritter Mausklick bringt weitere elf zum Vorschein.

Die Funktionen - Schritt für Schritt

Wer lieber mit Pull-Down-Menüs arbeitet, kommt auch nicht zu kurz. Die rechte Maustaste über der Menüleiste gedrückt, aktiviert sechs Menüs. Alle unten erläuterten Befehle lassen sich über diese Menüs anwählen und verfügen zusätzlich über Tastatur-Shortcuts (Kurzbefehle). »DisKey« unterscheidet dabei - im Gegensatz zu anderen Programmen - zwischen rechter und linker AMIGA-Taste sowie zwischen Groß- und Kleinschreibung. Beachten Sie dazu bitte die Schreibweise der einzelnen Tastenkombinationen (siehe oben). Innerhalb dieses Workshops wollen wir die zur Verfügung stehenden Befehle in der Reihenfolge abhandeln, wie sie in dem Befehlsfenster erscheinen. Hinter jedem Befehlswort folgt in spitzen Klammern die entsprechende Tastatur-Kurzform. Manche Befehle wurden mit einer speziellen »Kindersicherung« versehen. Will man einen Befehl anwählen, der derart geschützt ist, sollte man den Mausknopf lange gedrückt halten, oder man muß danach einen Requester beantworten. Bei Tastaturbedienung muß der Mausknopf natürlich nicht benützt werden, dafür müssen aber die jeweiligen Tasten länger gedrückt bleiben. Beobachten Sie dabei jeweils die Statuszeile rechts unten. Wenn dort »Alles klar!« erscheint, dürfen Sie den Mausknopf bzw. die Tasten wieder loslassen.

LESEN: < RAMIGA l >

Der Block, den Sie gerade bearbeiten, wird nochmals eingelesen. Dies ist dann sinnvoll, wenn Sie noch nicht gespeicherte Änderungen wieder rückgängig machen möchten. Zwar ist das auch mit <SHIFT Cursortaste> möglich, jedoch ist ein Klick auf »LESEN« viel komfortabler.

SCHREIBEN: < RAMIGA w > (Kindersicherung)

Der geänderte Block wird auf die Diskette geschrieben. Doch Vorsicht: DOS-Disketten verfügen über eine Prüfsumme (siehe »SUMME«),

die man zuvor auf alle Fälle anpassen muß, ansonsten erhält man als Quittung jede Menge Schreib-Lese-Fehler. Nach dem Schreiben ändern die zuvor roten Bytes abermals ihre Farbe und werden schwarz. In diesem Zustand lassen sie sich noch zurückholen. Wenn allerdings ein neuer Block geladen ist, sind die alten Daten unwiderruflich verloren.

DRUCKEN: < RAMIGA d >

Druckt den geladenen Block im »type opt h«-Format aus, also als ASCII- und HEX-Zeichen. Wer keinen Drucker hat, kann unter bestimmten Bedingungen diesen Menüpunkt benutzen (Einzelheiten siehe unten). Ist der Drucker ausgeschaltet, so ist ein Weiterarbeiten mit »DisKey« so lange nicht möglich, bis man den nach einiger Zeit erscheinenden DOS-Requester beantwortet oder den Drucker einschaltet.

MERKEN: < RAMIGA m >

Der geladene Block wird in einem Zwischenspeicher abgelegt, und kann später wieder zurückgeholt werden. So läßt sich beispielsweise ein veränderter Block kurzfristig ablegen, ohne ihn jedoch auf Diskette zu schreiben. Wenn man dann feststellt, daß die Änderungen, die man nach diesem Zeitpunkt vorgenommen hat, weniger gut waren, holt man sich einfach die vorhergehende Version wieder zurück.

Ein zweites Anwendungsgebiet ist das Kopieren von einzelnen Blöcken, auch auf andere Disketten. So können Sie mit dieser Funktion einen Bootblock von einer Diskette auf eine andere übertragen. Dazu muß man aber meist Block 0 und 1 kopieren, nicht nur Block 0.

HOLEN: < RAMIGA h >

Der mittels »MERKEN« gespeicherte Block wird eingeladen und überschreibt den gesamten Block des Speichers. Alle Bytes verfärben sich rot. Auf der Diskette selbst ist dadurch noch nichts geändert. Erst »SCHREIBEN« sichert den Block dauerhaft.

SUMME: < RAMIGA s >

Die Betriebssystem-Programmierer wollten verhindern, daß defekte Programme ausgeführt werden. Dazu haben sie fast alle benutzten Blöcke mit einer Prüfsumme verse-

hen. Nur wenn alle Prüfsummen einer Datei stimmen, wird diese problemlos geladen. Werden nun Änderungen in einem Block vorgenommen, stimmt die Prüfsumme dieses Blocks nicht mehr. Würde der Block so zurückgeschrieben, bekäme die Diskette einen Schreib-Lese-Fehler. Deshalb muß man vor dem Zurückschreiben die Prüfsumme anpassen. Dazu braucht man nichts weiter zu tun, als auf »SUMME« zu klicken. »DisKey« weiß dann automatisch, was zu tun ist, und trägt die Prüfsumme an der richtigen Stelle ein. Anschließend dürfen Sie den Block zurückschreiben. Auch die Bootblöcke verfügen über eine Prüfsumme. Nur dann, wenn die Prüfsumme über beide Bootblöcke stimmt, wird von der Diskette gebootet. Ist die Prüfsumme korrekt, jedoch kein vernünftiger Assembler-Code im Bootblock enthalten, können Sie beim Booten einen wunderschönen Guru erleben. Die Prüfsumme der Bootblöcke befindet sich im Block 0, muß aber angepaßt werden, wenn man in Block 1 etwas geändert hat. Selbst Kickstart-Disketten sind mit einer Prüfsumme versehen. Diese steht im Block 512. Wenn Sie auf einer Kickstart-Diskette irgendetwas in Block 1 bis 512 geän-

Sinnvolle Prüfsumme

dert haben, müssen Sie zum Schluß Block 512 anwählen und auf »SUMME« klicken. Tun Sie das nicht, kann ein Amiga 1000 zwar von der Kickstart-Diskette booten, würde aber beim nächsten RESET bemerken, daß die Prüfsumme der Kickstart-Diskette nicht stimmt, und diese dann erneut anfordern. Das mag zunächst sinnlos klingen. Bedenkt man aber, daß beim Amiga 1000 die Kickstart in einer Art RAM-Speicher steht, der theoretisch während des Arbeitens auch verändert werden kann, bekommt die Prüfsumme doch wieder einen Sinn, denn wer arbeitet schon gerne auf einem Amiga mit korruptem Betriebssystem. Um die Prüfsumme der Kickstart-Diskette zu berechnen, öffnet »DisKey« zunächst ein Status-Window und liest dann nacheinander Block 1 bis 512

ein. Wollen Sie diesen Vorgang abbrechen, klicken Sie einfach in das Status-Fenster.

SUCHEN: <RAMIGA S>

Mit diesem Menüpunkt durchsucht »DisKey« Teile der Diskette nach Zeichenketten, dies können HEX- oder ASCII-Zeichen sein. Zunächst wird gefragt, ob zwischen Groß- und Kleinschrift unterschieden werden soll. Dann wählen Sie zwischen ASCII- und HEX-Suchmodus und geben den entsprechenden String ein. Zum Schluß wird der Bereich, in dem gesucht werden soll, in Blöcken eingegeben. Wenn der gewünschte String gefunden wurde, man aber möchte, daß »DisKey« weitersucht, klickt man lange auf »SUCHEN«. Nach dem Erscheinen der Anzeige »Suche nach: Suchstring« wird der Mausknopf losgelassen. Das Programm setzt dann die Suche an der Stelle fort, wo der letzte String gefunden wurde.

FORMAT: <RAMIGA f>

»FORMAT« erlaubt Ihnen, einzelne Tracks zu formatieren. Diese werden dadurch mit Nullbytes überschrieben. Die Routine überprüft aber nicht, ob die Daten auch wirklich auf der Diskette angekommen sind, führt also kein Verify aus. Die danach erscheinende Errortable enthält nur in den seltensten Fällen fehlerhafte Tracks. Als Start- und Endtrack dürfen bei Disketten Werte zwischen 0 und 159 eingegeben werden. Eine Leereingabe beendet diese Routine vorzeitig. Wenn man allerdings den Rootblock (Track 80), den Bootblock (Track 0) oder die Bit-Allocation-Map (unterschiedlich) wegformatiert, muß man die Folgen tragen:

Das DOS wird spätestens nach dem nächsten Diskettenwechsel die Quittung in Form einer Fehlermeldung ausstellen. Vermeiden Sie das also, oder behandeln Sie die Diskette danach mit »INITDISK« (siehe unten).

REFORMAT: <RAMIGA r>

Hier gilt im wesentlichen, was schon zu »FORMAT« gesagt wurde. Allerdings weist »REFORMAT« doch einige Unterschiede auf, die diese Routine zu einer Funktion machen, die man nicht mehr missen möchte. Die mit »REFORMAT« behandelten Tracks werden grundsätzlich erst gelesen und anschließend mit den gelesenen Daten neu formatiert. Also ändert sich im Normalfall nichts an den vorhandenen Daten. Aber wozu dann das alles? Falls »REFORMAT« auf einen Fehler in der Magnetschicht stößt, mit dem das Betriebssystem nicht mehr zurechtkommt, also auf einen Track, der nicht mehr

Nützliche Funktion

gelesen werden kann, probiert »DisKey« noch einige Male sein Glück mit den üblichen DOS-Routinen, denn vielleicht handelt es sich nur um einen Staubpartikel, der sich irgendwann einen anderen Aufenthaltsort sucht. Falls das aber nicht klappt, entfaltet »DisKey« seine eigenen Fähigkeiten:

Er liest den Track mit »RAWREAD« und versucht dann, die Daten wiederherzustellen. Wenn »DisKey« glaubt, seine Arbeit bestmöglich verrichtet zu haben, formatiert es den Track und schreibt gleich beim Formatieren die decodierten Daten zurück. Dadurch lassen sich in vielen Fällen verloren geglaubte Daten wieder retten. Oft sind nur Kleinigkeiten auf der Diskette defekt, und manchmal liegen diese auch noch in Blöcken, die überhaupt nicht benutzt werden. Es wäre doch schade, wenn uns dadurch ein ganzer Track, also 11mal 512 Byte verlorengingen.

Damit sich diese Routine richtig entfalten kann, sollte man die defekte Diskette vorher unbedingt mit einem leistungsfähigen Kopierprogramm, z.B. A-Copier, auf eine neue Diskette kopieren, die keine Fehler in der Magnetschicht aufweist, und dann die gute Diskette entsprechend behandeln. Denn was nützt es, wenn »DisKey« die Daten erst umständlich zu restaurieren versucht und dann wegen eines Magnetschichtfehlers diesen Track nicht richtig auf die Diskette bekommt? Doch kommen wir nun noch kurz zu der Bedienung dieser Routine. Wie bei »FORMAT« wird

zunächst der Bereich angegeben, den Sie formatieren wollen. Anschließend folgt die Anzahl der Lese- und der Schreibversuche. Die Leseversuche beziehen sich auf die Anzahl der Versuche, die »DisKey« mit den üblichen Routinen vornehmen soll. Unter den Schreibversuchen versteht das Programm die Anzahl der Formatierungsversuche.

Behandelt man fehlerfreie Tracks mit »REFORMAT«, ändert sich an den darauf enthaltenen Daten normalerweise nichts. In ganz seltenen Fällen könnte es aber dennoch einmal zu Problemen kommen. Arbeiten Sie daher im eigenen Interesse stets nur mit Sicherheitskopien.

INFO: <RAMIGA SHIFT i>

Dieser Menüpunkt enthält neben einem kleinen Copyright-Hinweis Angaben über den Diskettenamen, den Diskettentyp und die Anzahl der freien und belegten Blöcke.

INFO => <RAMIGA i>

Nach dem Aktivieren dieses Menüpunkts werden Sie aufgefordert, mit der Maus ein Langwort anzuwählen. Dessen Wert wird als Zeiger auf einen Block der Diskette interpretiert. Das Programm legt diesen Block in einen Zwischenspeicher, öffnet anschließend ein Fenster und gibt darin alles Wissenswerte über den Block aus. Zunächst teilt Ihnen »DisKey« dabei mit, ob der Block frei oder benutzt ist, außerdem um welchen Blocktyp es sich handelt, ob es ein Datenblock, ein File-List-Block, der Rootblock usw. ist. Gehört der Block einem File oder Directory an, erfahren Sie zusätzlich den File-Namen, die Protect-Flags, den Kommentar und das übergeordnete Verzeichnis. Da auf DOS-Disketten sehr viele Zeiger auf bestimmte Blöcke vorhanden sind, ist diese Funktion sehr nützlich, um nachzusehen, für welchen Block ein bestimmter Zeiger zuständig ist. Wer Informationen über den momentan geladenen Block bekommen möchte, klickt statt auf ein Langwort irgendwo in das Fenster der zwölf Befehlsörter.

ENDE: <RAMIGA e> (Kindersicherung)

Dieser Menüpunkt beendet das Programm. Vorher erfolgt eine Sicherheitsabfrage.

WINDOW: <LAMIGA w>

Die Anwahl dieses Befehls öffnet und schließt ein kleines, rotes, verschiebbares Fenster, in dem das Langwort (LW) und das Byte, über dem der Cursor steht, in dezimaler Darstellung angezeigt werden. Außerdem wird der momentane Editiermodus (HEX oder ASCII) angegeben, der auch aus der weißen Umrandung der jeweiligen Fenster ersichtlich ist.

FÜLLEN: <LAMIGA SHIFT f>

Dies ist ein Menüpunkt, den man relativ selten benötigt. Es wird ein ASCII- oder HEX-String eingegeben und der Block automatisch mit diesen Bytes gefüllt. Will man beispielsweise - mit entsprechenden Fähigkeiten - einen Block komplett neu schreiben, kann man ihn hiermit zuvor mit Nullbytes füllen, damit man nicht den Überblick verliert. Aber auch unerwünschte Bootblöcke lassen sich so entfernen. Allerdings muß der alte Wert des ersten Langwortes in Block 0 - typischerweise »DOS« - wieder zurückgeholt werden, denn sonst erkennt das DOS die Diskette nicht mehr. Die Prüfsumme sollte in diesen Fällen jedoch nicht angepaßt werden, sonst versucht der Amiga, von der Diskette zu booten. Dies führt allerdings zu Schwierigkeiten, oder kennen Sie ein Assemblerprogramm, das nur aus Nullbytes besteht?

FILEHEAD: <LAMIGA f>

Diese Funktion lädt den Header-Block eines bestimmten Files oder Verzeichnisses. Das funktioniert aber nur, wenn das File auch für das DOS noch erreichbar ist, d.h., wenn die Zeiger auf dem File noch existieren. Ein Prüfsummen-Fehler macht der Funktion nichts aus.

Von Nutzen ist <LAMIGA F> dann, wenn man Fehler in der Verwaltung eines Files reparieren

möchte, aber nicht weiß, wo sich die entsprechenden Daten auf der Diskette befinden.

INITDISK: <LAMIGA i> (Kindersicherung)

Dieser Befehl schreibt Rootblock, Bootblock und Bit-Allocation-Map der Diskette neu, arbeitet also wie »Format quick«. Sinn und Zweck ist es dabei einerseits, Disketten in Sekundenschnelle zu löschen, andererseits mit »DisKey«-FORMAT formatierte Disketten für das DOS nutzbar zu machen. Die Daten, die auf einer mit »INITDISK« behandelten Diskette waren, sind danach nicht komplett gelöscht, nur das DOS kennt sie nicht mehr und ist danach auch befugt, sie zu überschreiben. Das PD-Programm »DiskSalv« wäre aber durchaus in der Lage, alle Daten wieder zu restaurieren. Lediglich die Daten, die im beweglichen Bitmap-Block oder im Bootblock (Bootintro) standen, sind nicht mehr vorhanden.

PRÜFEN: <LAMIGA p>

Es wird nacheinander ein bestimmter Bereich von Tracks eingelesen. Falls das Betriebssystem einen Fehler aufdeckt, wird dieser von »DisKey« registriert und später in einer Fehlertabelle ausgegeben. Start- und Endtrack dürfen dabei frei gewählt werden. Nützlich ist die Funktion, wenn Sie Fehler in der Magnetschicht lokalisieren möchten, um sie danach mit »REFORMAT« - wenigstens teilweise - zu beheben.

Achtung: In der uns vorliegenden Version von »DisKey« konnte diese Funktion nicht über Tastatur aufgerufen werden. Benutzen Sie daher sicherheitshalber das Pull-Down-Menü oder die Menüpunkte zwischen den beiden ASCII-Fenstern.

RAM: AUF: <LAMIGA a>

In der RAM-Disk wird der File »DisKey 0« geöffnet, den Sie mit den folgenden Menüpunkten zum Speichern von Daten verwenden können. Wenn Sie diesen Menüpunkt im Verlauf Ihrer Arbeit mit »DisKey« mehrmals verwenden, erhöht sich der Zähler stufenweise, d.h. die nächste Datei bekommt den Namen »DisKey 1« etc.

ALLES: <LAMIGA y>

Der gesamte Block wird an die RAM-Datei gehängt. Dadurch ist es

möglich, mit »DisKey« Bootblöcke zu archivieren. Zwar ist das nicht gerade komfortabel, aber immerhin noch besser, als erst ein neues Programm von der Diskette laden zu müssen. Die Vorgehensweise ist denkbar einfach: Zunächst wird mit »RAM: AUF« eine RAM-Datei geöffnet, dann Block 0 geladen und auf »ALLES« geklickt. Anschließend lädt man Block 1, klickt wieder auf »ALLES« und schließt die Datei mit »RAM: ZU«. Der Bootblock steht dann als 1024-Byte-File unter dem Namen »DisKey 0« in der RAM-Disk. Aber auch, wenn man Daten zwischenspeichern möchte, kann man dies über die RAM-Datei tun. Zwar ist das auch mit »MERKEN« und »HOLEN« möglich, allerdings läßt sich damit - im Gegensatz zu »ALLES« - jeweils nur ein Block sichern. Diese Daten werden dann mit dem Menüpunkt »-> ALLES« wieder zurückgeholt.

DATEN: <LAMIGA x>

Dieser Punkt läuft im Prinzip wie »ALLES«. Allerdings funktioniert »DATEN« nur in Datenblöcken, denn es werden nicht die gesamten 512 Byte gesichert, sondern nur die 488 Datenbyte, manchmal - beim letzten Datenblock einer Datei - auch weniger. Will man von Hand eine Datei zusammenflicken, so ist dieser Menüpunkt von Nutzen, da hierbei die Verwaltungsdaten nicht mit abgespeichert werden.

RAM: ZU: <LAMIGA z>

Die mit »RAM: AUF« geöffnete Datei wird geschlossen und darf anschließend nach Belieben bearbeitet werden. »RAM: ZU« wird automatisch beim Öffnen einer neuen Datei oder beim Beenden von »DisKey« aufgerufen.

= > RAM: <LAMIGA r>

Hierbei handelt es sich neben »REFORMAT« um die leistungsstärkste Funktion von »DisKey«, um verlorene Dateien zu restaurieren. Die Diskette selbst wird dabei nicht verändert, dafür aber die zu rettenden Daten in einer RAM-Datei abgespeichert. Eventuell zuvor geöffnete RAM-Dateien werden geschlossen. Die neue Datei erhält wieder den Na-

men »DisKey<n>«. Falls absolut unlesbare Tracks auftauchen, während »DisKey« eine Datei zu lesen versucht, wird die Datei »DisKey<n>« geschlossen und eine weitere Datei mit dem Namen »DisKey<n>. teil1« geöffnet. Für Programme ergibt dies zwar keinen Sinn, aber wenigstens ASCII-Dateien können so einigermaßen gerettet werden.

Die Aufteilung in zwei Dateien wird vorgenommen, damit unaufmerksame Anwender nicht denken, alle Daten seien gerettet, obwohl ein Stück fehlt. Falls sich Harderrors (Fehler in

Restoration verlorener Daten

der Magnetschicht) auf einer Diskette befinden, empfiehlt es sich, diese vorher mit »REFORMAT« zu beseitigen, damit die »=> RAM:«-Funktion besser eingreifen kann. Um »=> RAM:« benutzbar zu machen, muß man sich in dem File-Header oder in einem Datenblock der Datei befinden, wobei man in jedem Fall den File-Header bevorzugen sollte. Befindet man sich im File-Header, erhält »DisKey« Informationen über die entsprechenden Datenblöcke sowohl durch die Zeiger, die im Header stehen, als auch durch die miteinander verketteten Datenblöcke. So sind immer zwei Informationen über den nächsten zu rettenden Block vorhanden. Falls diese unterschiedlich ausfallen, versucht »DisKey« den richtigen Block abzuspeichern.

In einem Datenblock wird nur ein Zeiger auf den nächsten gefährdeten Block benutzt. Tritt ein schwerer Fehler auf, gibt es keine Möglichkeit, den Rest der Datei zu retten - der Vorgang muß abgebrochen werden. Die in der RAM-Disk entstandene Datei sollte man später unter dem alten Namen auf eine fehlerfreie Diskette zurückkopieren. Erinnern Sie sich noch, wie man in den File-Header-Block einer Diskette gelangt? Dies geschieht mit »FILEHEAD«. Es funktioniert allerdings nicht bei gelösch-

ten Dateien, da dann die Zeiger auf den Header-Block unterbrochen wurden. Aber mit Hilfe der Suchroutine kann man nach dem Filenamen suchen. Er beginnt immer beim 433. Byte eines Blocks. Falls der gesuchte String an anderer Stelle auftaucht, handelt es sich nicht um den gewünschten File-Header. Für zukünftige »DisKey«-Versionen ist auch eine Option vorgesehen, mit der man nach bestimmten Blocktypen suchen kann. Haben Sie den Block gefunden, genügt ein Klick auf »=> RAM:«, und »DisKey« beginnt mit dem Einlesen der Daten. Dabei zeigt das Programm im Statusfenster alle momentan gelesenen Blöcke an. Ist der File-Header nicht mehr auffindbar, so sucht man nach einem Datenblock der Datei, möglichst nach dem ersten. Dies ist allerdings nur dann möglich, wenn man eine Bytefolge der Datei kennt. Ein anschließender Klick auf »=> RAM:« startet den Restaurierungs-Vorgang ab dem Datenblock, in dem man sich momentan befindet. Vorherige Datenblöcke werden dabei nicht gerettet.

-> ALLES: <LAMIGA SHIFT y>

Dies ist quasi die Gegenfunktion zu »ALLES«. Es wird eine beliebige Datei eingelesen. Anschließend überschreibt ein Block dieser Datei den momentan bearbeiteten Block. So können mit »ALLES« gespeicherte Daten - beispielsweise Bootblöcke - wieder geladen werden. Außer dem Dateinamen muß der Benutzer noch den Blockoffset eingeben, ab dem die Datei geladen werden soll. Blockoffset 1 lädt ab dem ersten Byte, Offset 2 ab dem 513. Byte etc.

-> DATEN: <LAMIGA SHIFT x>

»-> DATEN« ist die Umkehrung zu »DATEN« und dient dazu, in der RAM-Disk abgelegte Daten wieder einzuladen. Die Arbeitsweise ist dieselbe wie bei »-> ALLES«. Der Unterschied besteht nur darin, daß »-> DATEN« nur innerhalb von Datenblöcken anwählbar und die Datei in Blöcke von 488 Byte unterteilt ist. Diese 488 Byte werden entsprechend dem Aufbau der Datenblöcke ab dem 7. Langwort eingeladen.

SPRINGEN: <CTRL LAMIGA s>

Nach dem Anwählen dieser Funktion klicken Sie ein beliebiges Lang-

wort an. Es wird als ein Zeiger auf einen Block interpretiert, der anschließend geladen wird. Da das Filesystem seine Blöcke durch Zeiger verkettet, gelangt man durch diese Funktion sehr schnell zu einem bestimmten Block, ohne die Blocknummer direkt eingeben zu müssen. Anfänger können sich den Block zuvor auch mit »INFO = >« ansehen.

ROOT: <CTRL LAMIGA r>

Diese Funktion lädt den Rootblock eines Speichermediums. Bei Disketten läßt sich dessen Nummer (880) zwar noch gut merken und auch per Hand eingeben, bei Festplatten, re-setfesten RAM-Disks etc. variiert dieser Wert jedoch.

BOOT: <CTRL LAMIGA b>

Lädt den Bootblock, also Block 0, einer Diskette. Einziger Zweck ist dabei, dem Anwender etwas Tipparbeit abzunehmen.

BAM: <CTRL LAMIGA a>

Lädt die Bit-Allocation-Map (BAM) der Diskette. In der BAM sollte nur Änderungen vornehmen, wer sich damit auskennt.

FOLGE: <CTRL LAMIGA f>

Wenn man sich in einem Datenblock oder in einem File-Header befindet, lädt diese Funktion den nächsten Datenblock der Datei. So ist es möglich, sich innerhalb einer Datei Stück für Stück bis ans Ende zu bewegen.

VORHER: <CTRL LAMIGA v>

Befindet man sich im zweitem oder einem höheren Datenblock, lädt »DisKey« den vorhergehenden Datenblock dieser Datei. Da »DisKey« diesen Block nur über den File-Header- bzw. den File-List-Block abfragen kann, müssen die dortigen Zeiger intakt sein. »VORHER« ist die Umkehrfunktion zu »FOLGE«.

LIST: <CTRL LAMIGA l>

Lädt den (ersten) File-List-Block einer Datei, falls man sich im File-Header einer größeren Datei befindet. Verfügt eine Datei über mehrere Listblöcke, so lassen sich diese nacheinander laden, indem man - ausgehend vom ersten List-Block - mehrmals auf »LIST« klickt.

HEADER: <CTRL LAMIGA h>

Lädt den File-Header-Block einer Datei, sofern man sich in einem dazugehörigen Daten- oder List-Block befindet. Sehr wichtig ist diese Funktion, wenn man mit »=> RAM:« eine Datei retten möchte, bisher aber nur einen Daten- oder Listblock gefunden hat.

PARENT: <CTRL LAMIGA p>

Hiermit laden Sie - ausgehend von einem File-Header-Block oder einem Directory-Header - das übergeordnete Verzeichnis, also entweder einen Directory-Header oder den Rootblock.

=> USED: <LAMIGA u>

Die momentan geladene Block wird in der Bit-Allocation-Map als benutzt eingetragen. Solange der Diskvalidator nicht eingreift, ist es dem DOS strengstens untersagt, diesen Block zu überschreiben.

__> FREI: <LAMIGA u>

Der aktuelle Block wird freigegeben, so daß ihn das DOS mit beliebigen Daten überschreiben darf. »-> USED« sollten Sie nur mit Vorsicht benutzen. Zu leicht könnte es geschehen, daß man dem DOS wichtige Blöcke zum Überschreiben reserviert und dies bereut, wenn dadurch wichtige Daten zerstört wurden.

Um Befehle wie »VORHER«, »FOLGE«, »HEADER« usw. besser verstehen zu können, folgt nun eine Beschreibung zum Aufbau der Blöcke unter dem Amiga-Filesystem. Wie die einzelnen Blöcke miteinander verknüpft sind, sehen Sie in Bild 1 (Seite 32).

Der Bootblock

Ein recht unkomplizierter Block ist der Bootblock (Tabelle 1). Als erster Eintrag ist hier die Diskettenkennung vermerkt, normalerweise »DOS\0«. Kickstart-Disketten tragen die Kennung »KICK«. Deswegen erscheint auf der Workbench »DF0:KICK«, wenn sich eine Kickstart-Diskette im Laufwerk befindet. Hatten Sie schon einmal eine Diskette, die auf der Workbench als »DF0:TüBu« angezeigt wurde? Hierbei handelt es sich

Der Bootblock

Langwort	Bedeutung
1	Kennung, \$444F5300
2	Prüfsumme über die beiden Bootblöcke
3	Zeiger auf den Rootblock, \$00000370
4 bis ?	Assemblerprogramm, Viren, Bootladeblöcke

Tabelle 1. Der Aufbau des unkomplizierten Bootblocks

Der Rootblock

Langwort	Bedeutung
1	Erste Kennung des Rootblocks, \$00000002
2 und 3	Reserviert, gewöhnlich \$00000000
4	Anzahl der Einträge in der Hash-Tabelle, also 72
5	Reserviert, gewöhnlich \$00000000
6	Prüfsumme über den Rootblock
7 bis 78	Hash-Tabelle
79	Das Bitmap-Flag. Wenn es ungleich 0 ist, ist die Bitmap gültig.
80 bis 105	Zeiger auf die Bitmap-Blöcke. Bei Disketten ist nur das 80. Langwort belegt. (Wird von »BAM« benutzt)
106 bis 108	Genaueres Erstellungsdatum
109 bis 124	Diskettenname. Das erste Byte kennzeichnet die Länge des Namens
125 bis 127	Reserviert, gewöhnlich 0
128	Zweite Kennung des Rootblocks, \$00000001

Tabelle 2. Im Rootblock finden Sie die Zeiger für File-Directory-Header

Der Directory-Block

Langwort	Bedeutung
1	Erste Kennung des Directoryblocks, \$00000002
2	Zeiger auf den eigenen Block
3 bis 5	Reserviert, gewöhnlich \$00000000
6	Prüfsumme über den Directoryblock
7 bis 78	Hash-Tabelle
79 und 80	Reserviert, gewöhnlich \$00000000
81	Protection-Flags
82	Reserviert, gewöhnlich \$00000000
83 bis 105	Kommentar. Das erste Byte kennzeichnet die Länge des Kommentars
106 bis 108	Genaueres Erstellungsdatum
109 bis 124	Verzeichnisname. Das erste Byte kennzeichnet die Länge des Verzeichnisses.
125	Nächster Eintrag mit gleichem Hash-Wert
126	Zeiger auf übergeordnetes Directory (wird von »PARENT« benutzt)
127	Reserviert, gewöhnlich 0
128	Zweite Kennung des Directoryblocks, \$00000002

Tabelle 3. Der Directory-Block ist dem Rootblock sehr ähnlich

um eine Diskette, die mit Turbo-backup kopiert wurde, wobei der Kopiervorgang vorzeitig abgebrochen wurde. Wenn im ersten Langwort eine für das Filesystem unbekannt Kennung enthalten ist, wird die gesamte Diskette nicht angenommen. Langwort 2 enthält die Prüfsumme über beide Bootblöcke und Langwort 3 einen Zeiger auf den Rootblock, der üblicherweise den Wert 880 aufweist. Anschließend folgt das Assemblerprogramm zum Booten. Bootunfähige Disketten haben eine ungültige Prüfsumme. Weder der Zeiger auf den Rootblock noch das

Assemblerprogramm sind dann nötig. Die meisten anderen Blöcke enthalten zwei Kennungen, an denen das Filesystem erkennt, um welchen Blocktyp es sich handelt. Die erste Kennung befindet sich im ersten Langwort, die zweite im letzten.

Der Rootblock

Der wichtigste Block einer Diskette ist der Rootblock (Tabelle 2). Hier sind die Zeiger auf File-Header und Directory-Header vermerkt. Dafür stehen 72 Einträge zur Verfügung.

Der File-Header-Block

Langwort	Bedeutung
1	Erste Kennung des File-Header-Blocks, \$00000002
2	Zeiger auf den eigenen Block
3	Anzahl der im Header vermerkten Blöcke, maximal 72
4	Reserviert, gewöhnlich \$00000000
5	Zeiger auf den ersten zugehörige Datenblock
6	Prüfsumme über den File-Header-Block
7 bis 78	Zeiger auf die Datenblöcke
79 und 80	Reserviert, gewöhnlich \$00000000
81	Protection-Flags.
82	Größe des Files in Bytes
83 bis 105	Kommentar. Das erste Byte kennzeichnet die Länge des Kommentars.
106 bis 108	Genaues Erstellungsdatum
109 bis 124	Dateiname. Das erste Byte kennzeichnet die Länge des Namens.
125	Nächster Eintrag mit gleichem Hash-Wert
126	Zeiger auf übergeordnetes Directory (wird von »PARENT« benutzt)
127	evtl. Zeiger auf den ersten List-Block (wird von »LIST« benutzt)
128	Zweite Kennung des File-Header-Blocks, \$FFFFFFFD

Tabelle 4. Im File-Header-Block sind die Zeiger umgekehrt angeordnet

Der File-List-Block

Langwort	Bedeutung
1	Erste Kennung des File-List-Blocks, \$00000010
2	Zeiger auf den eigenen Block
3	Anzahl der im List-Block vermerkten Blöcke, maximal 72
4	Reserviert, gewöhnlich \$00000000
5	Laut Literatur der Zeiger auf den ersten Datenblock, in der Praxis aber unbenutzt, \$00000000
6	Prüfsumme über den Filelistblock
7 bis 78	Zeiger auf die Datenblöcke, wie beim Fileheader angeordnet
79 bis 125	Reserviert, gewöhnlich \$00000000
126	Zeiger auf Fileheader (wird von »HEADER« benutzt)
127	evtl. Zeiger auf den nächsten List-Block (wird von »LIST« benutzt)
128	Zweite Kennung des File-List-Blocks, \$FFFFFFFD

Tabelle 5. Der Listblock enthält bei größeren Dateien die restlichen Datenblöcke

Anhand des File-Namens kann das DOS errechnen, in welchem Eintrag der Zeiger zu einem bestimmten File vermerkt ist. Das folgende kurze Basic-Listing soll beim Errechnen dieses Wertes helfen.

```
INPUT "Filename => ", nam$
laenge=LEN (nam$)
hash=laenge
FOR i=1 TO laenge
  IF MID$ (nam$, i, 1) >= "a"
  AND MID$ (nam$, i, 1) <="z"
  THEN
    MID$ (nam$, i, 1)=CHR$ (ASC
(MID$ (nam$, i, 1))-32)
  END IF
NEXT i
FOR i=1 TO laenge
  hash=hash*13
```

```
hash=hash+ASC (MID$
(nam$, i, 1))
hash=hash AND 2047 \
NEXT i
hash=hash MOD 72
PRINT "Der Filename "; nam$; "
trägt den Hashwert"; hash; "."
PRINT "Er wird also im
Langwort"; hash+7; "
eingetragen!"
END
```

Da aber öfters mehr als 72 Einträge im Root Platz finden müssen und manchmal mehrere Filenamen im selben Eintrag vermerkt werden, verkettet das Filesystem in solchen Fällen die entsprechenden Blöcke. Mehr dazu folgt in der Beschreibung der Directory-Blöcke. Im Rootblock befinden sich außerdem die Zeiger auf die Bit-Allocation-Map-Blöcke, die die freien und belegten Blöcke verwalten. Disketten verfügen nur über einen solchen Block, Festplat-

ten allerdings über mehrere, da hier wesentlich mehr Blöcke zu verwalten sind. Während eines Schreibzugriffs sind die BAM-Blöcke ungültig, da sich gerade etwas auf der Diskette ändert. Zu diesem Zeitpunkt wird ein Flag im Rootblock gesetzt, das diesen Status kennzeichnet. Nimmt man anschließend die Diskette aus dem Laufwerk, oder führt man einen RESET durch, so wird beim nächsten Einlegen der Diskette zunächst die BAM überprüft und gegebenenfalls richtiggestellt, was eine gewisse Zeit in Anspruch nimmt.

Der Directory-Block

Der Directory-Block (Tabelle 3) weist gewisse Ähnlichkeiten mit dem Rootblock auf. Im Langwort 125 befindet sich noch ein zusätzlicher Zeiger, der auf den nächsten Block mit gleichem Hash-Wert zeigt. Das kann ein File-Header-Block oder ein weiterer Directory-Block sein. Diese Liste kann bei großen Verzeichnissen relativ lang werden. Entsprechend erhöht sich auch die Zugriffszeit auf ein File, das am Ende einer solchen Liste hängt, da erst alle vorangegangenen Einträge durchlaufen werden müssen.

Der File-Header-Block

Der File-Header-Block (Tabelle 4) ähnelt wiederum dem Directory-Block. Die Hash-Tabelle wird durch die Zeiger auf die zum File gehörigen Datenblöcke ersetzt. Seltsamerweise sind diese Zeiger umgekehrt angeordnet, das heißt, der letzte Zeiger zeigt auf den ersten Datenblock usw.

Der File-List-Block

Auch der File-List-Block (Tabelle 5) kann seine Verwandtschaft mit dem Headerblock nicht leugnen. Im Listblock sind bei größeren Dateien die restlichen Datenblöcke vermerkt, da der Header selbst nur 72 Einträge aufnehmen kann. Falls ein einziger Listblock nicht genügt, werden mehrere miteinander verkettet.

Der Datenblock

Bisher haben wir uns nur mit den Verwaltungsdaten beschäftigt. Natürlich gibt es auch noch einen Blocktyp, der die eigentlichen Daten

aufnimmt. Dies ist der Datenblock, der relativ wenig Gemeinsamkeiten mit den anderen Blöcken aufweist (Tabelle 6). Er nimmt unter dem bisherigen Amiga-Filesystem maximal 488 Byte Daten auf und besitzt im Unterschied zu den anderen Blöcken nur die erste Kennung.

Die Bit-Allocation-Map (BAM)

Der letzte noch nicht behandelte Block des Amiga-Filesystems ist die Bit-Allocation-Map. Diese besitzt keine Kennung. Im ersten Langwort befindet sich die Prüfsumme über die BAM, die restlichen Langwörter werden intern in einzelne Bits aufgeteilt. Jeweils ein Bit repräsentiert einen Block der Diskette. Ist ein Bit gesetzt, so ist der Block frei, ist das Bit gelöscht, so ist er belegt. Bei Disketten werden 55 Langwörter benötigt, um alle Blöcke der Diskette zu verwalten. Bootblöcke sind übrigens immer belegt und verfügen über keinen Eintrag in der Bit-Allocation-Map.

Die »DisKey«-Konfigurationsdatei

Nun haben wir die nötige Theorie durchgearbeitet, um sinnvoll mit »DisKey« arbeiten zu können. Wie bereits erwähnt, besitzt »DisKey« eine eigene Konfigurationsdatei, die wichtige Einstellungen vornimmt (einige dieser Parameter lassen sich auch nachträglich über das Pull-Down-Menü »DisKey.Prefs« ändern). Die Konfigurationsdatei wird grundsätzlich im aktuellen Verzeichnis unter dem Namen »DisKey.Prefs« gesucht. Wird die Datei dort nicht gefunden, sieht »DisKey« im S:-Directory nach. Falls »DisKey.Prefs« auch dort nicht vorhanden ist, übernimmt das Programm die Voreinstellungen.

Die Konfigurationsdatei ist ein denkbar einfach aufgebautes ASCII-File, das sich mit jedem Texteditor bearbeiten läßt. Die Schlüsselwörter, die unten erläutert werden, müssen immer am Anfang einer Zeile stehen. Dann folgt ein von beliebig vielen Leerzeichen umgebenes »=«-Zeichen und schließlich der gewünschte Wert. Kommentare darf man nach Wunsch hinzufügen, ohne diese irgendwie kennzeichnen zu müssen.

Oftmals besteht der String nach dem »=«-Zeichen nur aus einem Flag, also »TRUE« oder »FALSE«. In

Der Datenblock

Langwort	Bedeutung
1	Kennung des Datenblocks, \$00000008
2	Zeiger auf den File-Header
3	Laufende Nummer des Datenblocks
4	Anzahl der benutzten Bytes des Datenblocks, maximal \$1E8, also 488
5	Zeiger auf den nächsten Datenblock der Datei
6	Prüfsumme über den Datenblock
7 bis 127	Die eigentlichen Daten

Tabelle 6. In diesem Block sind die eigentlichen Daten enthalten

diesem Fall genügt es, den ersten Buchstaben anzugeben. Falls mehrere Buchstaben angegeben werden, interpretiert »DisKey« diese als Kommentar. Es spielt keine Rolle, ob die Schlüsselwörter in Groß- oder Kleinschrift eingegeben werden. Lediglich bei Umlauten sollte man auf die korrekte Schreibweise achten.

Falls »DisKey.Prefs« auf Ihrer Diskette nicht vorhanden ist, können Sie sich diese Datei selbst erstellen. Beachten Sie dabei die nachfolgenden Erläuterungen der verschiedenen Schlüsselwörter und Werte.

BAMCHANGE = T oder F

Wenn der Benutzer mit »=> FREI« und »=> USED« die BAM ändert, muß das dem DOS explizit mitgeteilt werden. Diese Mitteilung benötigt allerdings eine gewisse Zeit und wirkt deswegen manchmal recht störend. Nach dem Verlassen von »DisKey« wird diese Meldung automatisch gesendet. Möchte man zusätzlich während der Arbeit mit »DisKey« im Multitasking-Betrieb auf Daten der Diskette zugreifen, sollte man hier die Voreinstellung »TRUE« übernehmen.

BASE = 0 oder 1

Wie Sie bemerkt haben, zählt »DisKey« die Langwörter, über denen der Cursor steht, von 1 bis 128 durch. Auch in den Tabellen dieses Workshops über die Blockbelegung wurde diese Zählweise angewandt. Wenn man die Langwörter statt dessen lieber von 0 bis 127 durchnummeriert haben möchte, gibt man »BASE=0« ein. Der Wert von »BASE«

gilt auch für »-> ALLES« und »-> DATEN«. Voreingestellt ist »BASE=1«.

COLOR <n> = RGB

Falls Ihnen die Farben des »DisKey«-Screens nicht zusagen, ändern Sie hiermit die Farbgregister 0 bis 3. Der Wert <n> entspricht dabei der Nummer des Farbgregisters und RGB der Farbe, wie Sie sie von den drei Schieberegler Rot-Grün-Blau bei Malprogrammen kennen. RGB wird hexadezimal eingegeben, das heißt es sind Ziffern sowie die Buchstaben A bis F erlaubt. Um das Register 2 auf die Farbe Schwarz einzustellen, geben Sie »Color2=000« ein. »Color1=FFF« ergibt die Vordergrundfarbe Weiß, »Color3=0F0« dagegen ein Grün als Farbe 3.

CURSORDELAY= <n> und CURSORSPED= <n>

Diese und die nächste Option sind vor allem für die Arbeit mit dem Workbench-Editor »Ed« hilfreich. Um mit »Ed« arbeiten zu können, muß eine relativ geringe Cursorgeschwindigkeit eingestellt sein. Damit »DisKey« nicht ebenfalls mit dieser Geschwindigkeit arbeiten muß, kann man sie hiermit ändern. Wegen möglicher Kompatibilitätsprobleme mit anderen Programmen, und weil diese beiden Befehle ohnedies nur selten benötigt werden, werden zukünftige Versionen von »DisKey« diese Option nicht mehr enthalten. »Cursordelay« steht für die Zeitspanne, die der Cursor wartet, bis die Tastenwiederholung einsetzt. »Cursor-speed« bezeichnet die Tastenwiederholungsgeschwindigkeit. Der Wert <n> wird in $\frac{1}{50}$ s eingegeben, »Cursordelay=15« setzt demnach eine Tastenwiederholrate von $\frac{15}{50}$ s, »Cursorspeed=2« eine Tastenwie-

derholgeschwindigkeit von $\frac{2}{50}$ s. Die beiden Werte sind bereits voreingestellt. Bei Cursordelay dürfen bis zu zweistellige Zahlen eingegeben werden, bei Cursorspeed nur einstellige. Nach dem Starten von »DisKey« ist ein Ändern dieser Werte allerdings nicht mehr möglich.

DRIVE= <n> oder <Device-name:>

Mit »Drive« wird das Laufwerk eingestellt, von dem nach dem Starten von »DisKey« gelesen werden soll. Man gibt entweder den Devicenamen an, also »Drive=df0:« oder die Nummer, z. B. »Drive=0«. In der Share-Version sind beide Einstellungen identisch. Voreingestellt ist »Drive=0«.

EDMODE = A oder H

Für die Arbeit mit »DisKey« stehen Ihnen zwei verschiedene Editiermodi zur Verfügung. Normalerweise ist das der ASCII-Modus, da dieser mit »EdMode=A« voreingestellt ist. Ziehen Sie dagegen den Hex-Modus vor, brauchen Sie nur »EdMode=H« einzugeben.

INHIBIT = T oder F

Manche Disketten sind so schwer beschädigt, daß das DOS bereits beim Einlegen der Diskette mit »Disk corrupt« abstürzt. Setzt man das Inhibit-Flag mit »Inhibit=T«, so wird dies verhindert. Das DOS ist dann auch nicht mehr in der Lage, von der Diskette zu lesen oder sie gar zu beschreiben. Wenn das Inhibit-Flag gesetzt ist, braucht das DOS nach Änderungen in der Bit-Allocation-Map nicht informiert zu werden. Es ist aber zu beachten, daß die aktuelle Version 2.0 von »DisKey« bei »Inhibit=TRUE« keine Diskettenwechsel mehr erkennt. Deswegen wurde »Inhibit=FALSE« voreingestellt.

Kindersicherung = T oder F

Manchmal kann die eingebaute »Kindersicherung« sehr lästig und zeitraubend sein. In diesem Fall wird sie einfach mit »Kindersicherung=F« ausgeschaltet. Man sollte dann aber besondere Vorsicht walten lassen. Der Normalfall ist eine mit »Kindersicherung=T« aktivierte Kindersicherung.

Lace = T oder F

Mit »Lace=T« läßt sich »DisKey« auf einem Interlace-Screen starten. Diese Option ist nur für NTSC-Amigas von Bedeutung und deswegen normalerweise ausgeschaltet. »Lace« kann nachträglich nicht geändert werden.

PrintFile = <Pfadname> #

Mit dieser Funktion definieren Sie den Pfad für den Druckbefehl von »DisKey«. Aber auch, wenn man keinen Drucker angeschlossen hat, ist es sogar möglich, die Funktion »DRUCKEN« zu benutzen. Die Daten werden dann nicht an den Drucker geschickt, sondern in eine Datei, die mit »PrintFile« definiert wird.

Raffinierte Druckoptionen

Voreingestellt ist »PrintFile=PRT: #«. Das »#«-Zeichen ist nötig, damit »DisKey« erkennt, wann der Pfadname zu Ende ist. Eventuell folgende Bytes werden als Kommentar interpretiert. Ist kein Drucker angeschlossen, kann hier eine Datei angegeben werden:

PrintFile=RAM:DisKeyDruckDatei#

Wer möchte, kann auch »PrintFile=NIL: #« einstellen. Falls man dann versehentlich auf »DRUCKEN« klickt, muß man nicht lange warten, bis das DOS einen Error-Requester zum Vorschein bringt, den man mit »Cancel« beantworten muß.

PrintInfo = T oder F

Beim Drucken wird normalerweise eine Statuszeile ausgegeben, die den Diskettennamen und den entsprechenden Block beinhaltet. Falls das unerwünscht ist, gibt man hier »PrintInfo = F« ein.

RAWREAD = T oder F

Wie bereits erwähnt, wird zur Identifizierung des Lamer-Virus von »DisKey« der Bootblock stets doppelt gelesen. Das Lesen mit »RAWREAD«

jedoch kostet Zeit. Wenn Sie sicher sind, daß sich kein Lamer-Virus im System befindet, und daß Sie keine defekten Tracks zu lesen haben, dürfen Sie an dieser Stelle »RAW-READ=F« eintragen. »REFORMAT« funktioniert trotzdem noch.

Retten = <Pfadname # >

Normalerweise wird mit »RAM:AUF« und »=> RAM:« ein File auf der RAM-Disk geöffnet. Mit »Retten=DH0:Empty/ # « lenken Sie das File auf die Festplatte ins Verzeichnis »Empty« um. Vergessen Sie dabei jedoch den Schrägstrich nicht. Voreingestellt ist »Retten=RAM: # «.

Screen = W oder C

Diese Angabe bestimmt, ob »DisKey« auf dem Workbenchscreen oder auf einem Customscreen startet. Bei genügend Speicher ist in jedem Fall mit »Screen = C« ein eigener Screen vorzuziehen, da sich das »DisKey«-Fenster nicht verkleinern läßt und (noch) keine Iconify-Option eingebaut ist.

SetPrefs = T oder F

Wenn man möchte, daß die eingestellte Cursorgeschwindigkeit benutzt wird, setzt man »SetPrefs=T«, ansonsten »SetPrefs=F«. Wer »MachII« benützt, sollte auf jeden Fall in der Konfigurationsdatei »SetPrefs=F« einstellen, da sonst Kompatibilitätsprobleme auftreten.

ShowNull = T oder F

Wenn es stört, daß »DisKey« beim Einlegen einer Diskette den Bootblock (Block 0) anstatt des Rootblocks anzeigt, schreibt hier »ShowNull=F«. Voreingestellt ist wegen der Virengefahr allerdings »ShowNull=T«.

So, das war's. Wenn Sie diesen Workshop aufmerksam gelesen und durchgearbeitet haben, sind Sie jetzt ein richtiger »DisKey«-Profi und verstehen etwas vom Aufbau einer Amiga-Diskette. Wir wünschen Ihnen viel Spaß und Freude bei Ihren Streifzügen durch das Innenleben Ihrer Disketten. ag

Bezugsquellen:

»Share 5«

A.P.S. elektronik, Sonnenborstel 31, 3071 Steimbke, Tel.: 05026/1700

Herrmanns & Kommelter, Vom Bruck Platz 45, 4150 Krefeld, Tel.: 02151/399833

Weiterführende Literatur:

Das große Floppy-Buch, Verlag DATA BECKER, Merowinger Str. 30, 4000 Düsseldorf 1, DM 59,-

Alles paletti mit Amiga-Basic

Lieber mit Schieber

Man kann ein Programm professioneller wirken lassen, wenn Werte nicht über die Tastatur eingegeben, sondern mit der Maus über einen Schieberegler eingestellt werden. Schieberegler kann man beim Amiga programmieren, indem man die Routinen aus der intuition.library einsetzt; es geht aber auch anders. Das folgende Basic-Programm zeigt, wie man einen solchen Schieberegler mit wenig Aufwand programmiert:

```
LINE (50,20)-(100,150),2,bf
LINE (50,20)-(100,150),3,b
LOCATE 12,15 : PRINT "Wert : "
Maus:
v=MOUSE(0)
x=MOUSE(1)
y=MOUSE(2)
WHILE MOUSE(0)=0 : WEND
IF x>50 AND x<100 AND y>20 AND y<150 THEN GOTO Regeln
GOTO Maus
Regeln:
LINE(51,21)-(99,149),2,bf
LINE (52,148)-(98,y),3,bf
LINE (98,y)-(98,y),3,bf
LOCATE 12,25 : PRINT y
GOTO Maus
```

Probieren Sie das Programm aus. Wenn Sie innerhalb des schwarzen Kastens die linke Maustaste gedrückt halten, folgt der orange Balken Ihren Bewegungen; rechts wird der aktuelle Wert der y-Koordinate des Schiebers angezeigt. So lassen sich beliebige Werte einstellen, die Sie im weiteren Programm einsetzen können.

Marcus Högner/lub

Basic schlägt Alarm

Auch in Basic kann man Alarm-Meldungen (Alerts) ausgeben. Das folgende Listing zeigt, wie man vorgehen sollte:

```
DECLARE FUNCTION DisplayAlert& LIBRARY
LIBRARY "intuition.library"
ON ERROR GOTO fehler

REM absichtlich Fehlermeldung erzeugen
OPEN "name" FOR INPUT AS#1

ende:
CLOSE 1
LIBRARY CLOSE
END

REM Fehlerabfrage und Aufruf Alert
fehler:
IF ERR <> 0 THEN CALL Alert
RESUME ende
```

```
SUB Alert STATIC
type& = 0 : hoehe% = 60
fehler = ERR : fehler$ = STR$(fehler)
text$ = CHR$(0)+CHR$(150)+CHR$(15)
text$ = text$+"Programmabbruch wegen Fehlernummer: "
text$ = text$+fehler$+CHR$(0)+CHR$(1)
text$ = text$+CHR$(0)+CHR$(150)+CHR$(28)+"Dr "+CHR$(252)
text$ = text$+"cken Sie BITTE die linke Maustaste"
text$ = text$+CHR$(0)+CHR$(1)+CHR$(0)+CHR$(160)+CHR$(45)
text$ = text$+"GESCHRIEBEN VON WOLFGANG STELLWAGEN"+CHR$(0)
warten:
```

```
taste& = DisplayAlert&(type&,SADD(text$),hoehe%)
IF taste& <> 1 THEN warten
END SUB
```

Das Programm nutzt eine Subroutine zur Fehlerabfrage mit einem Recovery-Alert in Basic. Aufgerufen wird der Alert mit:

```
DisplayAlert (Typ,Text,Höhe)
```

Die Variable Typ gibt an, welcher Art der Alert ist (Recovery-Alert = 0). Text ist ein Zeiger auf den darzustellenden Text und Höhe gibt die Höhe des Alerts in Punkten auf dem Bildschirm an. Mit veränderter Fehlerabfrage und anderen Texten kann man das Programm seinen Wünschen anpassen.

Wolfgang Stellwagen/lub

Riesiger Screen mit Automatik

Der Amiga ist in der Lage, mehr als 640 x 512 Punkte auf dem Monitor darzustellen. Wie? Im sogenannten Overscan (bis 704 x 564). Um in Basic ein größeres Bild verwenden zu können, muß der Bildschirm nach links oben verschoben werden, damit alle Punkte sichtbar sind. Diese Einstellung erfolgt mit den Preferences. Sie kann aber auch per Programm vorgenommen werden:

```
LIBRARY "intuition.library"
LIBRARY "graphics.library"
DECLARE FUNCTION ViewAddress& LIBRARY
ViewAdd&=ViewAddress&
POKEW ViewAdd&+14,112
POKEW ViewAdd&+12,30
CALL RemakeDisplay
SCREEN 1,704,560,tiefe,4
WINDOW 2,,16,1
LINE (0,0)-(695,555),3,B
FOR i=1 TO 10000
NEXT i
POKEW ViewAdd&+14,129
POKEW ViewAdd&+12,44
RemakeDisplay
END
```

Mit »MoveScreen« läßt sich ein Screen lediglich in der Vertikalen verschieben. Eine Möglichkeit, ein Bild auch in der Horizontalen zu bewegen und zu zentrieren, besteht im Ändern der View-Daten. Die letzten vier Zeilen des Beispiels stellen vor Verlassen des Programms die alte Bildposition wieder her. Es kann unter Umständen nötig sein, im Overscan das Bild mit den Reglern am Monitor zu verkleinern, oder je nach Gerät andere Startkoordinaten in der View-Struktur einzusetzen. Hier sollten Sie ein wenig experimentieren.

Bruno Goldmann/lub

Schnelles Blättern mit Basic

Die Tasten <Cursor hoch> und <Cursor unten> in Verbindung mit <Shift> dienen zum seitenweisen Durchblättern in einem Basic-Programm. Wenn Sie ein langes Basic-Programm mit etwa 60 Seiten Umfang durchblättern, dauert dies recht lange. Sie erreichen eine höhere Geschwindigkeit, wenn Sie das Listing vor dem Umblättern mit <Shift> und zweimal <Cursor rechts> aus dem sichtbaren Bereich verbannen. <Shift> und zweimal <Cursor links> holen den Text zurück.

Jens Spitzcok/lub

Reklame für Basic

Eine Laufschrift macht in jedem Vorspann einen professionellen Eindruck. Dabei ist die Programmierung in Basic so einfach. Wir können ein Unterprogramm verwenden:

```
REM VORBEREITUNGEN:
REM DIM SHARED LS%(6+(y2%-y1%+1)*2*INT((ABS(DeltaX)+16)/16)*Tiefe)
REM x1%,y1% , x2%,y2% : Zu scrollender Bereich
REM Anzahl : Wie oft scrollen
REM DeltaX : Siehe Basic-Handbuch, Seite 9 - 223
```

```
SUB LSchrift(x1%,y1%,x2%,y2%,Anzahl,DeltaX) STATIC
DX=(ABS(DeltaX)-1)*SGN(DeltaX)
FOR i=1 TO Anzahl
IF DeltaX<0 THEN GET(x1%,y1%)-(x1%-DX,y2%),LS% : ELSE
GET(x2%,y1%)-(x2%-DX,y2%),LS%
SCROLL (x1%,y1%)-(x2%,y2%),DeltaX,0
IF DeltaX<0 THEN PUT(x2%+DX,y1%),LS%,PSET : ELSE
PUT(x1%,y1%),LS%,PSET
NEXT
END SUB
```

Der Aufruf der Subroutine lautet:

```
CALL LSchrift(x1%,y1%,x2%,y2%,Anzahl,DeltaX)
```

Damit das Unterprogramm arbeiten kann, muß vorher noch das Array LS% folgendermaßen definiert werden:

```
DIM SHARED LS%(6+(y2%-y1%+1)*2*INT((ABS(DeltaX)+16)/16)*Tiefe)
```

Tiefe gibt die Anzahl der Bitplanes an. Ein Beispiel:

```
DIM SHARED LS%(6+(100-70+1)*2*INT((ABS(2)+16)/16)*2)
FOR i=1 TO 20: FOR ii= 1 TO 20 : PRINT ii;
NEXT ii: PRINT : NEXT i:
CALL LSchrift (10,70,309,100,300!,2!) END
```

Sie können das Unterprogramm in Ihre Basic-Programme mit dem MERGE-Befehl einbinden. Ein Bildschirmbereich wird gescrollt, wobei Bildteile, die über den Rand geschoben werden, auf der anderen Seite wieder erscheinen. *Rolf Beck/ub*

Der Copper frißt Basic

Die Programmierung des Coppers ist eine Wissenschaft für sich. Dieser Spezialchip im Amiga kann auf vielfältige Art eingesetzt werden, um den Inhalt des Bildschirms zu beeinflussen. Das folgende Programm zeigt, wie man den Copper auch von Basic ansteuern kann:

```
1 LIBRARY "exec.library"
2 DECLARE FUNCTION AllocMem& LIBRARY
3 mem& = AllocMem&(60,2)
4 IF mem& = 0 THEN PRINT "Kein Speicher" : END
5 FOR I = 0 TO 58 STEP 2 : READ a
6 POKEW mem&+i,a : NEXT
7 plane1& = PEEKL(WINDOW(7)+46)+192
8 plane2& = PEEKL(WINDOW(7)+46)+196
9 POKEW mem&+2 ,PEEKW(plane1&)
10 POKEW mem&+6 ,PEEKW(plane1&+2)
11 POKEW mem&+10 ,PEEKW(plane2&)
12 POKEW mem&+14 ,PEEKW(plane2&+2)
13 copptr& = PEEKL(PEEKL(PEEKL(4)))+50
14 oldcop& = PEEKL(copptr&):POKE Copptr&,mem&
15 WHILE MOUSE(0) = 0 : WEND
16 POKEL copptr&,oldcop&
17 CALL FreeMem&(mem&,60)
18 LIBRARY CLOSE : END
19 DATA &he0,0,&he2,0,&he4,0,&he6,0,&h100,&ha000
20 DATA &h92,&h3c,&h94,&hd0,&h8e,&h2c81,&h90,&h29c8
21 DATA &h180,&hf00,&h6001,&hfffe,&h180,&hf0f0,
22 DATA &hf001,&hfffe,&h180,&h00f,&hffff,&hffff
```

Das Programm arbeitet wie in der Tabelle gezeigt. Eine Warnung: Wenn Sie versuchen, einen Screen nach unten zu ziehen, während das Programm läuft, schaltet Intuition automatisch auf die alte Copperliste um. *Brian Postma/ub*

- 3 Speicher für die Copperliste reservieren
Chip-Memory = 2
- 4 Programm beenden, wenn kein Speicher vorhanden ist
- 5 Copperliste in den reservierten Bereich kopieren
- 7 Zeiger auf die Bitplanes des aktuellen Screens berechnen
- 9 Zeiger in der Copperliste plazieren
- 13 Zeiger auf die aktuelle Copperliste aus GFX_Base ermitteln
- 14 Alten Zeiger retten, neue Liste initialisieren
- 15 Warten bis die linke Maus-Taste gedrückt wird
- 16 Alte Liste initialisieren
- 17 Speicher freigeben
- 18 Bibliothek schließen und Ende
- 19 Daten der Copperliste

Patch: Basic ohne Bilder

Viele Basic-Programmierer haben sich ihre »persönliche« Basic-Diskette angelegt — mit Autostart, Amiga-Basic und allen wichtigen »bmap«-Dateien. Auf der Diskette läßt sich Platz sparen, wenn man auf die Workbench verzichtet; Basic-Programme lassen sich auch vom CLI starten. Dann sind die Icons überflüssig. Man kann die »info«-Dateien löschen. Allerdings erzeugt Amiga-Basic immer neue Piktogramme, wenn man ein Programm sichert. Die Lösung hierfür ist ein Patch:

```
CLEAR, 50000&
NeueStelle$ = ""
For I = 1 TO 12
NeueStelle$ = NeueStelle$ + CHR$(0) :NEXT I
OPEN ":AmigaBasic" FOR INPUT AS #1 LEN=4096
OPEN ":NeuesBasic" FOR OUTPUT AS #2 LEN =4096
PRINT #2, INPUT$(H57a4, #1);
PRINT #2, INPUT$(H57a4, #1);
PRINT #2, INPUT$(H57a4, #1);
PRINT #2, INPUT$(H57a6, #1);
PRINT #2, NeueStelle$;
A$=INPUT$(12, #1)
PRINT a$
WHILE NOT EOF(1)
PRINT #2,INPUT$(1, #1);
WEND
CLOSE #1, #2
```

Dieser Patch ist für die Basic-Version mit einer Länge von 103500 Byte geeignet. Ein kleiner Nebeneffekt: Programme werden vom Basic-Interpreter schneller gespeichert. *Carsten Mallek/ub*

So liegen Sie richtig

In Basic ist immer das zuletzt definierte Fenster aktiviert. Möchte man nun in einem anderen Fenster Daten von der Tastatur einlesen, leitet WINDOW n die Eingabe entsprechend um. Bevor jedoch eine Eingabe erfolgen kann, muß das Fenster mit der Maus aktiviert werden. Der Aufruf der Intuition-Anweisung »ActivateWindow(&WindowHdl)« vor dem INPUT-Befehl macht diesen Handgriff überflüssig:

```
LIBRARY "intuition.library"
WINDOW 2, "Ein Test-Fenster", (10,10)-(100,100),15
CALL ActivateWindow(WINDOW(7))
INPUT a$
```

Das kurze Beispiel zeigt, wie Sie ein Fenster auch ohne Maus aktivieren. In der ersten Zeile steht ein Statement zum Öffnen der Intuition-Library. Eingefleischte Basic-Programmierer wissen selbstverständlich, daß dieser Befehl nur funktioniert, wenn sich die Datei »Intuition.bmap« im selben Verzeichnis wie das Basic-Programm oder im Ordner »s« der Startdiskette befindet. *Bruno Goldmann/ub*

Mehr Spaß mit C

Gute Reaktion im CLI

Manchmal möchte man die »Startup-Sequence« anhalten und auf eine Reaktion des Benutzers warten lassen. Es bietet sich an, einen CLI-Befehl zu schreiben. In C ist dies relativ einfach: Sie können das Makro »getchar« aus »stdio.h« verwenden:

```
#include <stdio.h>
main()
{
  getchar();
}
```

Das ist einfach. Das kleine Programm wartet auf das Betätigen einer Taste. Noch professioneller sieht ein Programm aus, das auf den Druck auf eine Maustaste wartet. Dafür gibt es noch kein Makro. Doch Sie können dem Programm über IDCMP eine Meldung schicken, wenn die Maustaste gedrückt wird. Hierzu brauchen Sie den Zeiger auf das Fenster, in welchem Ihr Programm läuft. Glücklicherweise kann man den Pointer (Zeiger) auf das aktuelle CLI-Fenster über Intuition-Base abfragen:

```
/* WaitMouse.c - Wartet auf Linke Maustaste */
#include <intuition/intuitionBase.h>
struct IntuitionBase *IntuitionBase;
struct Window *WindowPtr;
main()
{
  IntuitionBase = (struct IntuitionBase*)
    OpenLibrary("intuition.library",0);
  WindowPtr = IntuitionBase->ActiveWindow;
  ModifyIDCMP(WindowPtr,MOUSEBUTTONS);
  Wait(1 << WindowPtr->UserPort->mp_SigBit);
  ReplyMsg(GetMsg(WindowPtr->UserPort));
  ModifyIDCMP(WindowPtr,0);
  CloseLibrary(IntuitionBase);
}
```

In kompilierter Form (+L Option) kann man das Programm wunderschön als CLI-Befehl verwenden und in eigene Start-Sequenzen einbauen.

Rade Sotonica/ub

Anmeldeleiste für Geräte

Ein nützliches Tool für alle Programmierer ist das C-Programm »Device«. Es gibt die Namen und Adressen der Message-Ports aller dem Amiga angemeldeten Geräte aus.

```
#include <libraries/dosextens.h>

extern struct DosLibrary *DOSBase;
struct DeviceList *dev;

void main()
{
  dev = (struct DeviceList *) BADDR(((struct DosInfo *)
  BADDR(((struct RootNode *) (DOSBase->dl_Root))->rn_Info ))
  ->di_DevInfo);
  while (dev != 0)
  {
    switch ((int)(dev->dl_Type))
    {
      case DLT_DEVICE : printf("DEVICE : ");break;
      case DLT_DIRECTORY : printf("Directory : ");break;
      case DLT_VOLUME : printf("Volume : ");break;
      default : printf("Unknown : ");
    }
    printf("(Handler MsgPort at %6x) ",dev->dl_Task);
    Write(Output()
    ,BADDR(dev->dl_Name)+1
```

```
,(int) *((char *)BADDR(dev->dl_Name)));
  printf(":\n");
  dev = (struct DeviceList *) BADDR(dev->dl_Next);
}
}
```

Die Aufrufe zum Übersetzen und Linken mit dem Aztec-C-Compiler lauten:

```
cc device.c +L -S
ln device.o -LC32
```

Wenn Sie das C-Programm übersetzt haben, kopieren Sie den neuen Befehl »device« am besten in den C-Ordner Ihrer Workbench.

Peter Worofka/ub

Was läuft im Amiga?

Der System-Befehl STATUS zeigt leider nur alle Tasks an, die vom CLI aus gestartet wurden. Das C-Programm »task« zeigt dagegen alle Tasks an, die im Moment am Laufen sind:

```
#include <exec/execbase.h>
extern struct ExecBase *SysBase;
VOID Print_Task(task)
REGISTER struct Task *task;
{
  UBYTE state=task->tc_State;
  printf("%-20s %9ld %8ld ", task->tc_Node.ln_Name,
  (ULONG)task->tc_SPUpper-(ULONG)task->tc_SPLower-2L,
  (LONG)task->tc_Node.ln_Pri);
  if(state==TS_INVALID) printf("invalid\n");
  if(state==TS_ADDED) printf("added\n");
  if(state==TS_RUN) printf("running\n");
  if(state==TS_READY) printf("ready\n");
  if(state==TS_WAIT) printf("waiting\n");
  if(state==TS_EXCEPT) printf("exception\n");
  if(state==TS_REMOVED) printf("removed\n");
}
VOID main()
{
  register struct Task *task;
  SHORT tasks=1;
  APTR nodes[50], lastnode;
  register APTR node;

  printf("Task Name Stacksize Priority Status\n");
  printf("-----\n");
  Print_Task((APTR)SysBase->ThisTask);
  node=nodes;
  Disable();
  for(task=(struct Task *)SysBase->TaskWait.lh_Head;
  task->tc_Node.ln_Succ;
  *node=(APTR)task, node++,
  task=task->tc_Node.ln_Succ, tasks++);
  for(task=(struct Task *)SysBase->TaskReady.lh_Head;
  task->tc_Node.ln_Succ;
  *node=(APTR)task, node++,
  task=task->tc_Node.ln_Succ, tasks++);
  Enable();
  lastnode=node;
  for(node=nodes; node!=lastnode; node++)
    Print_Task(*node);
  printf("\n");
  printf("Active Tasks: %d\n", tasks);
}
/* Compiler- und Linkeraufrufe (Aztec-C) */
/* CC task.c +L */
/* ln task.o -lc */
```

Als erstes gibt »task« den Namen des Tasks aus, dahinter die Stackgröße, dann die Task-Priorität und zu guter Letzt den Status. Wie Sie vielleicht wissen, kann ein Task 7 verschiedene Zustände annehmen:

- | | |
|-------------|---|
| 1. invalid: | Dieser Task ist ungültig. |
| 2. added: | Der Task wurde gerade dem System hinzugefügt. |

- 3. running: Dies ist der Task, der gerade abgearbeitet wird. Es kann aber natürlich immer nur ein Task am Laufen sein.
- 4. ready: Dieser Task wird als nächster abgearbeitet.
- 5. waiting: Dieser Task ist im Moment am »Schlafen« und wird durch das System geweckt.
- 6. exception: Der Task löste einen Ausnahmezustand aus.
- 7. removed: Der Task wird gerade vom System »entbunden«.

Zum Schluß wird noch die Anzahl der im Moment laufenden Tasks ausgegeben. *Peter Fettke/ub*

Ist irgendwo ein Fenster auf?

Sie wollen einen Überblick der zur Zeit geöffneten Screens und Fenster? Nehmen Sie das Programm »Screen« zur Hilfe:

```
#include <intuition/intuition.h>
#include <intuition/intuitionbase.h>

struct IntuitionBase *IntuitionBase;
struct Screen *sc;
struct Window *wi;
void main()
{
  IntuitionBase = (struct IntuitionBase *)
  OpenLibrary("intuition.library",0);
  sc = IntuitionBase->FirstScreen;
  while (sc !=0)
  {
    printf("SCREEN :      (structure at %x) %s\n",sc,
    sc->Title);
    wi = sc->FirstWindow;
    while (wi != 0)
    {
      printf("      WINDOW : (structure at %x) %s\n",wi,
      wi->Title);
      wi = wi->NextWindow;
    }
    sc = sc->NextScreen;
  }
  CloseLibrary(IntuitionBase);
}
```

Das C-Programm listet alle vorhandenen Screens und Windows auf, indem es deren Namen und die Adressen der Kontrollstrukturen ausgibt. *Peter Worofka/ub*

Amiga sieht schwarz

Es macht keinen professionellen Eindruck, wenn der Anwender beobachten kann, wie ein Programm eine Grafik aufbaut. Man sollte in diesem Moment die Bildschirmdarstellung abschalten. Das kann in C über die Makros »ON_DISPLAY« und »OFF_DISPLAY« erreicht werden, die in der Include-Datei »gfxmacros.h« definiert sind.

```
#include <graphics/gfxmacros.h>
main()
{
  OFF_DISPLAY();
  Delay(100);
  ON_DISPLAY();
}
```

Das Beispiel zeigt, wie man die Makros einsetzt.

Thomas Wagner/ub

Kurze Unterbrechung

Im Leben geschieht es immer wieder, daß Sie eine Arbeit kurzfristig unterbrechen müssen, weil etwas Dringendes ansteht. Auch im Computer gibt es Aufgaben, die so wichtig sind, daß der Prozessor für einen Augenblick alles stehen und liegen läßt. Diese Form der Unterbrechung, Interrupt genannt, wird im Amiga fast ausschließlich vom Betriebssystem genutzt.

Es gibt jedoch viele Gründe, die Interrupts des MC 68000 auch in eigenen Programmen zu nutzen. Besonders nützlich ist dabei der Vertical-Blank-Interrupt (VBI), der immer auftritt, wenn der

Elektronenstrahl des Monitors zur linken, oberen Ecke zurückkehrt. Da das Bild in einer Sekunde 50mal aufgebaut wird, gibt es jede fünfzigste Sekunde einen VBI. Er kann dazu verwendet werden, Änderungen der Grafik mit dem Bildschirmaufbau zu synchronisieren. Außerdem ist er für alle Aufgaben geeignet, die in regelmäßigen Abständen ausgeführt werden müssen, beispielsweise erfolgt die Steuerung des Mauszeigers im VBI.

Damit mehrere parallel laufende Tasks den VBI nutzen können, wird er über einen Interrupt-Server verwaltet. Jedes Programm, das eine eigene Routine in den VBI »einklinken« will, muß eine

```
#include "exec/memory.h"
#include "exec/interrupts.h"
#include "hardware/intbits.h"
```

```
struct Interrupt *VBI;
UWORD counter=0x0f00;
```

```
main()
{
  extern void VBIServer();

  VBI=(struct Interrupt *)AllocMem
  (sizeof(struct Interrupt),MEMF_CHIPMEMF_PUBLIC);
  if(VBI==NULL)exit(-1);
  VBI->is_Node.ln_Type=NT_INTERRUPT;
  VBI->is_Node.ln_Pri=0;
  VBI->is_Node.ln_Name="VBI-Demo";
  VBI->is_Data=(APTR)&counter;
  VBI->is_Code=VBIServer;

  AddIntServer(INTB_VERTB,VBI);

  Delay(200); /* warten ... */

  RemIntServer(INTB_VERTB,VBI);
  FreeMem(VBI,sizeof(struct Interrupt));
}
```

```
#asm
public _VBIServer
_VBIServer:
  movem.l a2-a6/d2-d7,-(sp)
  move.l a1,a0

  sub.w #0100,(a0)
  cmpi.w #0,(a0)
  bne.s _VBI1
  move.w #0f00,(a0)
_VBI1:  move.w (a0),$dff180

  moveq.l #0,d0
  movem.l (sp)+,a2-a6/d2-d7
  rts

#endasm
```

Listing 2. Beispiel für einen Interrupt-Server in C

Interrupt-Struktur anlegen. Die Interrupt-Struktur enthält eine Node-Struktur (Knoten), über die sie beim Aufruf der Exec-Funktion AddIntServer() in die Interrupt-Server-Liste eingebunden wird. Zusätzlich enthält die Interrupt-Struktur die Zeiger »is_Code« und »is_Data«, in denen man die Adresse der eigentlichen Routine und eines eventuell benötigten Datenfeldes einträgt.

Der Knoten enthält ebenfalls einige Felder, die vor dem Aufruf von AddIntServer() zu initialisieren sind. Im Feld »ln_Type« wird festgelegt, um was für eine Node es sich handelt. In unserem Fall ist es der Typ »NT_INTERRUPT«. Das Feld »ln.Pri« bestimmt die Priorität der Routine. Je höher sie ist (0-255), desto früher wird das Programm nach Auftreten des Interrupts abgearbeitet. Im Feld »ln_Name« kann ein Zeiger auf den Namen des Interrupts stehen, es hat jedoch nur Bedeutung beim Debuggen und kann auf »NULL« gesetzt werden.

Listing 2 enthält ein Beispiel für den Einsatz eines VBI-Servers. Der Interrupt-Server wurde in Assembler geschrieben, während die Initialisierung in C erfolgt. *Emanuel Tepass/C. Kögler/sq*

Tips zu Assembler

PRINT in Assembler

Befehle wie PRINT oder »printf()« erleichtern in Hochsprachen die Ausgabe eines Textes auf dem Bildschirm. Wohlgermerkt, in Hochsprachen — Assembler-Programmierer sind gezwungen, den Text über die Funktion »Write« aus der »dos.library« auszugeben. Dies sieht in den meisten Programmen etwa so aus (Beispiel für Devpac-Assembler):

```
....
....
CALLDOS Output
move.l d0,d1
move.l #textanfng,d2
move.l #textende-textanfng,d3
CALLDOS Write
....
....
textanfng:
dc.b "alloha"
textende:
```

Die Übersichtlichkeit eines Listings geht bei dieser Methode verloren, da die Ausgaberroutine und der zugehörige Text an unterschiedlichen Stellen im Programm stehen. Das Unterprogramm »Print« schafft Abhilfe. Wir zeigen seine Anwendung an einem Beispiel für den Devpac-Assembler:

```
indir ":include/"

include exec/exec_lib.i
include libraries/dos_lib.i
include libraries/dos.i

* dos.library öffnen
start:
move.l #dosname,a1
moveq #0,d0           alle Versionen
CALLEXEC OpenLibrary
tst.l d0
beq raushier         keine dos.library
move.l d0,_DOSBase   Zeiger retten

* Achtung, Trick 17
bsr Print
dc.b "Hallo Welt",10,0
cnop 0,2

* dos.library schließen
move.l _DOSBase,a1
CALLEXEC CloseLibrary

raushier:
rts                 auf Wiedersehen!
_DOSBase dc.l 0      Zeiger
dosname  DOSNAME

* hier beginnt die Unterroutine
Print:
move.l (sp),a0
move.l a0,d2
Loop:
tst.b (a0)+
bne.s Loop
move.l a0,d0
addq.l #1,d0
bclr #0,d0
move.l d0,(sp)
move.l a0,d3
sub.l d2,d3

* Aufruf der DOS-Funktion Write
move.l _DOSBase,a6
CALLDOS Output
```

```
move.l d0,d1
CALLDOS Write
rts
```

Folgender Trick kommt zum Einsatz: Der MC 68000 legt beim Aufruf eines Unterprogramms eine Rücksprungadresse als Langwort auf den Stapel (Stack). Die Adresse zeigt auf den Maschinenbefehl hinter dem Sprungbefehl. In unserem Fall befindet sich an dieser Stelle der auszugebende Text. Die Unterroutine »Print« holt sich diese Adresse vom Stapel, ermittelt die Länge des Textes und gibt ihn aus. Nun muß die Rücksprungadresse durch Addition der Textlänge korrigiert werden. Sie wird zusätzlich auf Wortlänge justiert, um einen Fehler durch Sprung auf eine ungerade Adresse zu verhindern. Wir fügen beim Devpac-Assembler hierzu die Anweisung »cnop 0,2« hinter der Definition des Strings ein. Wichtig: ein Text muß mit einer »0« enden. Der vollständige Aufruf von »Print« lautet:

```
bsr Print
dc.b "Text",0
cnop 0,2
```

In unserem Beispiel verwenden wir zusätzlich:

```
dc.b "Text",10,0
```

Der Amiga gibt in diesem Fall hinter dem Text einen Zeilenvorschub (»10«) aus.

Die Idee, die Rücksprungadresse auf dem Stack um die Länge eines Textes zu erhöhen, hatte der Amerikaner Glen Bredon. Er machte ähnliche Routinen in 6502-Assembler auf dem C64 und Apple populär.

Jörg Schmidt/ub

Falscher Fehler

Ruft man eine selbstgeschriebene Assembler-Routine aus der »Startup-Sequence« auf, kann es beim Rücksprung (»rts«) zu einer Fehlermeldung kommen, obwohl das Programm einwandfrei arbeitet. Die Ursache ist im Prozessor-Register d0 zu suchen. Enthält es nach Ablauf eines Programms einen Wert ungleich null, nimmt das CLI (Command Line Interpreter) an, daß ein Fehler aufgetreten ist. Ihr Programm sollte daher vor dem Rücksprung den Inhalt des Registers d0 löschen:

```
ende:
clr.l d0 ; Register löschen
rts ; und zurück
```

Die durch einen Rückgabewert ungleich null erzeugten Fehlermeldungen, »Returncode« genannt, spielen beim Ablauf von Stapeldateien wie der »Startup-Sequence« eine Rolle: Mit dem CLI-Befehl FAILAT legt man fest, ab welchem Rückgabewert die Bearbeitung einer Stapeldatei abgebrochen wird.

Immo Müller de Vries/C. Kögler

Die Maus sieht so anders aus

Sie suchen einen Weg, um das Aussehen des Mauszeigers zu verändern? »MouseChange« ist die Lösung für alle Assembler-Programmierer:

```
ExecBase = 4
OpenLibrary = -408
CloseLibrary = -414
AllocMem = -198
FreeMem = -210
SetPointer = -270
ClearPointer = -30-30
start:
move.l ExecBase, a6 ; Intuition öffnen
lea intname, a1
jsr OpenLibrary(a6)
```



```

move.l d0, intbase
; Speicher für Mausdaten anfordern (Chip-Ram !)
move.l #[enddata-sprdata], d0
moveq #2, d1 ; 2 = ChipRam
move.l ExecBase, a6
jsr AllocMem(a6)
move.l d0, dest
beq notallocated
move.l #[enddata-sprdata-1], d0
lea sprdata, a0
move.l dest, a1
copy:
move.b (a0)+, (a1)+ ; Mausdaten kopieren
dbf d0, copy
move.l intbase, a6
move.l 52(a6), a0 ; ActivWindow = (IntBase+52)
move.l dest, a1 ; Zeiger auf Datenliste
moveq #16, d0 ; Höhe der Maus
moveq #16, d1 ; Breite der Maus
moveq #-7, d2 ; X-Offset Ansprechpunkt
moveq #-7, d3 ; Y-Offset Ansprechpunkt
jsr SetPointer(a6) ; Maus verändern
wait:
btst #6, $bfe001 ; Auf linke Maustaste warten
bne wait
move.l 52(a6), a0 ; Window = (IntBase+52)
jsr ClearPointer(a6) ; Alte Maus aktivieren
move.l dest, a1
moveq #[enddata-sprdata], d0
move.l ExecBase, a6
jsr FreeMem(a6) ; Speicher freigeben
notallocated:
move.l intbase, a1 ; Intuition schließen
jsr CloseLibrary(a6)
rts
even
sprdata:
; Bitplane: 1 2
dc.w 0,0
dc.w %0000010000100000 , %0000001111000000
dc.w %0000110000110000 , %0000000110000000
dc.w %0001100000011000 , %0000000110000000
dc.w %0011000000001100 , %0000000110000000
dc.w %0110000000000110 , %0000000110000000
dc.w %1100000000000011 , %0000001111000000
dc.w %0000000000000000 , %1000011001100001
dc.w %0000000110000000 , %1111110000111111
dc.w %0000000110000000 , %1111110000111111
dc.w %0000000000000000 , %1000011001100001
dc.w %1100000000000011 , %0000001111000000
dc.w %01100000000000110 , %0000000110000000
dc.w %00110000000001100 , %0000000110000000
dc.w %0001100000011000 , %0000000110000000
dc.w %0000110000110000 , %0000000110000000
dc.w %0000010000100000 , %0000001111000000
dc.w 0,0
enddata:
dest: dc.l 0
intbase: dc.l 0
intname: dc.b 'intuition.library',0

```

Erkennen Sie den neuen Mauszeiger bereits im Listing? Das wesentliche Merkmal des Programms ist die Verwendung der Routinen »SetPointer« und »ClearPointer« aus der »intuition.library«. Das Programm ist für den Seka-Assembler geschrieben. Besitzer eines anderen Assemblers können es jedoch im wesentlichen unverändert übernehmen. Nach Verändern des Zeigers

wartet »MouseChange«, bis Sie die linke Maus-Taste betätigen; danach sieht die Maus wieder wie vorher aus.

Günter Auwärter/ub

Alle Offsets für Assembler

Für Assembler-Programmierer, die häufig Systemroutinen benutzen, ist es oft mühsam, die Offsets für die Einsprünge in die Libraries zu berechnen beziehungsweise abzutippen; und nicht jeder Assembler stellt die Offsets in Form von Include-Dateien zur Verfügung. Es gibt einen relativ einfachen Weg, sich solche Include-Dateien selber anzufertigen: Man kann die FD-Dateien auf der Extras-Diskette in ein assemblerlesliches File verwandeln. Das folgende Basic-Programm zeigt, wie man eine solche Konvertierung vornimmt. Die Einbindung der Offsets in eigene Programme sollte dann jeder Programmierer selbst vornehmen können.

Timmo Stange/ub

```

INPUT "FD-Datei (mit Pfad)",Source$
INPUT "Ziel-Datei (ebenfalls mit Pfad)",Dest$
OPEN "I",1,Source$
OPEN "O",2,Dest$
WHILE NOT(EOF(1))
Zeile$ = "" : z$ = ""
WHILE z$ <> CHR$(10) AND NOT (EOF(1))
z$ = INPUT$(1,1)
Zeile$ = Zeile$+z$
WEND
z$ = LEFT$(Zeile$,1)
IF z$ <> "*" AND z$ <> "#" AND LEN(Zeile$) > 1 THEN
PRINT #2,LEFT$(Zeile$,INSTR(Zeile$,"(")-1);" EQU ";counter
counter = counter-6
ELSE
IF z$ = "*" THEN PRINT #2,";";Zeile$
IF z$ = "#" AND MID$(Zeile$,3,4) = "bias" THEN
counter = -VAL(MID$(Zeile$,8,2))
END IF
WEND
CLOSE 1
CLOSE 2

```

Immer wieder, immer wieder...

Rekursion ist, wenn eine Routine sich selbst aufruft. Dies geschieht solange, bis eine bestimmte Bedingung erfüllt ist. Diese Technik der Programmierung erfordert viel Umsicht. Leicht verstrickt sich der Computer in einer endlosen Schleife. Rekursive Programmierung trifft man häufig in Hochsprachen wie C oder Prolog an. Eine sinnvolle Anwendung rekursiver Programmierung in Assembler zeigt das folgende Beispiel; ein Sprite soll um 180 Grad gedreht werden:

```

invers:
cmp.l #0,(a0) ; letztes Kontroll-Wort er
reicht?
beq inv_ende ; Ja, Abbruch der Rekursion
move.l (a0)+,-(SP) ; zwei Daten worte auf Stack
jsr invers ; und noch eine Runde
move.l (SP)+,(a1)+ ; Spritedaten umgekehrt in
Datenfeld schreiben

```

```

inv_ende:
rts

```

Bevor die Routine aufgerufen werden kann, muß sich in den Adreßregistern a0 und a1 ein Zeiger auf den Anfang der Sprite-Daten befinden. Dies wäre folgendermaßen realisierbar:

```

lea.l sprite_date+4,a0 ; Anfangsadresse des
2. Worts
move.l a0,a1 ; Merker auf Anfang nach a1
jsr invers

```

Bei Rekursionen wird, wie an dem Beispiel gezeigt, laufend eine Routine mit JSR aufgerufen. Der Stack muß also groß genug sein, um die Rücksprungadressen aufzunehmen.

Bernd Schönberger/ub

Basteln mit Modula

Modula-2 sucht Devices

Modula-2-Programmierer haben es ebenfalls leicht, Informationen über die angeschlossenen Devices zu erhalten. Das folgende Listing für den Modula-2-Compiler M2Amiga zeigt einen Weg:

```
MODULE Devices;
IMPORT Dos;
FROM Dos      IMPORT DeviceListPtr, DeviceListType,
                DosLibraryPtr;
FROM Strings  IMPORT Copy;
FROM SYSTEM  IMPORT ADR;
FROM Terminal IMPORT WriteString, WriteLn;
VAR
  dosPtr: DosLibraryPtr;
  devPtr: DeviceListPtr;
  devName: ARRAY[0..9] OF CHAR;
BEGIN
  dosPtr := ADR(Dos);
  devPtr := dosPtr.root!.info!.devInfo;
  WHILE devPtr # NIL DO
    IF devPtr.type = device THEN
      Copy(devName, devPtr.name!, 1, INTEGER
(devPtr.name![0]));
      WriteString(devName); WriteString(' ');
    END;
    devPtr := devPtr.next
  END;
  WriteLn();
END Devices.
```

Zur Vorgehensweise: Geben Sie das Listing mit einem Editor ein. Sie sollten M2Emacs wählen, der zum M2Amiga-Paket gehört. Stellen Sie im Menü »Optionen« den Punkt »erzeuge Ikonen« ein. Wenn Sie nun das Programm — geben wir ihm den Namen »Devices.mod« — speichern, erzeugt der Editor im aktuellen Fenster ein Symbol (Icon) für die Programmdatei. Das Icon wird sichtbar, wenn Sie das aktuelle Fenster schließen und wieder öffnen.

Das Übersetzen mit dem Compiler ist nun denkbar einfach: zuerst ein Mausklick auf das Symbol mit Namen »Devices.mod«, dann eine < Shift >-Taste halten und ein Doppelklick auf das Symbol des Übersetzers, »m2c«. Der Compiler erzeugt nun den Objektcode »Devices.obj«. Diese Datei besitzt ebenfalls ein Symbol auf der Workbench. Es erscheint, wenn Sie das Fenster abermals schließen und öffnen. Ein doppelter Mausklick auf das Icon und der Linker generiert das lauffähige Programm.

Eine Anregung zum Programm: Wer experimentierfreudig ist, sollte das Listing als Grundstock für eigene Routinen in Zusammenhang mit den Devices wählen (Informationen zu Devices finden Sie im Insider-Kurs, Seite 110). Wie wär's, wenn Sie beispielsweise die Ausgabe der Device-Namen auf dem Bildschirm ausbauen?

Edgar Schwarzlub

Fensteroperationen in Modula-2

Nachdem im AMIGA-Magazin bereits ein C-Programm veröffentlicht wurde, das dem Benutzer das Vergrößern eines CLI-Fensters auf PAL-Größe abnimmt, hier nun eine Version für Modula-2-Programmierer:

```
MODULE Size;
FROM Intuition IMPORT OpenIntuition, IntuitionBasePtr,
                WindowPtr, SizeWindow;
FROM SYSTEM  IMPORT ADR;
VAR IntuiPtr : IntuitionBasePtr;
    WinPtr   : WindowPtr;
```

```
BEGIN
  IntuiPtr := OpenIntuition();
  WinPtr   := IntuiPtr!.activeWindow;
  SizeWindow(WinPtr,0,56);
END Size.
```

Das Programm holt zunächst mit Hilfe der Funktion OpenIntuition() einen Zeiger auf die Struktur IntuitionBase. In dieser Struktur steht unter ActiveWindow ein Zeiger auf das aktuelle Fenster. Mit SizeWindow() ändert man die Größe des Windows.

Mit Hilfe des Zeigers auf das aktuelle Fenster lassen sich im Programm natürlich auch andere Dinge mit dem Fenster »anstellen«. Hier sind Ihrer Experimentierfreudigkeit keine Grenzen gesetzt.

Stefan Kaiserlub

Modula-2 geht den direkten Weg

Da die Abfrage der Maustasten über das Betriebssystem des Amigas umständlich zu handhaben ist, fragen viele Programme die Hardware-Register direkt ab. Eine solche Abfrage ist auch in Modula-2 möglich. Die beiden folgenden Routinen wurden für den Modula-2-Compiler M2Amiga geschrieben. Sie arbeiten auch mit der PD-Version von der Fish-Disk 113. Die beiden Prozeduren sind als Modul ausgelegt und können durch:

```
FROM MouseButtons IMPORT leftMouseButton, rightMouseButton;
in ein eigenes Programm importiert werden. Man kann die Prozeduren auch direkt ins Programmmodul schreiben, sie sind dann aber nicht mehr für andere Module verfügbar. Zunächst das Definitionsmodul:
```

```
DEFINITION MODULE MouseButtons;

(* Abfrage der rechten und linken Maustaste *)

(* TRUE : rechte Maustaste gedrückt *)
PROCEDURE rightMouseButton() : BOOLEAN;

(* TRUE : linke Maustaste gedrückt *)
PROCEDURE leftMouseButton() : BOOLEAN;

END MouseButtons.
```

Anschließend folgt das Implementationsmodul:

```
IMPLEMENTATION MODULE MouseButtons

FROM SYSTEM IMPORT BITSET;

PROCEDURE rightMouseButton() : BOOLEAN;
VAR
  POTGOR[ODFF016H] : BITSET;
BEGIN
  IF (10 IN POTGOR) THEN RETURN FALSE ELSE RETURN TRUE END;
END rightMouseButton;

PROCEDURE leftMouseButton() : BOOLEAN;
VAR
  CIA[0BFEO01H] : SET OF [0..7];
  (* Adresse eines CIA-Registers *)
BEGIN
  (* wenn 7 Bit nicht gesetzt, ist die M-Taste gedrückt *)
  IF (6 IN CIA) THEN RETURN FALSE ELSE RETURN TRUE END;
END leftMouseButton;
END MouseButtons.
```

Das Demo-Programm zeigt die Anwendung des neuen Moduls:

```
MODULE MouseButtonDemo;
(* Demonstration des Moduls MouseButtons *)

FROM MouseButtons IMPORT leftMouseButton, rightMouseButton;
FROM Terminal IMPORT WriteString, WriteLn;

BEGIN
LOOP
  IF leftMouseButton() THEN
```



```
WriteString('linke Maustaste');WriteLn;
END;
IF rightMouseButton() THEN
  WriteString('rechte Maustaste');WriteLn;
  EXIT;
END;
END;
WriteString('Ende'); WriteLn;
END MouseButtonDemo.
```

Das Demo-Programm besteht aus einer Endlosschleife. Sobald Sie die linke Maustaste drücken, gibt der Amiga eine Meldung auf dem Bildschirm aus. Drücken Sie die rechte Maustaste, wird die Schleife verlassen. Zur Funktionsweise der Prozeduren: Durch Angabe der Speicherstelle in eckigen Klammern hinter der Variablen wird eine MengenvARIABLE auf ein Register des I/O-Bausteins gesetzt. Die Mengeninhalte entsprechen den Bits des Registers. Somit kann der Zustand bestimmter Bits im Register leicht abgefragt werden.

Martin Murnleiter/ub

Modula-2 auf den Spuren des CLI

Die Befehle des CLI lassen sich mit der Funktion »Execute« aus der »dos.library« aufrufen. Das gilt natürlich auch für Modula-2-Programme. Ein Beispiel zeigt, wie man dabei vorgeht:

```
MODULE ExecuteTest ;

FROM SYSTEM IMPORT ADDRESS,ADR ;
FROM Dos      IMPORT FileHandlePtr,Input,Output,
                  Execute ;
```

```
VAR in,out : FileHandlePtr ;
    test  : INTEGER ;
```

```
BEGIN (* Demo *)
```

```
in := Input() ;
out := Output() ;
test := Execute(ADR("cd df1:"),in,out) ;
test := Execute(ADR("list"),in,out) ;
END ExecuteTest .
```

Das Programm zeigt den Aufruf der Befehle »CD df1:« und »LIST«. Die Parameter »in« und »out« sollten beim Aufruf von »Execute Test« aus dem CLI durch »0« ersetzt werden, jede Ausgabe bezieht sich dann auf das aktuelle CLI-Fenster. Probieren Sie das Ganze ruhig einmal mit anderen CLI-Befehlen aus. Noch eine Bemerkung zum Schluß: Laut ROM-Kernel-Manual soll der Funktionswert, den Execute liefert, vom Typ BOOLEAN sein, allerdings arbeitet die Funktion tatsächlich mit dem Typ INTEGER.

Stefan Kaiser/ub

Parole: Modula-2

Alle reden von Datenschutz — Sie auch? Dann schauen Sie sich das folgende Modula-2-Programm (Listing 1) an: Es fragt den Benutzer nach einem Paßwort und läßt den Zugang zum System nur bei korrekter Eingabe zu.

Geben Sie das Programm mit einem Texteditor ein. Nachdem der Quellcode kompiliert und gelinkt wurde, steht das Programm »Passwort« zur Verfügung. Es fühlt sich im Verzeichnis »C« einer Start-Diskette (Boot-Diskette) »heimisch«, wo es auch hinkopiert werden sollte. Binden Sie den Aufruf des Programms in die »Startup-Sequence« einer bootfähigen Disk als erste Anweisung ein. Wenn der Amiga mit dieser Diskette startet, fragt er den Anwender zunächst nach dem Paßwort; einmal darf man die falsche Lösung eingeben, beim zweiten Mal blockiert der Amiga weitere Zugriffe zum System. Der »Spion« ist gezwungen, den Amiga durch Drücken der Tastenkombination <Amiga links Amiga rechts Ctrl> neu zu starten. Wird das richtige Paßwort eingegeben, erscheint eine kurze Meldung, und der Bootvorgang wird fortgesetzt. Als Paßwort sind Zahlen im Bereich von 0 bis 4294967295 (LONGCARD) wählbar. Sie geben Ihr persönliches Kennwort im Listing an. Voreingestellt ist 666666.

Norbert Cohen/ub

```
MODULE Passwort; (* kann mit der Demo-Version M2Amiga
                  kompiliert und gelinkt werden *)
FROM SYSTEM IMPORT ADR;
FROM InOut IMPORT Write,WriteLn,WriteString,ReadLongCard;
FROM Intuition IMPORT DisplayAlert;
VAR Eingabe, Passwort, Zaehl : LONGCARD ;
    al: RECORD
      x1: CARDINAL; s1: ARRAY [0..71] OF CHAR;
      x2: CARDINAL; s2: ARRAY [0..56] OF CHAR;
    END (* RECORD *) ;
PROCEDURE C1s;
BEGIN
  Write(CHR(12));
END C1s;
BEGIN (* Hauptprogramm *)
  Passwort:= 666666;
  C1s;
  REPEAT
    Zaehl:= Zaehl + 1;
    WriteString(" Bitte Passwort eingeben!!!"); WriteLn;
    WriteString(" ");
    ReadLongCard(Eingabe);
    IF Eingabe <> Passwort THEN
      WriteLn;
      WriteString(" Passwort ist falsch."); WriteLn;
      WriteString(" Sie haben noch einen Versuch."); WriteLn;
      ELSE Zaehl:= 2
    END (* IF *) ;
  UNTIL (Zaehl = 2);
  IF Eingabe <> Passwort THEN
    WHILE TRUE DO
      al.s1:= " Keine Zulassung zum System. ";
      al.x1:=40; al.s1[0]:=CHAR(100); al.s1[71]:=CHAR(1);
      al.s2:= " System muß neu hochgefahren werden!!!";
      al.x2:=146; al.s2[0]:=CHAR(180); al.s2[56]:=CHAR(0);
      IF DisplayAlert(0, ADR(al),256) THEN END;
    END (* WHILE *) ;
  ELSE
    WriteLn;
    WriteString(" Passwort ist korrekt!!"); WriteLn;
  END (* IF *) ;
END Passwort .
```

Listing 1. Ein Programm in Modula-2, das zeigt, wie man Eingaben von der Tastatur liest.

Hardware-Zugriff in Modula-2

Für den direkten Zugriff auf die Hardware des Amiga bietet das Modula-2-System von Meyer Vogt das Modul Hardware. Dieses Modul, es fehlt auf der Public Domain-Version (Fish-Disk 113), verlängert allerdings den ausführbaren Code erheblich. Es gibt aber noch einen anderen, einfacheren Weg, um den Inhalt von Hardware-Registern zu beeinflussen: Man definiert eine Variable an einer festen Adresse. Die Syntax hierfür sieht folgendermaßen aus:

```
VAR Name [Adresse] : TYPE
```

Ein Beispiel zeigt die Details, auf die Sie achten müssen:

```
MODULE HardwareDemo ;
VAR VHPOSR [ODFF006H],
    COLOR00 [ODFF180H] : CARDINAL ;
    i : LONGCARD ;
BEGIN (* HardwareDemo*)
  FOR i := 0 TO 100000 DO
    COLOR00 := VHPOSR ;
  END (*FOR*) ;
END HardwareDemo .
```

Das Programm liest die Rasterstrahlposition aus dem Register VHPOSR und schreibt den Wert in das Register COLOR00, das für die Farbe des Hintergrunds verantwortlich ist. Daraus resultiert ein buntes Durcheinander von Farben auf dem Bildschirm.

Stefan Kaiser/ub

Durchblick bei Amiga-DOS

Steuersequenzen des CLI

Kennen Sie die Steuersequenzen des CLI? Die Tabelle gibt einen Überblick. Dabei steht »ESC« für die Escape-Taste. Bei einem ECHO-Befehl muß ESC durch *e ersetzt werden! *Rolf Beck/ub*

ESC [<n> u	n Zeichen pro Zeile
ESC [<n> x	linker Rand in Pixeln
ESC [<n> y	oberer Rand in Pixeln
ESC [<n> t	Anzahl der Zeilen
ESC [c	wieder normale Werte
ESC [0 <space> p	Cursor aus
ESC [<space> p	Cursor ein
ESC [<y> ; <x> H	Cursor an Position (x;y) setzen
ESC [<n> A	Cursor n Zeichen hoch
ESC [<n> B	Cursor n Zeichen runter
ESC [<n> C	rechts
ESC [<n> D	links
ESC [J	Fenster ab Cursorposition löschen
ESC [K	Zeile ab Cursorposition löschen
ESC [<a> m	setzen von
ESC [<a> ; m	Schriftstil und Farbe
ESC [<a> ; ; <c> m	
a = 0	normal
1	fett
2	schwarz
3	kursiv
4	unterstrichen
7	invers
b = 30,31,32,33	Vordergrund blau, weiß, schwarz, orange
c = 40,41,42,43	Hintergrund blau, weiß, schwarz, orange
CTRL H	Zeichen löschen
CTRL I	Tabulator
CTRL M	Return
CTRL N	alternativer Zeichensatz
CTRL O	originaler Zeichensatz
CTRL L	Bildschirm löschen
CTRL G	Blitz
CTRL J	Cursor down
CTRL K	Cursor up
Left-Amiga V	Retry im Requester
Left-Amiga B	Cancel im Requester

Die Steuersequenzen des CLI im Überblick

Festplatten: freie Auswahl

Sicher haben sich schon viele Amiga2000-Benutzer mit PC-Karte und Festplatte darüber geärgert, daß in vielen Requestern zur Auswahl von Laufwerken zwar »DH0:« oder »DH1:« erscheinen, aber in den allerwenigsten Programmen an die Benutzer der PC-Festplatte mit Amiga-Partition gedacht wird. Um ein Anklicken mit der Maus zu ermöglichen, müssen Sie die als »JH0:« oder »JH1:« formatierte Festplatte mit dem Namen »DH0:« oder »DH1:« versehen. Dies kann nachträglich mit dem Workbench-Menüpunkt »Rename« geschehen oder bereits beim Formatieren:

```
dpformat drive jh0: name "DH0"
```

Analog sieht der Befehl JH1: aus. Dieser Tip erspart den Betroffenen auf jeden Fall eine Menge Tipparbeit und erleichtert den Umgang mit der Festplatte. (Dieter Schnepf/ub)

Leerstellen in Filenamen

Häufig tauchen in Dateinamen Leerstellen auf. Diese Dateien lassen sich im CLI nur ansprechen, wenn Sie den Namen in Anführungszeichen setzen:

```
RUN "Programmname mit Space"
COPY programm TO "ein Beispiel"
```

Auf diese Art können Sie auch eigene Dateinamen mit Leerzeichen im Namen speichern. Eine Variante ohne Anführungszeichen ist die Verwendung der Tastenkombination <ALT> und <SPACE>. Beachten Sie hierbei, daß Sie beim Laden eines solchen Files ebenfalls diese Kombination eingeben müssen. Ein Trick, der auch zum Schutz einer Datei eingesetzt werden kann. Auch viele Anwenderprogramme, in denen der Name in Dialogfenstern eingegeben wird, erlauben Leerzeichen im Text. Sollte dies nicht möglich sein, hilft es ebenfalls, statt Leerzeichen die Tastenkombination <ALT SPACE> zu verwenden.

Bernd Eckert/ub

Nur ein Zylinder

Wenn man sich mit dem CLI beschäftigen möchte, aber nur ein Laufwerk hat, ist es ratsam die CLI-Befehle, die der Amiga normalerweise aus dem Verzeichnis »C« liest, in der RAM-Disk abzulegen. Hierzu schreibt man eine Befehlsdatei (Batch-File), in der alle notwendigen Anweisungen stehen. Dies geschieht mit der Anweisung: »ED RAM-Disk« <RETURN>. Dieser Befehl startet den Editor. Geben Sie nun nacheinander folgende Befehle ein. (RETURN nach jeder Zeile nicht vergessen):

```
PROMPT "RAM %n> " Ändert die CLI-Bereitschaftsanzeige
MAKEDIR ram:c ' Richtet C-Directory im RAM ein
COPY c/cd to ram:c ' Kopiert den Befehl CD in die RAM-Disk
COPY c/Dir to ram:c ' Kopiert den Befehl DIR in die RAM-Disk
COPY c/... ,
```

Jeder sollte in der Liste die Befehle ergänzen, von denen er meint, daß sie benötigt werden. Dabei muß man sich darüber im Klaren sein, daß jeder überflüssige Befehl im RAM kostbaren Speicher verbraucht.

```
PATH ram:c
```

Der letzte Befehl veranlaßt den Amiga, Befehle zuerst in der RAM-Disk zu suchen. Wer viele Befehle kopiert, faßt mehrere in einem Arbeitsgang zusammen:

```
COPY df0:c/(COPY | ECHO | TYPE | DIR | LIST) ram:c
```

Der Editor wird durch die Tastenkombination <ESC> <X> <RETURN> verlassen. Ist die Diskette nicht schreibgeschützt, wird die Datei »RAM-Disk« gespeichert. Wenn die RAM-Disk nun durch »Execute RAM-Disk« aktiviert wird, ist der Anwender in der Lage, sich auch auf anderen Disketten umzusehen. Er ist nicht mehr von der Arbeits-Diskette abhängig. Der Aufruf der RAM-Disk kann natürlich gleich in der »Startup-Sequence« untergebracht werden.

Sollten Sie einmal einen Befehl aufrufen, der nicht im RAM vorhanden ist, sucht der Amiga im C-Directory der Startdiskette. Ist diese gerade nicht eingelegt, macht der Amiga durch einen Requester auf das Versäumnis aufmerksam.

ED lenkt um auf Schreibmaschine

Für Besitzer eines Druckers, die diesen als Schreibmaschine verwenden möchten, ist der folgende Rat. Erstellen Sie sich Ihr eigenes Textverarbeitungsprogramm und geben folgendes ein:

```
Ed ram:NAME < RETURN >
```

Nun kann man den Text in Ruhe eingeben. Bei der späteren Durchsicht auf Schreibfehler, sind diese einfach zu korrigieren. Um den Text zu Papier zu bringen, reicht der Aufruf:

```
< ESC > sa "prt:"
```

Nun wird der Text gedruckt. Möchten Sie den Text auf Diskette speichern, drücken Sie:

```
< ESC > sa "DF0:Name des Textes"
```

Möchten Sie Ihr »Textverarbeitungsprogramm« ohne zu speichern verlassen, drücken Sie einfach <ESC> Q.

Norbert Cohen/ub

Formatieren ohne Nachtanken

Haben Sie sich schon einmal darüber geärgert, daß der Amiga immer wieder auf die Workbench zugreifen möchte, wenn Sie Disketten formatieren oder kopieren wollen? Es gibt aber einen Weg, ohne die Workbench auszukommen. Kopieren Sie die beiden erforderlichen Programme ebenfalls in die RAM-Disk. Wieder einmal schreiben wir eine Befehls-Datei »ED INDI«. Dabei darf man nicht vergessen, nach jeder Zeile die RETURN-Taste zu drücken. Geben Sie folgendes ein:

```
Makedir ram:System 'Directory erstellen.  
Copy System/FORMAT ram:System ins RAM kopieren  
Copy System/DISKCOPY ram:System 'DISKCOPY kopieren  
ASSIGN SYS: RAM: 'Verzeichnis umlenken
```

Dieses Batchfile wird durch »Execute INDI« aktiviert. Legen Sie nun einmal anstelle Ihrer Workbench eine leere Diskette in DF0, und klicken das Diskicon der leeren Disk an. Wählen Sie nun INITIALIZE aus dem Menü DISK der Workbench. Sie werden feststellen, daß die Diskette ohne das lästige Einlegen der Workbench formatiert werden kann. Gleichermaßen verhält es sich mit dem Kopieren von Disketten. Sollten Sie nur über ein Laufwerk verfügen, klicken Sie das Icon der zu kopierenden Disk einmal an, und wählen DUPLICATE aus dem Workbenchmenü. Folgen Sie nun den Anweisungen des Computers. Verfügen Sie über zwei Laufwerke, ziehen Sie einfach das Originalicon über das Icon der anderen Diskette. Nachdem Sie den Requester mit CONTINUE beantwortet haben, startet der Kopiervorgang.

Eine fette Schreibmaschine

Die meisten Drucker sind in der Lage, Texte unterstrichen oder fett auszugeben. Hierzu müssen Sie nicht unbedingt eine Textverarbeitung oder das Notepad bemühen. Arbeiten Sie vom CLI:

In der Ausgabe 10/87 steht ein Tip, um den Amiga als Schreibmaschine zu verwenden. Mit einem einzigen Befehl im CLI werden alle Eingaben der Tastatur direkt an den Drucker umgeleitet:

```
COPY * TO prt:
```

Sobald Sie eine Zeile eingeben und <Return> drücken, schickt der Amiga den Text an den Drucker. Aber immer nur in der gleichen Schrift zu schreiben, ist auf Dauer langweilig. Geben Sie beispielsweise nach der obigen Zeile folgendes ein:

```
< ESC > [3m
```

Sie werden feststellen, daß alle Texte, die Sie von nun an eintippen, auf dem Drucker kursiv erscheinen. Dieser Zustand bleibt erhalten, bis Sie eine weitere Steuersequenz an den Drucker sen-

den. Selbst mitten in einer Textzeile können Sie die Schriftart ändern. Einige Codes stehen zur Auswahl:

Sequenz	Schriftart
< ESC > 1m	Bold
< ESC > 3m	Italic
< ESC > 4m	unterstreichen
< ESC > 4m	normal

Den »Schreibmaschinen-Modus« beenden Sie mit <CTRL \>. Selbstverständlich können Sie die Escape-Sequenzen auch im CLI verwenden, ohne zu drucken. (Norbert Cohen /ub)

Preferences auf Abwegen

Wußten Sie, daß man die Preferences auf einer Diskette einstellen kann, ohne daß sich das Programm auf der betroffenen Diskette befindet? Dies geht folgendermaßen vor sich: Wenn der Computer dazu auffordert, die Workbench-Diskette einzulegen, schieben Sie die Diskette ins Laufwerk, auf der die Änderungen vorgenommen werden sollen. Nach dem Booten stoppen Sie mit <CTRL D> die »Startup-Sequenz«, um im CLI zu bleiben. Falls Sie Besitzer eines Zweitlaufwerkes sind, gehen Sie wie folgt vor: Legen Sie die Workbench ins zweite Laufwerk. Kopieren Sie die Preferences in die RAM-Disk:

```
COPY DF1:Preferences RAM:
```

Falls Sie nur ein Laufwerk besitzen, gehen Sie anders vor. Legen Sie die Workbench ins Laufwerk und geben Sie ein:

```
df0:c/COPY Preferences TO RAM:
```

Danach legen Sie die zu bearbeitende Diskette wieder ein und starten das Programm Preferences:

```
RAM:Preferences
```

Stellen Sie die gewünschten Parameter ein und speichern Sie diese mit SAVE. Und schon werden die Parameter auf der Startdiskette gerettet. Der Vorteil liegt eindeutig darin, daß man mehr Speicherplatz auf der Diskette hat. J.R./ub

Alle Laufwerke im Griff (Requester)

Sicherlich kennen Sie alle die kleinen Dialog-Fenster mit Gerätebezeichnungen (DF1,DF2,DH0) in den Diskmenüs von DeluxePaint, Zing,

Haben Sie sich auch schon darüber geärgert, daß dort DH0 steht, obwohl man vielleicht über eine Filecard im PC-Slot (Sidecar) verfügt, also im Janussteckplatz. Warum kann dort also nicht JH0 stehen? Nun, damit ist jetzt Schluß:

Als erstes braucht man einen Filemonitor, zum Beispiel »Filezap« (Fish-Disk 14). Als Programm wählen wir DeluxePaint. Das Programm wird über den Filemonitor geladen und kann begutachtet werden (Sicherheitskopie verwenden). Die Änderung ist recht einfach. Man sucht in dem Programmcode die Bezeichnungen DH0 und ersetzt diese durch JH0, VD0 oder ähnliches — zurückschreiben nicht vergessen. Von nun an können Sie auch Ihre Festplatte, eine resetfeste RAM-Disk oder ein 5/4-Zoll-Laufwerk direkt anklicken. Genauso können Sie im übrigen PAL-Versionen erzeugen. Suchen Sie im Programmcode zum Beispiel von ED nach »RAW« oder »CON«. Hinter diesen Kürzeln stehen die Koordinaten des Fensters. Passen Sie diese Ihren eigenen Bedürfnissen an. Robert Albrecht/ub

Schnelle Hardcopies

Die Geschwindigkeit des Grafikausdrucks erhöht sich immens, wenn Sie, während der Drucker arbeitet, zur Workbench oder zum CLI zurückschalten.

Sie erreichen dies, indem Sie die linke <Amiga>-Taste und <N> gleichzeitig drücken. Ist die Hardcopy fertig, gelangen Sie mit der Kombination <linke Amiga-Taste M> zurück ins alte Programm. Auch Programme, die viel rechnen, lassen sich auf diese Art beschleunigen. (Werner Frankenberg/ub)

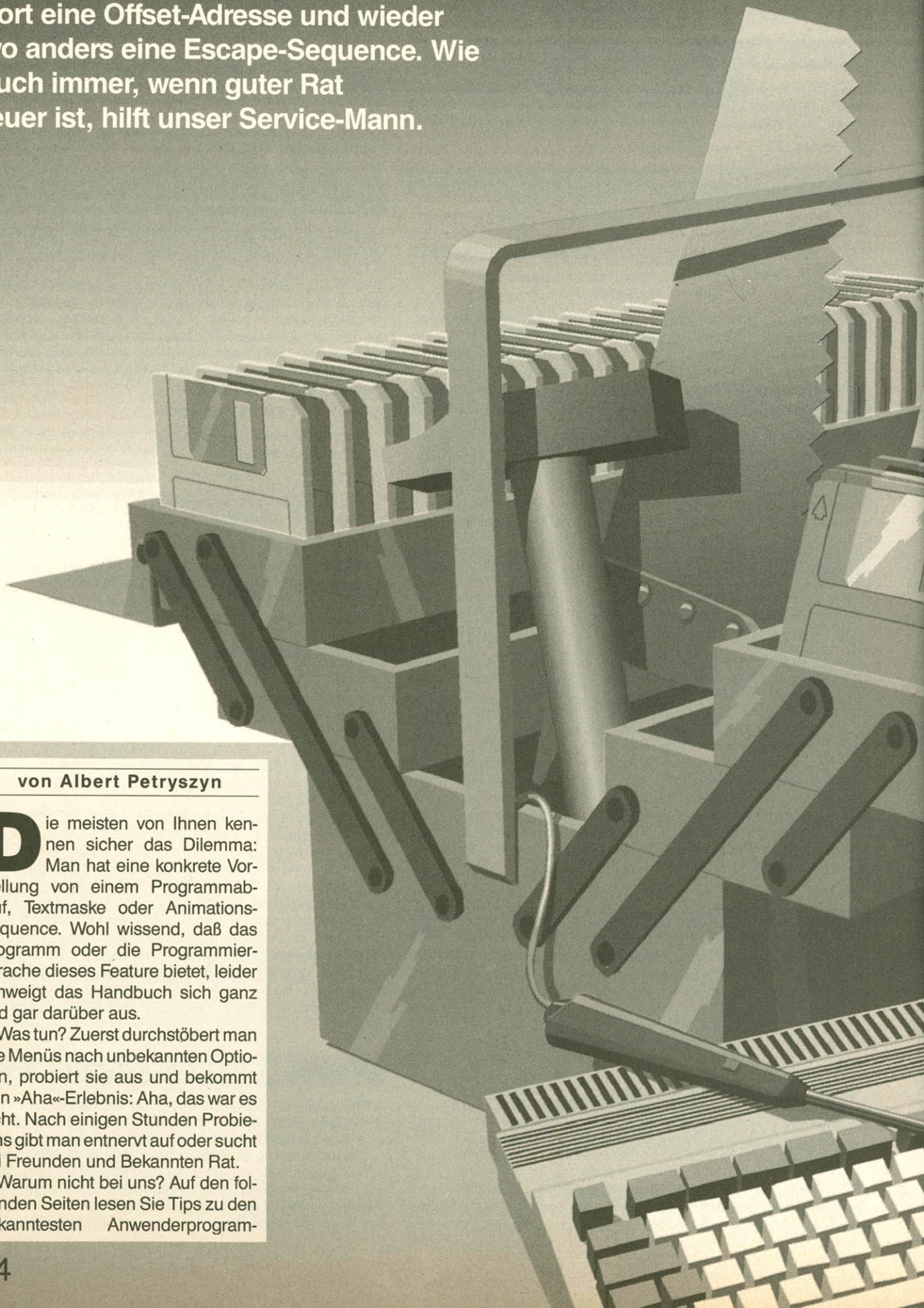
Oft sind es die kleinen Dinge im Leben, die einem das Computerdasein schwer machen. Hier eine Input-Routine dort eine Offset-Adresse und wieder wo anders eine Escape-Sequence. Wie auch immer, wenn guter Rat teuer ist, hilft unser Service-Mann.

von Albert Petryszyn

Die meisten von Ihnen kennen sicher das Dilemma: Man hat eine konkrete Vorstellung von einem Programmablauf, Textmaske oder Animations-Sequence. Wohl wissend, daß das Programm oder die Programmiersprache dieses Feature bietet, leider schweigt das Handbuch sich ganz und gar darüber aus.

Was tun? Zuerst durchstöbert man alle Menüs nach unbekanntem Optionen, probiert sie aus und bekommt sein »Aha«-Erlebnis: Aha, das war es nicht. Nach einigen Stunden Probierens gibt man entnervt auf oder sucht bei Freunden und Bekannten Rat.

Warum nicht bei uns? Auf den folgenden Seiten lesen Sie Tips zu den bekanntesten Anwenderprogram-



Der Service-Techniker



men. Wie z.B. Beckertext, Documentum und Wordperfect, wo wir zeigen, wie Sie Ihre Druckerausgabe perfekt gestalten. Ob Etikettendruck, Serienbriefe oder Sonderzeichen, mit unserem Service kein Problem.

Oder wollen Sie Superbase-Datei optimieren? Unser Techniker sagt Ihnen, auf was Sie achten müssen.

Auch die Grafik-Freaks kommen nicht zu kurz. Wir zeigen wie Sie mit Deluxe Paint III und Deluxe Video III eine komplette Präsentation erstellen. Oder berechnen Sie realistische Computerbilder mit Turbo Silver 3.0 und Sculpt. Unser Service-Mann zeigt Ihnen wie Sie Texturen, Oberflächeneigenschaften und Lichter effektiv einsetzen.

Die Programmierer aller Sparten (Assembler, C, Modula und Amiga-Basic) finden in diesem Heft einen wahren Ideen-Pool, aus dem sie schöpfen können. Amiga-DOS-User werden in die geheimen Pfade des Betriebssystems eingeweiht und der kleine Komponist bekommt den Notenschlüssel zur Schatztruhe von Sonix.

Um Ihnen die Suche zu erleichtern finden Sie im untenstehenden Infokasten, wo der Service-Techniker für Ihr Programm im Startloch wartet. *pe*

Tips Übersicht

Basic	xxx	WordPerfect	xxx
C	xxx	Sonix	xxx
Assembler	xxx	Superbase	xxx
Modula	xxx	Turbo Silver	xxx
AmigaDos	xxx	Sculpt	xxx
Beckertext	xxx	DPaint und	xxx
Documentum	xxx	DVideo	

Mit List

So mancher professionelle Schreiber mag Beckertext und seine Junior-Version Textomat belächeln. Man verweist stolz auf den Leistungsumfang der MS-DOS-Programme und vergißt dabei, daß PC-Software wesentlich teurer ist.

Wer einen Volkswagen kauft, erwartet nicht die Leistung eines Porsche. Das Handbuch hilft beim »Einfahren« der Software. Mit unseren Tips & Tricks nutzen Sie die Leistungsreserven.

Drei Varianten bieten Beckertext und Textomat für die Wahl des Zeilenabstandes: Sie können den Text ein-, eineinhalb und zweizeilig formatieren. Nur drei? Tatsächlich verfügen Sie über eine weitere Einstellung, um die Zeilendichte zu beeinflussen: der Parameter »Zeilenabstand« im Dialogfenster »Formular«.

Eigentlich dient diese Einstellung dazu, Programme an Druckertypen anzupassen: Für Geräte, die das Blatt in Schritten von 1/72 Zoll transportieren (z.B. IBM-Printer-Modus), sollte der Wert 12 eingetragen sein. Drucker, die nur einen Zeilenvorschub von 1/60 Zoll erlauben (z.B. Epson-kompatible), benötigen hingegen den Wert 10. So ist gewährleistet, daß die Programme die Seitenlänge richtig berechnen.

Zeilenabstand

Sie sollten den Wert so einstellen, wie es in der Programm-Dokumentation empfohlen wird. Wenn es darauf ankommt, möglichst viel auf einer Seite unterzubringen, ist der Wert so weit zu verringern, wie es die verwendete Schriftgröße des Druckers erlaubt. Ein zu kleiner Wert führt dazu, daß sich Zeilen überlappen. Beckertext und Textomat akzeptieren für die Einstellung des Zeilenabstands nur gerade Werte.

Ein Anwendungsbeispiel wäre ein Bus- oder Zugfahrplan im Mini-Format: Wenn Sie den Zeilenabstand so gering wie möglich einstellen und zusätzlich die Schmalschrift

Ihres Druckers verwenden, können Sie sich eine Tabelle zusammenstellen, die die wichtigsten Zugverbindungen enthält und nur wenig Platz in der Brieftasche beansprucht.

Auch Programme wie Beckertext und Textomat wollen bisweilen nicht so, wie sie sollen – und nicht immer ist der Anwender schuld. Hier sind zwei Programmfehler und Verfahren, um sie zu umgehen:

■ Wenn Sie mit Beckertext eine Zeile zentrieren und nach Abschluß der Zeile mit <Return> ein zweites Mal <Return> drücken (um eine Leerzeile einzufügen), wird der zentrierte Text erneut verschoben. Um diesen Fehler zu beheben, gehen Sie in die Zeile mit dem zentrierten Text zurück und drücken <Help>. Wenn Sie danach die Zeile mit den Cursortasten verlassen (also nicht wieder <Return> drücken), bleibt die richtige Formatierung erhalten.

■ Bis zu 999 Zeichen enthalten Zeilen bei Beckertext und Textomat. So viel wird kaum jemand benötigen. Keine Seltenheit ist eine Textbreite von 140 Zeichen (in Kombination mit der Schmalschrift des Druckers) bei umfangreichen Tabellen. Bei der Darstellung eines solchen Dokuments bewegt das Programm den Text auch horizontal über den Bildschirm. Ein Programmierfehler sorgt dafür, daß dabei mitunter eine Zeile verlorengelht, wenn man vom rechten an den linken Rand zurückspringt. Der nur scheinbar verschwundene Text läßt sich durch Antippen von <Help> wieder auf den Bildschirm zurückholen.

Mit Beckertext lassen sich mehrere Texte bearbeiten. Wenn Sie einen der Texte speichern, vergewissern Sie sich, daß der richtige Name im Datei-Eingabefeld steht. Das Programm trägt dort häufig den Namen des zuletzt geladenen (oder gespeicherten) Textes ein, statt den Namen des Textfensters zu übernehmen.

Sie brauchen nicht in Panik geraten, wenn Sie versehentlich einen

Alle Druckerschriften und Schriftattribute nutzen, mathematische Sonderzeichen einsetzen, kleine Tücken überwinden – wir zeigen Ihnen, wie Sie mehr aus Beckertext und Textomat herausholen.

von Karsten Lemm

Text überschrieben haben: Bevor Beckertext und Textomat eine Datei überschreiben, legen sie von der vorigen Version eine Sicherheitskopie mit dem Suffix ».BAK« im gleichen Verzeichnis an. Speichern Sie den Text zunächst unter dem richtigen Namen, laden Sie dann die Kopie des überschriebenen Textes und sichern Sie ihn wieder.

Ein weiteres Problem bei Beckertext: Wenn Sie in der Eingabezeile »Ordner« Laufwerk oder Verzeichnis wechseln und <Return> drücken, lädt das Programm normalerweise die Inhaltsübersicht. Befindet sich in der Dateizeile noch der Name eines Textes, versucht das Programm, diesen sofort zu laden. Das kann beabsichtigt sein. Falls Sie aber vergessen haben, die Dateizeile zu löschen, erhalten Sie häufig die Meldung: »Eine Datei mit diesem Namen ist auf der angegebenen Diskette nicht vorhanden.«

Wenn Sie danach das Dateifenster erneut aufrufen, passiert nichts – nicht einmal, wenn Sie in die Ordnerzeile gehen und <Return> drücken. Sie können Beckertext dennoch dazu bringen, den Inhalt des gewünschten Verzeichnisses anzuzeigen: Nehmen Sie die entsprechende Diskette aus dem Laufwerk und legen Sie die Diskette wieder ein. Festplatten-Besitzer müssen

& Tücke

Ordner oder Gerät wechseln, um Beckertext aus dem »Dornröschenschlaf« zu wecken.

Solche Schwierigkeiten lassen sich vermeiden, wenn Sie die Datei-zeile durch Drücken der <Esc>-Taste löschen, und die Ordnerzeile nach Änderungen mit <Cursor unten> statt <Return> verlassen.

Die Trennautomatik von Textomat und Beckertext ist recht zuverlässig, hat allerdings einen kleinen Fehler, der bei schmalen Spalten von Bedeutung sein kann: Die Automatik trennt nicht, wenn die erste Hälfte des zu trennenden Wortes gerade noch in die Zeile paßt, das heißt, wenn die Zeile dadurch vollständig »gefüllt« wird und der Trennstrich mit dem rechten Rand abschließt.

Trenn-Hilfe

Sie können das Programm dennoch veranlassen, das Wort zu trennen: Plazieren Sie den Cursor auf dem ersten Buchstaben des zu trennenden Wortes. Drücken Sie die Korrekturtaste (<Backspace>). Der Cursor wird in die vorige Zeile springen und das Leerzeichen hinter dem letzten Wort löschen. Drücken Sie nun <Help>, um den Absatz neu zu formatieren und danach die Leertaste (<Space>), um das Leerzeichen wieder einzufügen. Beachten Sie, daß die von Ihnen erzwungene Trennung wieder verlorengeht, wenn Sie den Absatz noch einmal mit <Help> formatieren.

Eine zweite Möglichkeit zur Trennung wäre, das Wort wie auf einer Schreibmaschine mit einem »harten« Trennstrich (Minustaste) zu teilen und dann auf <Help> zu drücken. Im Gegensatz zu den »weichen« Trennstrichen, die die Automatik setzt und die Sie mit <Ctrl> einfügen können, bleibt eine harte Trennung bestehen, wenn das Wort seinen Platz am Ende der Zeile verläßt. Sie sollten solche Trennstriche nur dann einsetzen, wenn Sie den Absatz nicht mehr verändern.

Beckertext besitzt eine Rechtschreibkontrolle, die Sie auch während des Schreibens einschalten können. Das Lexikon wird dann vollständig in den Arbeitsspeicher geladen, weil Diskettenzugriffe den Arbeitsablauf verlangsamen. Das Problem dabei: Je größer der Wortschatz des Lexikons wird, um so weniger Speicher bleibt dem Text. Die Programmierer sind deshalb auf die Idee gekommen, die Kontrollfunktion nicht nur mit ganzen Wörtern, sondern auch mit Silben arbeiten zu lassen. Beckertext erkennt das Wort »untergehen«, auch wenn im Lexikon nur die Bestandteile »unter« und »gehen« gespeichert sind.

Das klingt clever, hat aber einen Haken: Wenn Sie den Wortschatz nicht mit Bedacht erweitern, läßt die Funktion bald die seltsamsten Konstruktionen als korrekt gelten. Je

Übrigens: Sobald Sie das Lexikon geladen haben, richtet sich auch die Trennautomatik nach dessen Wortschatz, so daß es zu kuriosen Fehltrennungen kommt, wenn das Lexikon zu viele Silben-Kombinationen erlaubt.

Wollen Sie Ihr Lexikon »aufräumen«? Laden Sie es wie einen Text und Sie können es bearbeiten. Vielleicht stellen Sie fest, daß es nicht vollständig im Arbeitsspeicher Platz hat, wenn Ihr Amiga »nur« über 1 MByte RAM verfügt. Glücklicherweise wird der Wortschatz des Lexikons im ASCII-Format gespeichert, so daß Sie nicht unbedingt Beckertext benutzen müssen, um ihn zu bearbeiten.

Verwenden Sie statt dessen ein Programm, das weniger Speicherplatz benötigt, z.B. den Text-Editor »MEMACS«, der sich auf der Extras-Systemdiskette befindet. Wenn Sie das Lexikon nicht außergewöhnlich erweitert haben, sollte es sich komplett laden und bearbeiten lassen. Speichern Sie den Text nach der Änderung wieder als ASCII-Datei (nur mit einer Kopie der Originaldiskette durchführen).

Für den Druck mit Einzelblättern ist zweierlei zu beachten: Zunächst sollten Sie mit dem Programm Preferences die Einstellung »Single Sheet« aktivieren, da Beckertext und Textomat diesen Parameter übernehmen.

Darüber hinaus werden Sie vermutlich etwas mit den Werten im Formular-Menü experimentieren. Viele Drucker, die einen halbautomatischen

Einzelblatt-Einzug besitzen, können einen Bereich von etwa 2,5 cm am oberen und unteren Blattrand nicht bedrucken. Sie sollten daher die Werte »Papierlänge« und »bedruckbare Höhe« entsprechend ändern. Bei den genannten 2,5 cm Freiraum ist für die erste Einstellung ein Wert um 60 einzusetzen, für die zweite Einstellung etwas weniger.

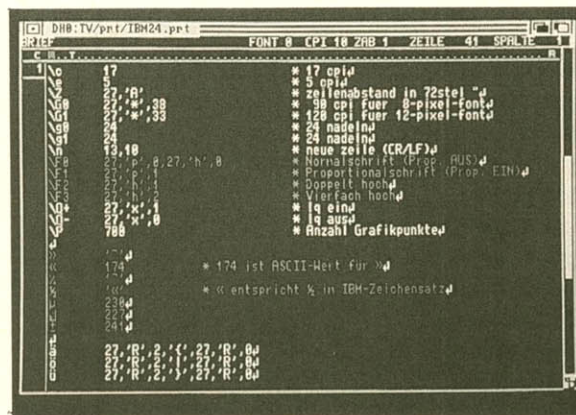


Bild 1. Druckertreiber: Die Treiber können wie Texte geladen und bearbeitet werden

mehr Silben der Wortschatz enthält, desto mehr (unsinnige) Kombinationen gibt es. Sie sollten also, wenn Sie ein neues Wort in das Lexikon aufnehmen, möglichst keine Wortteile durch das Pluszeichen als eigenständige Silben definieren.

Cee-Pee-liih

Ein großer Pluspunkt von Textomat und Beckertext ist die Flexibilität beim Einsatz verschiedener Schrift-dichten. Die Programme nehmen Ihnen (Rechen-)Arbeit ab, wenn Sie bei der Formatierung einige Grundsätze beachten:

Nehmen wir an, Sie schreiben einen Text mit 60 Zeichen pro Zeile und einer Schrift-dichte von 10 cpi (cpi steht für »Characters per Inch«, Zeichen pro Zoll). Wenn Sie einen Absatz in einer etwas schmaleren Schrift von 12 cpi schreiben wollen, wird das Programm automatisch berücksichtigen, wie viele Zeichen zusätzlich in die Zeile passen. Es bleibt also bei der ursprünglichen Textbreite.

Wenn Sie die Zeilenbreite ebenfalls ändern, bezieht sich der Wert nicht länger auf 10, sondern auf die nun eingeschalteten 12 cpi. Haben Sie die Textbreite so auf 50 Zeichen mit 12 cpi verringert, und schalten darauf wieder auf 10 cpi, werden nur noch etwa 40 Zeichen in eine Zeile passen. Um zur ursprünglichen Textbreite von 60 Zeichen zurückzukehren, ist die Zeilenbreite erneut zu ändern.

Neue Schriften

Haben Sie sich den Beckertext-Treiber für Ihren Drucker schon mal angesehen? Das kann sich lohnen, denn wenn Sie selbst »Hand anlegen«, läßt sich mehr aus dem Drucker herausholen, als Textomat und Beckertext vorsehen:

Laden Sie den Treiber wie einen Text. Einzelheiten dazu finden Sie im Handbuch. Betrachten Sie die Zeilen, in denen die Schriftattribute definiert werden. Manche Drucker bieten außer Kursiv- oder Fettschrift Attribute wie »Schatten« oder »Outline«. Um sie zu nutzen, sind andere Attribute zu »opfern«. Beachten Sie, daß die Attribute »doppeldruck« und »fett« zusammen die Schriftart »fett« aus dem Menü »Schrift« ergeben.

Beispiel: Im Standard-Druckertreiber ist für die Schriftart »unterstreichen ein« die Steuersequenz

3+ 27, '-', 1

eingetragen. Wenn Sie dieses Attribut durch die Schattenschrift des Druckers ersetzen wollen, tragen Sie in diese Zeile die Steuersequenz ein, die den Drucker anweist, die Schattenschrift einzuschalten. Löschen oder überschreiben Sie nicht die interne Steuersequenz der Textverarbeitung (»3+«). In den meisten Fällen wird auch die »27« stehenbleiben, da sie den ASCII-Wert des »Escape«-Codes darstellt, mit dem viele Drucker ihre Steuersequenzen einleiten. Nähere Informationen finden Sie in Ihrem Druckerhandbuch und in den Anleitungen zu Beckertext und Textomat.

Proportionalschrift

Für Anwender von Beckertext gibt es einen anderen Weg, um zusätzliche Schriftarten des Druckers zu nutzen: Wie Sie wissen, bietet Ihnen das Programm die Möglichkeit, über das Menü »Druckerfont« bis zu zehn Schriften einzuschalten. Da (derzeit) nur die wenigsten Geräte so viele Zeichensätze bieten, können Sie statt eines Fonts auch ein Schriftattribut aktivieren. Beispiel: Ihr Drucker besitzt fünf eingebaute Schriften, die Sie als Druckerfonts 0 bis 4 im Druckertreiber definiert haben. Sie könnten Font 5 mit der Outline-Schriftart belegen.

Der Vorteil dieser Methode: Sie verlieren keines der vordefinierten Attribute. Der Nachteil: Für Beckertext sind derart definierte Schriftarten keine Schriftarten, die Sie beliebig ein- und ausschalten können, sondern Druckerfonts, die sich nur zeilenweise wechseln lassen. Außerdem müssen Sie die neue Schriftart bei der Definition mit einer der Druckerfonten fest verknüpfen. Wenn die »Courier«-Schrift schattiert gedruckt werden soll und ein anderes Mal die »Prestige«, sind zwei Fontdefinitionen anzulegen. Nur wenn Sie, wie oben beschrieben, etwa »unterstreichen« durch »schattiert« ersetzen, können verschiedene Schriften mit der Zusatz-Schriftart kombiniert werden.

Die meisten Drucker besitzen Proportionalschrift. Von Beckertext und

Textomat kann die Schrift über einen Umweg angesprochen werden:

Jede Zeile, die eine Steuersequenz für das Aktivieren einer Schrift enthalten soll, beginnt mit einem der Kommandos »F0« bis »F9«. Rechts daneben befindet sich die jeweilige Steuersequenz. Die folgenden Beispielzeilen können Sie ohne Änderungen übernehmen, wenn Sie einen 24-Nadel-Drucker besitzen, der in der Epson-LQ-Betriebsart arbeitet (Treiber: lq800.prt):

```
F0 27, 'p', 0, 27, 'h', 0 *
Normalschrift (Prop. AUS)
F1 27, 'p', 1 *
Proportionalschrift
(Prop. EIN)
F2 27, 'h', 1 *
Doppelt hoch
F3 27, 'h', 2 *
Vierfach hoch
```

Wir haben die Fonts 2 und 3 mit weiteren Schriftattributen belegt. Beckertext weiß allerdings nicht, wieviel Platz Zeichen mit diesen Attributen beanspruchen. Das heißt, Sie müssen im »Formular«-Requester die Seitenlänge entsprechend verringern, wenn Sie eine der Schriftarten »Doppelt hoch« oder »Vierfach hoch« verwenden.

Ein Hinweis für Textomat-Anwender: Auch Sie können die Proportionalschrift Ihres Druckers nutzen, wenn Sie auf eine der Zeichendichten aus dem »Schrift«-Menü verzichten. Falls Sie selten die Breitschrift verwenden, können Sie mit der Einstellung »5 cpi« die Proportionalschrift ansprechen. Es genügt jedoch nicht, in der Zeile

```
E 27, 'W' * Breitdruck
```

die Steuersequenz für das Einschalten der Proportionalschrift einzutragen. Vielmehr müssen Sie zusätzlich in der Zeile

```
Öe 5 * 5 CPI
```

den Wert für die Zeichenbreite ändern. Diese Einstellung ist für das Programm die Grundlage zur Berechnung der Textbreite nach einem Wechsel der Zeichendichte.

Welcher Wert ist der richtige für Proportionalschrift? Im Prinzip keiner und jeder, denn Merkmal der Proportionalschrift ist die variable

Zeichenbreite. Jeder Buchstabe nimmt unterschiedlich viel Raum ein, z.B. ein »W« mehr als ein »I«. Da Textomat und Beckertext auf diese Besonderheit nicht vorbereitet sind, müssen Sie einen Kompromiß eingehen. Ein realistischer Wert läge zwischen 12 und 15, wenn Sie mit der Pica-Proportionalschrift Ihres Druckers arbeiten. Pica entspricht ohne Proportionalschrift 10 cpi, und in der Regel passen bei Proportionalschrift einige Zeichen mehr in eine Zeile.

Noch eine Änderung am Treiber kann nötig sein, damit dieser Trick funktioniert: Um von der Proportionalschrift wieder zu einem festen Zeichenabstand wechseln zu können, sollten Sie in den Zeilen, die mit »A« bis »C« eingeleitet werden, die Steuersequenz für das Ausschalten der Proportionalschrift einfügen. Ob dies bei Ihrem Drucker erforderlich ist, sollten Sie durch Experimentieren herausfinden.

Sie werden sehen, daß sich die Arbeit lohnt, denn die Schönschrift Ihres Druckers wird in Kombination mit der Proportionalschrift eleganter aussehen. Allerdings ist es wegen der variablen Zeichenbreite nicht möglich, Blocksatz, Zentrierung oder rechtsbündiges Textformat zu nutzen.

Viele Drucker besitzen mehrere Betriebsarten (»Emulationen«), in denen verschiedene Zeichensätze verfügbar sind. Die Zeichensätze enthalten neben den üblichen weiteren Sonderzeichen, die nicht ohne weiteres genutzt werden können, da die Zeichensätze des Amiga in der Regel nicht mit denen des Druckers übereinstimmen. So hat das »Pi« (π) im Amiga-Zeichensatz den ASCII-Wert 182. Der Drucker erwartet - in der IBM-Grafik-Betriebsart - für dieses Zeichen den Wert 227. Mit Beckertext und Textomat ist es möglich, dieses Problem zu umgehen, so daß sich auch komplizierte mathematische Formeln ausdrucken lassen. Wir werden am Beispiel des IBM-Grafikzeichensatzes beschreiben, was im einzelnen zu tun ist.

Zunächst benötigen Sie einen Font-Editor, z.B. den »Fed« von der »Extras«-Diskette. Damit lassen sich Zeichen entwerfen, die zwar im Druckerzeichensatz vorhanden sind, nicht aber in den von Beckertext ver-

wendeten Bildschirm-Fonts. Sie finden diese Zeichensätze im »Fonts«-Verzeichnis der Programmdiskette. Sie beginnen mit »DBFont...«. Die Schriften »DBFont_super« und »DBFont_sub« brauchen Sie nur dann zu ändern, wenn Sie die Sonderzeichen auch in Verbindung mit den Modi »hochstellen« und »tiefstellen« verwenden wollen.

Es bietet sich an, in erster Linie Zeichen zu ändern, die nicht benötigt werden und günstig auf der Tastatur liegen. Das mit <ALT a> aufgerufene »æ« könnte durch das griechische Alpha ersetzt werden.

Vergessen Sie nicht, den geänderten Zeichensatz unter demselben Namen wieder zu speichern, damit

Sonderzeichen schwarz auf weiß

er von Beckertext oder Textomat erkannt wird. (Verwenden Sie auf jeden Fall eine Kopie der Original-Programmdiskette.)

Danach ist es nötig, den Druckertreiber so zu ändern, daß die auf dem Bildschirm dargestellten Zeichen tatsächlich gedruckt werden. Laden Sie den von Ihnen verwendeten Treiber wie einen Text. Am Ende der Datei befinden sich bereits einige Steuersequenzen für internationale Sonderzeichen.

Sie sollten für jedes Zeichen, das Sie ersetzen wollen, die entsprechende Zeile ändern. Es ist wichtig, daß Sie nicht für ein solches Zeichen eine neue Zeile am Ende des Treibers hinzufügen. Das sollten Sie nur für Zeichen tun, die im Treiber noch nicht definiert sind.

Wie die Änderungen im einzelnen aussehen, hängt von Ihrem Drucker ab. Bei manchen Geräten lassen sich verschiedene Betriebsarten nur per DIP-Schalter einstellen. Sollte das bei Ihrem Drucker der Fall sein, sieht die Zeile für das Pi so aus:

227

Es genügt, wenn Sie links das Zeichen und rechts daneben (mit

einigen Leerzeichen Abstand) den ASCII-Code des Drucker-Zeichensatzes schreiben. Eine Übersicht mit allen Zeichen und den dazugehörigen Werten finden Sie im Handbuch Ihres Druckers.

Wichtig ist in diesem Fall, daß sich mit der Betriebsart des Druckers nicht nur der Zeichensatz ändert, sondern meistens auch der Befehlsatz. Das heißt: Vergewissern Sie sich, daß die im Treiber aufgeführten Steuersequenzen etwa für die Schriftarten auch für die neue Betriebsart (in unserem Beispiel: IBM-Grafik) gelten. Andernfalls müssen Sie die Sequenzen entsprechend ändern.

Achten Sie besonders auf den Zeilenabstand: Drucker, die in der Epson-Betriebsart einen Zeilenvorschub von 1/60 Zoll ausführen, schieben das Papier im IBM-Modus nur um 1/72 Zoll vor. Im »Formular«-Menü muß dann der Wert für den »Zeilenabstand« von 10 auf 12 erhöht werden, damit die Seite vom Programm korrekt formatiert wird.

Bei manchen Druckern treten solche Schwierigkeiten nicht auf, weil es möglich ist, die Betriebsarten oder Zeichensätze »softwaremäßig« zu wechseln, das heißt mittels eines Steuercodes. Ein solcher Drucker ist z.B. der Citizen 120D, für den unsere Beispielzeile so aussieht:

27, '51', 227, 27, '50'

Wieder steht links das zu druckende Zeichen. Dann folgt zunächst der Code zum Einschalten des IBM-Zeichensatzes (»27,51'«), danach der ASCII-Wert für das Pi, und schließlich wird mit »27,50'« wieder zum ursprünglichen Zeichensatz zurückgeschaltet.

Wenn Sie alle Änderungen vorgenommen haben, speichern Sie den Druckertreiber als ASCII-Datei und unter einem besonderen Namen (z.B. Sonderdruck) im Verzeichnis »prt« der Programmdiskette. Beachten Sie, daß Sie den Treiber noch einmal laden müssen, damit die Änderungen wirksam werden. Bei Beckertext geht das aus dem Programm heraus. Textomat ist dafür neu zu starten.

pa/pe

Dieser Tip stammt von Ingo Stampe aus Droste Hülshoff. Wir haben ihn bearbeitet und erweitert.

Um sinnvoll mit »Documentum« zu arbeiten, benötigen Sie mindestens 1 MByte Speicher. Natürlich funktioniert das Programm auch mit weniger Speicherplatz. Gehen wir zur Sache - von Anfang an. Alle folgenden Maßnahmen führen Sie bitte nicht mit Ihrer Originaldiskette, sondern mit einer Sicherheitskopie durch.

Voreinstellungen:

Nach dem Öffnen des »Documentum«-Disketten-Icon klicken Sie den großen Schriftzug »Documentum« mit der linken Maustaste einmal an und wählen im Workbench-Menü »INFO«. In der nun erscheinenden Infodatei nehmen Sie verschiedene Einstellungen vor. Sie können beispielsweise den Speicher für Ihre Texte einstellen, der durchschnittlich gebraucht wird. Die Voreinstellung liegt bei 50 KByte. Geben Sie den benötigten Speicherwert ein und »Documentum« wird Ihnen bei jedem Laden den gewünschten dort eingegebenen Speicher reservieren.

Bei der Wahl der Speichergröße sollten Sie auf keinen Fall den vollen angegebenen Wert ausnützen, weil dies zum Systemabsturz führen könnte, wenn das Programm zu wenig Arbeitsspeicher für interne Berechnungen oder Pufferspeicher braucht. In den seltensten Fällen benötigt man 350 KByte Textspeicher.

Das Einstellen der Bildschirmfarben, mit denen Sie arbeiten wollen, ist kein Problem. Zur besseren Darstellung haben sich die Farben Weiß auf Schwarz bewährt. Wählen Sie dazu in dem Tooltypes-Fenster mit der Maus den Pfeil nach unten und geben der Reihe nach ein:

```
Farbe0=000 | Farbe2=0ae
Farbe1=fff | Farbe3=000
```

Geben Sie bei »Tabulatortaste« »LEERZEICHEN«, bei »ICONS« »aus« ein. Haben Sie auf Ihrer Speichererweiterung keine akkugepufferte Uhr, können Sie den nächsten Eintrag getrost vernachlässigen. Im anderen Fall tippen Sie bei »Zeit« entweder »Aktuell« oder »Variabel«. Steht hier aktuell, wird im Text später

Documentum V1.0

Das professionelle Textsystem

für ein bestimmtes Zeichen im Text die aktuelle Uhrzeit gesetzt. Jetzt können Sie noch bestimmen, ob die Zeichensätze (Fonts) sofort oder ob Sie später vom Programm geladen werden sollen. Die weiteren Einstellungen sind druckerspezifisch. Der Zeilenvorschub und der Grafikmodus werden über Escape-Sequenzen (27,...) gesteuert. Die Werte sind bereits für Epson-Drucker eingestellt. Besitzen Sie einen anderen Drucker, müssen Sie für die einzelnen Sequenzen Ihr Handbuch befragen. Vergessen Sie nicht, die Einstellungen mit »Save« zu speichern.

Um »Documentum« sofort nach dem Booten betriebsbereit zu haben, könnte die Startup-Sequence folgendermaßen aussehen:

```
Addbuffers df0: 10
echo "nxe[33m>>Documentum<<
    xe[31mTextverarbeitung
1 MB"
echo "(C) 1989 Markt &
Technik Verlag München"
c:SetPatch >NIL: ; Systemrou-
tinen korrigieren
FF >NIL: -0 ; Textausgabe be-
schleunigen
resident CLI L:Shell-
Seg SYSTEM pure add; Shell ak-
tivieren
mount newcon:
SetMap d ; Deutsche Tastatur-
belegung aktivieren
path ram: c: s: sys:prefs
add ; Pfad für Workbench fest-
legen
LoadWB delay ; Vor Weiter-
machen warten
documentum:documentum
endcli >NIL:
```

Mit »Documentum« kann ein Text anschaulich gestaltet werden, und es bleibt erspart, vorher seitenweise ein Handbuch zu wälzen. Doch die Einführung läßt leider wesentliche und interessante Tricks aus.

Jetzt wird »Documentum« beim Laden gestartet und Sie brauchen keine Zeit mit dem Auswählen auf der Workbench zu verschwenden.

Wenn Sie ein Dokument speichern, sollten Sie es durch ein »Suffix« klassifizieren.

Hängen Sie z.B. an Briefe das Suffix ».bf« an. Ein Brief an Max wird unter dem Filenamen »Max.bf« gespeichert. Andere Texte können Sie z.B. mit dem Suffix ».tx« kennzeichnen. Damit finden Sie Ihre Texte schneller. Schreiben Sie beim Laden in den Suffixkasten ».bf« oder ».tx«. Der Amiga zeigt dadurch nur diejenigen Texte im Directory, die der Einordnung entsprechen. Um einen Text möglichst schnell zu laden, ohne Eingaben über die Tastatur, klicken Sie das File oder Directory zweimal nacheinander an, der Text wird sofort ohne Return-Bestätigung geladen.

Ausdruck:

Wenn Sie einen grafikfähigen Matrixdrucker besitzen, können Sie andere Schriftarten verwenden. Diese

onelle stem

werden im Grafikmodus ausgedruckt. Vergessen Sie nicht, im Requester »Fonts suchen«, den genauen Pfad einzugeben. Im Menü »Fonts« können Sie die Zeichensätze anwählen, die Sie im Menü »Projekt/Fonts laden« geladen haben. Achten Sie darauf, daß »Documentum« Zeichensätze kleiner druckt, als angegeben. Sie sollten nur Schriftarten mit angegebener Höhe größer als 25 nehmen. Das heißt hinter den Zeichensätzen im Menü »Fonts« erscheinen zum Teil Größenangaben. Dieser Wert sollte nicht kleiner als 25 sein, ansonsten ist die Schrift kaum noch lesbar. Möchten Sie vom Drucker die Schriftart einstellen, können Sie im Menü »Fonts« »Druckerfont« anklicken.

Will man ein gesondertes Seitenformat laden oder speichern, sollte man darauf achten, daß im »Seitenformatladen/speichern«-Requester das Suffix ».frm« eingestellt ist. Das Seitenformat wird jedesmal auf die Einheit Pixel voreingestellt, vergessen Sie bei Ihrer Planung nicht, sie auf Millimeter zu stellen, da die Rechnung damit leichter fällt.

Möchten Sie Ihren Text vor dem Ausdrucken noch einmal auf dem Bildschirm im WYSIWIG-Verfahren (What You See Is What You Get) sehen, so wählen Sie »Zeige Druck«.

Haben Sie mit Drucker-Escape-Sequenzen gearbeitet, kann Ihr Text völlig anders aussehen, als später auf dem Papier.

Blockbearbeitung:

In »Documentum« und anderen Textverarbeitungen versteht man darunter das Auswählen und Bear-

beiten von Textteilen. Wenn Sie das Menü »Edit/Block markieren« angewählt haben, setzen Sie gleichzeitig die Startmarke der gewünschten Blockoperation. Sie haben die Möglichkeit, über die linke Maustaste und Bewegungen der Maus, Ihren Block anzuwählen. Für die Markierung eines Blocks ist es wichtig, die Markierung mit der Maus bis zum Ende der gewünschten Stelle zu fahren. Beziehen Sie den Return-Pfeil in die Markierung mit ein.

Es kann passieren, daß »Documentum« eine Zeile verschluckt, wenn Sie keinen Return-Pfeil am Ende einer Zeile gesetzt haben. Sie sollten daher immer am Ende einer Zeile ein Return drücken.

Wählen Sie im Menü »Edit« die Funktion »Block kopieren« an und die Markierung verschwindet. Nach Anwahl des Punktes »Neuer Text« geben Sie bei Speichergröße den Wert 50 ein. Jetzt können Sie im Edit-Menü »Block kopieren« anwählen und haben einen Block in ein anderes Dokument kopiert. Eine weitere Möglichkeit der Blockoperation: Wählen Sie nach der Blockanwahl »nächster Text« und setzen den Cursor irgendwo in Ihren zweiten Text, der sich im Speicher befindet. Dort gehen Sie auf den Punkt »Block kopieren«. Der Text wird genau an der Stelle eingefügt, an welcher sich der Cursor befindet. Auf die gleiche Weise können Sie Textteile löschen, ohne ständig die Del- bzw. Back-Taste zu benutzen.

Tastenkommmandos:

»Documentum« kennt viele Tastaturkommmandos. Sie können beispielsweise mit der ALT-Taste und

<CURSOR aufwärts> zum Textanfang bzw. mit <ALT CURSOR abwärts> zum Textende springen. Mit <ALT CURSOR rechts> bzw. <ALT CURSOR links> gelangen Sie zum Zeilenanfang bzw. -ende.

Die Kombination <SHIFT Cursor-tasten> bedeutet ein Wort vor bzw. zurück oder eine Seite hoch bzw. runter.

Floskelfunktion:

Manch einer fragt sich, wozu die Floskelfunktion existiert. Man kann diese Funktion einsetzen, wenn man oft lange oder komplizierte Ausdrücke im Text verwendet. Man wählt die mit dem Ausdruck zu belegende Funktionstaste oder -kombination und trägt den Ausdruck ein. So kann man Tabellen gut gestalten. Die Tabulatortaste ist leider nur auf einen Wert einstellbar. Gibt man bei der Floskeleingabe die Anzahl von Leerzeichen ein, die zwischen gleichlang bleibenden Wörtern stehen, kann individuell der Tab-Sprung ausgewählt werden.

Eine gute Einrichtung ist die Querverweisliste. Hier können Sie abfragen, wo im Text welche Wörter stehen. Das heißt bei Anwahl dieses Menüs wird »Documentum« einen neuen Textbildschirm aufbauen und alle Zeilen bzw. Spaltenpositionen für das gesuchte Wort angeben. Um die Anzahl des Wortes bei häufiger Verwendung festzustellen, positionieren Sie den Cursor auf die erste Zeilen- bzw. Spaltenangabe, lesen den Zeilenwert in der Kopfzeile ab und positionieren den Cursor auf die letzten Zeilen der Spaltenangabe in der Liste. Diese zwei Werte subtrahieren Sie, bilden den Betrag davon und zählen 1 dazu. Anschließend erstellen Sie eine Querverweisliste zu diesem Text. Dazu setzen Sie den Cursor in der Querverweisliste auf »Zeile:??« und »Spalte:??«, lesen den Wert ab, positionieren den Cursor auf die letzte Angabe (Zeile:?? Spalte:??) und führen die o.a. Berechnung durch. Bei Tabulatortaste stellen Sie bitte »Leerzeichen« ein, falls noch nicht geschehen. Außerdem sollte dies schon in der Info-Datei, wie oben erwähnt, geschehen sein. Um Attribute im Text zu setzen, merken Sie sich besser die Kombi-

nationen über die Tastatur, weil dies wesentlich schneller geht.

Und nun zur Fußnotenverwaltung noch einige Worte. Diese Einrichtung ist sehr wichtig für Referate und wissenschaftliche Texte. Vor und hinter dem Text der Fußnote muß ein invertierter Stern stehen, sonst erhalten Sie eine Fehlermeldung.

Formatkommandos:

Der wichtigste Teil bei »Documentum« sind die Formatkommandos. Hier werden die häufigsten Ursachen für einen fehlerhaften Ausdruck zu finden sein.

Zur Randeinstellung gibt es viele Möglichkeiten: Die Seitengrößen, -maßeinheiten und -einstellungen erfolgen generell in dem »Zeige Druck« bzw. »Ausdruck«-Requester. Hier allein findet die Eintragung der Werte statt.

Auch können Sie Ihren Text manipulieren. Geben Sie z.B. Erläuterungen zu einem Text ab, dann wählen Sie das Kommando »ESC/lz'Erläuterungen'« (ESC ist die Escape-Taste, links oben). Nun wird der ganze nachfolgende Text um diese »Erläuterungen«-Länge nach rechts eingezogen. Der Text, der aber noch vor dem Einrücken kommen soll, muß vor dem Kommando stehen. Eine Stellung auf gleichem Niveau ist nur durch Abschätzung möglich. Das heißt, wenn Sie »Erläuterungen:« und den Text nach dem Doppelpunkt fortlaufen lassen und ab der nächsten Zeile eingerückt hinter dem Doppelpunkt fortfahren möchten, dann müssen Sie abschätzen, welche Wörter noch hinter den Doppelpunkt passen und wann das ESC-Kommando folgen muß. Das gleich gilt für den rechten Rand.

Beachten Sie, daß Formatkommandos nur in einer neuen Zeile stehen können und sie durch ein RETURN abgeschlossen sein müssen. Mit Blocksatz, links- oder rechtsbündig und zentriert legen Sie Ihr Satz bild fest. Haben Sie allerdings bei Ihrem Drucker per DIP-Schalter oder über das »ESC/dk«-Kommando Proportionaldruck angewählt, funktioniert das Ganze nicht mehr. Proportionaldruck ist eine Ausdruck, bei

dem Buchstaben wie z.B. das »i«, »l« etc. nicht den gleichen Platz beanspruchen wie z.B. ein »b«. Auch bei der zentrierten Ausgabe werden Sie über den Proportionaldruck Schwierigkeiten bekommen. Diese Kommandos werden allgemein mit einer »1« bzw. einer »0« hinter dem Kommando aktiviert bzw. inaktiviert.

Falls die »ESC-pr«-Einstellung (Proportionaldruck) nicht funktioniert, nehmen Sie das »ESC/dk27,112,1«-Kommando zum Ein- und das »ESC/dk27,112,0«-Kommando zum Ausschalten des Proportionaldrucks. Das »/« tippen Sie nicht mit. **Achtung:** Proportionaldruck ist mit Blocksatz nicht kombinierbar.

Zeichensätze:

Selbstverständlich können Sie mit »Documentum« andere Zeichensätze verwenden. Allerdings ist es mit »Documentum« nicht möglich, den Druckerfont mit anderen Fonts zu kombinieren. Das heißt, möchten Sie einen Grafikfont, wählen Sie ihn im Menü »Fonts« an. Über das »ESC/ft« Kommando bestimmen Sie andere Fonts. Eine Rückkehr zum Druckerfont, oder überhaupt dessen Verwendung mit Grafikfonts ist allerdings nicht möglich.

Druckseiteneinstellung:

Den Seitenumbruch können Sie nun selbst in die Hand nehmen. Geben Sie z.B. das »ESC/ns«-Kommando und eine Null dahinter ein, wird ab dieser Stelle mit einer neuen Seite begonnen. Steht hinter dem Kommando eine andere Zahl, werden entsprechend der Zahl, soviel Zeilen zusammenhängend auf einer Seite ausgedruckt. Dieser Wert darf nie größer sein, als der Wert der Zeilen, die Ihre Seite aufnehmen kann.

Kopf- und Fußzeilen geben Sie entweder im Druck-Requester ein, oder über Escape-Kommandos im Text. Die Hierarchie der Befehle ist folgende: Im Text befindliche Befehle stehen über den angewählten Kommandos im Druck-Requester.

Bei der Kopf- und Fußzeilenbestimmung können Sie Seitenzahlen mit eingeben. Dieses Zeichen erhalten Sie durch gleichzeitiges Drücken von <ALT s>. Die Kopf- und Fußzeilen sind in drei Bereiche aufgeteilt: links, Mitte und rechts. Tippen Sie zwischen diesen Abteilungen das

»|«-Zeichen (Kombination <SHIFT >>), links neben der Backspace-Taste).

Sie haben die Möglichkeit, Ihrem Drucker bestimmte Steuerzeichen zu senden, das sind z.T. Escape-Sequenzen (27, etc.). Diese haben nichts mit dem Escape-Zeichen bei »Documentum« zu tun. Den Blocksatz wählen Sie über o.g. Code, doppelt hohe Zeichen bei Epson-kompatiblen Druckern über »ESC/dk27,119,1« an. Bitte beachten Sie auch hier die Hierarchie. Direkte SteuerCodes stehen über den vom Programm angewählten »Documentum«-Escape-Sequenzen.

Falls die Seitengrößeneinstellung nicht richtig funktioniert, helfen folgende Werte bei DIN-A4-Seiten weiter: Seitenanfang 16 mm, links 30 mm, rechts 5 mm, unten 4 mm. Seitenlänge 277 mm, Seitenbreite 203 mm. Um Proportionaldruck zu erreichen, geben Sie o.g. Kommando ein.

Speichern Sie diese Einstellungen unter »DIN_A4.frm« und laden Sie sie vor dem Ausdruck.

Beschriften von Briefumschlägen

Hier noch ein Tip, um Briefumschläge mit Adressen zu bedrucken. Die Einstellungen lauten Anfang 50 mm, links 90 mm, rechts 10 mm und unten 4 mm. Die Seitenlänge beträgt 100 mm und die Breite 203 mm. Diese Werte sollten Sie unter »Briefumschlag.frm« speichern.

Vor jedem Ausdruck laden Sie dann im Requester diese Voreinstellungen und können ihn dann ausdrucken oder sich den Text ansehen.

»Documentum« bietet sehr viele Möglichkeiten und ist kompatibel zu DTP-Programmen, da es im ASCII-Format Texte speichert.

Um »Documentum«-spezifische Steuerzeichen beim Speichern zu vernachlässigen, wählen Sie im »Speichern«-Requester einfach »mit Steuerzeichen« an, worauf sich das »mit« sofort in ein »ohne« verändern wird.

Wir hoffen, Ihnen einige Anregungen zur effektiveren Ausnutzung des Programms vermittelt zu haben. ag

Wordperfect

So geht's richtig

von Dr. Rudolf Egg

Die Textverarbeitung Wordperfect und Amiga sind wie gemacht füreinander. Die deutsche Version 4.1 bietet umfangreiche, professionell gestaltete Funktionen für praktisch alle Bereiche. Sie ist allerdings nicht völlig identisch mit der gleichnamigen MS-DOS-Version 4.1. Wer also neben seinem Original-Handbuch noch eine weitere Anleitung zu Rate ziehen möchte, z.B. den praktischen Kurzführer von Data Becker, wird dort nicht immer das finden, was er sucht. Erst recht muß derjenige die jeweiligen Besonderheiten beachten, wer mit den verschiedenen Wordperfect-Umsetzungen arbeitet. Hier sind die wesentlichen Unterschiede:

Die praktische Hilfe-Funktion von Wordperfect wird beim Amiga nicht über <F3>, sondern (sinnvollerweise) über die HELP-Taste aufgerufen. Mit <F3> schalten Sie beim Amiga dagegen die Kursivschrift ein, die bei MS-DOS nicht als eigene Funktion implementiert ist. Die Tastenkombination <CTRL F9> steuert beim Amiga nur die Funktion »Mischen«; ein »Sortieren« von Texten ist dagegen nicht vorgesehen. Außerdem gibt es wegen des anderen Zeichensatzes des Amiga bei der Funktion »Linien zeichnen« (<CTRL F2> und <1>) gewisse Einschränkungen.

Demgegenüber verfügt die Amiga-Version über eine Thesaurus-Funktion (<ALT F1>), mit der sich zu einzelnen Wörtern Synonyme und Antonyme anzeigen lassen. Dies ist für MS-DOS-Rechner erst ab Version 4.2 möglich. Schließlich sieht Wordperfect-Amiga die Möglichkeit einer einfachen Cursorsteuerung mittels des Mauszeigers vor, und alle Funktionen sind neben der F-Tasten-Steuerung zusätzlich über

Wordperfect ist ein Textprogramm, das fast keine Wünsche offenläßt. Wir wollen Ihnen mit einigen Tips und Tricks helfen, dieses umfangreiche Anwendungspaket auch sinnvoll zu nutzen.

Pull-Down-Menüs aufrufbar. Dieser für den Amiga eigentlich selbstverständliche Komfort wurde für MS-DOS-Anwender erst in der neuesten Version 5.1 realisiert.

Ein letzter, vermeintlicher Mangel der Amiga-Version ist gar keiner: das Fehlen der Funktion »Fenster« im Bildschirm-Menü (<CTRL F3>) von Wordperfect. Das hier vermißte Öffnen eines Fensters für die Bearbeitung einer zweiten Textdatei ist beim Amiga besser und komfortabler über die nachfolgenden Funktionen möglich:

Zunächst definieren Sie über den Menüpunkt »Umschalten« (<SHIFT F3>) einen neuen Textbereich. Anders als bei MS-DOS-Rechnern erlaubt der Amiga hier aber nicht nur eine, sondern – je nach verfügbarem Speicherplatz – bis zu 32 weitere, in Bearbeitung befindliche Textdateien, zwischen denen hin- und hergeschaltet wird. Als Fenster innerhalb

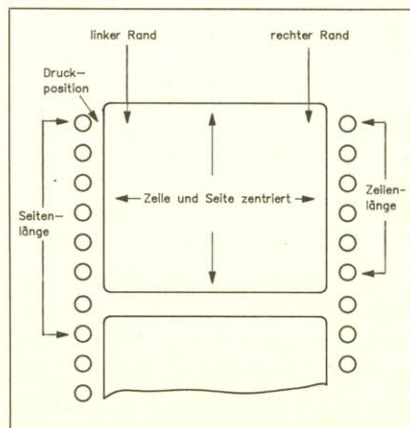


Bild 1. Die schematische Darstellung eines Etikettendrucks

von Text 1 holen Sie sich einen zweiten, in der genannten Weise geöffneten Text durch Klick auf das zusätzliche Schließsymbol von Wordperfect an der oberen Bildschirmleiste. Diesen vorerst nur kleinen Ausschnitt vergrößern und plazieren Sie mit der Maus nach Belieben. Durch einfaches Anklicken der jeweiligen Fensterfläche aktivieren Sie den wechselweisen Zugriff auf die gleichzeitig dargestellten Textfenster. Auch Blockoperationen (Kopieren, Verschieben usw.) zwischen diesen beiden Teilen gelingend denkbar einfach.

Was das Handbuch verschweigt

Einige Besonderheiten von Wordperfect Amiga, die im praktischen Gebrauch sehr nützlich sein können, werden im Handbuch nicht oder nur dürftig erläutert. Dies betrifft beispielsweise die Steuerung des Cursors, für die es einige zusätzliche Möglichkeiten gibt. Die genaue Auflistung finden Sie in Tabelle 1.

Etwas mühsam ist bei Wordperfect das Verschieben von Textblöcken, vor allem, wenn nur einzelne Wörter bewegt werden sollen. Bevor man die erforderlichen Tastenkombinationen, <ALT F4> und Cursor-tasten (Blockdefinition), <CTRL F4> und <1> (Verschieben aufrufen) und <CTRL F4> und <5> (Text abrufen), gedrückt hat, ist der gewünschte Text meist schon von Hand eingegeben. Doch es geht auch einfacher:

Angenommen, Sie wollen in einem Satz ein Wort umstellen. Dazu löschen Sie zunächst dieses Wort, indem Sie den Cursor auf einen beliebigen Buchstaben dieses Wortes plazieren und anschließend <CTRL DEL> oder <CTRL ←> drücken. Danach bewegen Sie den Cursor an die Stelle, an der Sie das soeben ge-

löschte Wort haben möchte und holen es sich mit <F1> und <1> wieder zurück - fertig. Der Trick dabei ist, daß »Wordperfect« gelöschte Textteile in einem dreistufigen Löschspeicher aufbewahrt, der sich auch für solche Textmanipulationen verwenden läßt. Ähnlich einfach gerät das Kopieren von Wörtern. Sie müssen dazu den mit <F1> abrufbaren Löschspeicher lediglich mehrmals benutzen. Wollen Sie mit diesem Trick nicht nur einzelne Wörter, sondern ganze Zeilen bewegen, so löschen Sie den zu verschiebenden Text mit <CTRL End> und verfahren anschließend wie beschrieben.

Etikettendruck leicht gemacht

Wer viel mit seinem Amiga arbeitet, braucht auch viele Disketten. Um Ordnung zu halten, sollte man auf eine klare und übersichtliche Beschriftung achten. Doch nicht jeder verfügt über eine leserliche Handschrift. Dies ist auch gar nicht nötig, denn wozu haben Sie denn eine Textverarbeitung? Allerdings sind beim Etikettendruck mit Wordperfect einige Punkte zu beachten, die wir kurz erklären möchten.

Zunächst einmal benötigen wir spezielle Lochrand-Etiketten für 3 1/2-Zoll-Disketten (sog. Computer-Labels), die es im Fachhandel gibt. Die üblichen Einfach-Etiketten sind für einen Drucker ungeeignet. Spannen Sie anstelle Ihres herkömmlichen DIN-A4-Endlospapiers diese Etiketten in Ihren Drucker ein - dabei Stachelwalze oder Traktor engstellen. Nun messen Sie die Breite Ihrer Etiketten von der linken zur rechten Kante (ohne den Lochrand) in Inch bzw. Zoll. Sollten Sie kein Lineal mit Incheinteilung haben, so rechnen Sie nach folgender Formel um:

1 Inch = 2,54 cm

1 cm = 0,394 Inch

Eine Etikettenbreite von 70 mm entspricht etwa 2,76 Inch. Dieses

Cursorsteuerung	
Taste	Bedeutung
<ALT Cursor nach links>	Zeilenanfang
<ALT Cursor nach rechts>	Zeilenende
<SHIFT Cursor nach links>	Zeilenanfang
<SHIFT Cursor nach rechts>	Zeilenende
<ALT Cursor nach oben>	Bildschirm zurück
<ALT Cursor nach unten>	Bildschirm vor
<CTRL Cursor nach oben>	Seite zurück
<CTRL Cursor nach unten>	Seite vor
<SHIFT Cursor nach oben>	Textanfang
<SHIFT Cursor nach unten>	Textende

Tabelle 1. Die Cursorsteuerung von Wordperfect auf einen Blick

Meßergebnis multiplizieren Sie mit der für Ihren Druck vorgesehenen Anzahl der Zeichen pro Inch (»cpi«). Als Standard gelten dabei 10 cpi, Schmalschriften arbeiten mit 12 oder 15 cpi. In unserem Beispiel ergäbe sich danach für die Standardschrift eine Breite von 2,76 x 10 Zeichen, also rund 27 Zeichen. Dies ist Ihre Zeilenbreite, die Sie nun zu Beginn eines neuen Textes und unter Berücksichtigung des Abstands der Druckposition Null von dem linken Etikettenrand, im Menü Zeilenformat (<SHIFT F8>) bei Option »3« (Ränder) eintragen sollten. Wählen Sie dort folgende Werte: links 3 Zeichen, rechts 27 Zeichen. Gleichzeitig setzen Sie den oberen Blattrand (<ALT F8> und <5>) auf null Halbzeilen. Zur Verdeutlichung dieses Systems betrachten Sie bitte Bild 1. In Ihrem Mustertext schreiben Sie jetzt einige Zeilen mit Ziffern von 1 bis 9 vom linken bis zum rechten Rand Ihres Bildschirms. Am einfachsten beginnen Sie dabei mit der Ziffer »3« - Ihrem linken Rand. Wenn Sie wollen, benutzen Sie dabei zur Arbeitersparnis auch die Kopierfunktion ganzer Zeilen mit dem schon beschriebenen Trick. Das Ergebnis könnte etwa so aussehen:

3456789112345678921234567

3456789112345678921234567

3456789112345678921234567

3456789112345678921234567

3456789112345678921234567

Sie sollten dabei mindestens so viele Zeilen eingeben, daß Ihr Musterdruck zwei Etiketten überschreibt. Vor dem Druck dieses Rasters positionieren Sie Ihren Druckkopf an die oberste Stelle (Zeile) des ersten Etiketts. Mit <SHIFT F7> und <1> drucken Sie anschließend

dieses Testmuster aus. Weil noch die Standard-Seitenlänge (72 Zeilen) eingestellt ist, erhalten Sie dabei am Ende einen großen Vorschub an Leerzeilen; bewegen Sie daher gegebenenfalls Ihre Endlosetiketten wieder etwas zurück.

Zählen Sie nun die Anzahl der Druckzeilen vom oberen Rand des ersten Etiketts bis zur ersten Zeile des zweiten Etiketts. Dies ist die neu zu definierende Seitenlänge (z. B. »17«), die Sie am Anfang Ihres Files mit <ALT F8> und <4> und <3> eingeben. Die ebenfalls dort einstellbare Zeilenzahl Ihres Etiketten-Files ermitteln Sie aus der Anzahl der Zeilen, die Sie pro Etikett tatsächlich beschreiben möchten (beispielsweise »9«). Nun fehlen noch die endgültigen Werte für den linken und rechten Rand. Dazu überprüfen Sie das Druckergebnis und stellen mit <SHIFT F8> und <3> die von Ihnen gewünschten neuen Werte für beide Ränder ein, etwa »5« und »25«. Damit Ihr File nicht unübersichtlich wird, sollten Sie vorher mit <SHIFT F3> einen Blick in die Steuerzeichen tun und dort zunächst die bisherigen Einstellungen mit löschen. Für das ordentliche Aussehen Ihrer Etiketten empfiehlt sich noch eine weitere Formatdefinition: das Zentrieren zwischen oberem und unterem Blattrand mit <ALT F8> und <3>. Im Steuerzeichenfenster (<ALT F3>) sollte Ihre erste Zeile jetzt etwa so aussehen:

[RdEinst:5,25][Ob.Blrd:
0][SeitLg:17,9][SeiteZentr]

Sind diese Vorarbeiten abgeschlossen, geben Sie den Text für Ihr erstes Etikett ein. Wordperfect wird dabei wie üblich am eingestellten rechten Rand einen Zeilenumbruch

vornehmen und auch das Seiteneende durch eine gestrichelte Linie markieren. Selbstverständlich können Sie – mit <RETURN> und <CTRL RETURN> – auch feste neue Zeilen und feste neue Seiten definieren, wobei letzteres einen Vorschub zum nächsten Etikett veranlaßt. Nach zwei oder drei Seiten bzw. Etiketten sollten Sie einen Probeausdruck starten. Sofern Sie mit dem Ergebnis zufrieden sind, dürfen Sie die Einstellungen als Musterfile für spätere Anwendungen speichern, andernfalls ändern Sie die bisherigen Definitionen für Rand, Länge und Format nochmals. Zur besseren Übersicht empfiehlt es sich dabei wiederum, die alten Werte im Steuerzeichenfenster zu löschen.

Etiketten-Styling

Die Beschriftung Ihrer Disketten sieht ansprechender aus, wenn der Text nicht nur seitenweise, sondern auch zwischen linkem und rechtem Rand zentriert erscheint. Weil der entsprechende Wordperfect-Befehl (<SHIFT F6>) aber nur zeilenweise wirksam ist, kann dessen Verwendung während der Texteingabe recht mühsam sein, vor allem dann, wenn Sie viele Etiketten zu beschriften haben. Dies läßt sich aber leicht umge-

hen, wenn Sie zunächst nur die Textzeilen Ihrer Etiketten schreiben, anschließend das gesamte File als Block definieren (<ALT F4> aufrufen) und für diesen Block alle Zeilen auf einmal mit <SHIFT F6> zentrieren. Vielleicht möchten Sie auf jedem Etikett einen bestimmten Text als Überschrift oder unterste Zeile, z.B. Ihren Namen, ausdrucken. Sie könnten sich dafür mit <CTRL F10> ein Makro definieren und dieses mit <ALT F10> bei jeder neuen Seite aufrufen. Wordperfect hat dafür aber noch eine viel elegantere Lösung, nämlich die Funktion »Kopf-/Fußtext«. Diese wird, wiederum am Anfang Ihres Textfiles, jedoch nach den Formatdefinitionen, mit <ALT F8> und <6> aufgerufen. Sie haben dabei die Wahl zwischen je zwei Kopf- und Fußzeilen (»A« und »B«). Die Texteingabe erfolgt auf einem separaten Bildschirm und wird mit <F7> beendet. Nach der Rückkehr zu Ihrem Haupttext ist zunächst keine Veränderung zu bemerken. In den Steuerzeichen (<ALT F3>) sind jedoch das entsprechende Kürzel und der Anfang der soeben geschriebenen Textzeile enthalten:

```
[Kopf- /Fußtext:n,n;text]
```

Der gesamte Text wird dagegen erst beim Druck auf der ersten bzw.

der letzten Zeile Ihrer Etiketten ausgegeben. Beachten Sie aber, daß sich Ihre verfügbare Seitenlänge stets durch die in den Kopf- und Fußzeilen vorhandenen Zeilen reduziert. Außerdem fügt Wordperfect automatisch noch eine Leerzeile zwischen Kopf-/Fußtext und Textanfang bzw. -ende ein. Da Ihnen das Seiteneende jedoch automatisch angezeigt wird, behalten Sie leicht die Übersicht.

Soll auf Ihre Etiketten eine fortlaufende Nummer gedruckt werden, hilft Ihnen Wordperfect auch hierbei. Der Trick ist, daß Sie an jeder Stelle im Text die Seitenzahl neu definieren können und daß sich dieser Wert auch als fortlaufende Nummer für Ihre Etiketten ausdrucken läßt. Zuerst rufen Sie hierfür mit <ALT F8> und <2> den Menüpunkt »Neue Seitenzahl« auf und geben in dem entsprechenden Fenster eine bis zu fünfstellige Zahl ein. Diese erscheint anschließend sofort als »Seite ...« am unteren Bildschirmrand von Wordperfect. Nun drücken Sie, am besten in Ihrer Kopfzeile (in <ALT F8> Option »Bearbeiten Kopfzeile« wählen), an der für Ihre Etikettennummer vorgesehenen Stelle, die Tastenkombination <CTRL B>, die als »^B« auf dem Bildschirm erscheint. Beim späteren Ausdruck wird dieser Code durch die Seitenzahl ersetzt. ag

Sonix Rock on the rocks

Das wohl bekannteste Musikprogramm für den Amiga ist »Sonix«. Trotz leichter Bedienung und guter Übersichtlichkeit bleiben immer noch Fragen offen.

von Martin Jobst

Auf den folgenden Seiten bieten wir eine kleine Sammlung praktischer Erfahrungen, die Ihnen den Umgang mit »Sonix« erleichtern soll. Nützliche Tips, um die Fähigkeiten des Programms besser zu nutzen.

IFF-Instrumente

Sonix bietet die Möglichkeit, Instrumente sowohl im IFF-Format als auch in einer speziellen Sonix-Form zu verarbeiten. Letztgenannte bietet den Vorteil, verschiedene Manipulationen am Instrument vornehmen zu können. Was tun, wenn ein Sound nur im IFF-Format vorliegt?

Ein Weg, IFF-Sounds ins Sonix-Format zu konvertieren, führt über Programme (etwa »Audiomaster«), die eine Option bieten, Instrumente im Sonix-Format zu speichern. Es geht jedoch wesentlich einfacher: Laden Sie einen IFF-Sound (etwa »IFF Electric Bass«, wählen Sie »Save as...« im Projekt-Menü und klicken Sie »Save« an. Das Instrument wird unter dem Namen erneut gespeichert – im Sonix-Format. Bei erneutem Laden des Instruments stehen Ihnen die Einstellmöglichkeiten eines Sonix-Instruments offen. Hintergrund: Während »Save as...« in den anderen Sonix-Modi dazu dient, ein File unter einem neuen Namen zu speichern, wird die Funktion bei

geladenem IFF-Instrument zum Konverter.

Dynamik

So vielfältig die Fähigkeiten von Sonix sind, so knausrig ist das Programm, was Notationssymbole betrifft. Das man keine dynamischen Veränderungen der Lautstärke innerhalb eines Musikstücks setzen kann, ist ein Manko. Besonders bei Soloparts schränkt dies den Spielraum des Musikers stark ein.

Ein Ausweg ist das Abspeichern von Instrumenten in verschiedenen Lautstärken. Sie speichern z.B. den Vibraphon-Klang für ein Solo unter »1Vibraphon«, »2Vibraphon« etc. mit verschiedenen Lautstärkeinstellungen. Je nach Bedarf setzen Sie auf der Partitur (den Notenzeilen) das Vibraphon mit der gewünschten Lautstärke ein. Keine Angst: Der Speicherbedarf eines Stücks steigt dadurch kaum, da Sonix das (speicherintensive) Sample nur einmal im Speicher hält und für die verschiedenen Einstellungen lediglich einige zusätzliche Bytes benötigt.

Klangfülle

Den letzten Schliff erhalten Soli aber erst durch einen weiteren Trick. Bedingt durch die beschränkte Anzahl an Tonkanälen, die je einer der vier Sonix-Spuren zugeordnet werden, klingen Melodien oft abgehackt und stumpf. Das rührt daher, daß ein Ton, bevor er ausklingen kann, oft schon von der folgenden Note einer Spur abgelöst wird. Jedes Piano-Solo etwa wird dadurch zunichte gemacht. Wir müssen daher einen Weg finden, eine gebundene Spielweise - genannt Legato - zu erzeugen.

Die Lösung liegt darin, eine Melodie auf zwei Spuren aufzuteilen. Sie legen einfach jeden zweiten Ton auf eine zweite Spur. Damit die Tempi stimmen, müssen die jeweils nun fehlenden - weil auf der anderen Spur - Noten durch entsprechende Pausen ersetzen. Unser Ziel ist aber immer noch nicht erreicht, da die Töne durch die Pausen erneut keine Gelegenheit zum Ausklingen haben. Um bei der Melodie die aufeinander folgenden Töne ineinander übergehen zu lassen, ersetzen wir einfach jede Note auf beiden Spuren mit ei-

ner Note doppelten Werts. Damit das Timing wieder stimmt, müssen natürlich die den verlängerten Noten folgenden Pausen um den Wert verkürzt werden, um den der Ton »gestreckt« wurde.

In der oberen Hälfte von Bild 1 sehen Sie ein einfaches Beispiel einer Melodie in Viertelnoten, die untere Hälfte zeigt deren Aufteilung auf zwei Spuren. Da alle Noten der Melodie gleichwertig sind, fallen fast alle Pausen weg, die bei der Aufteilung gesetzt werden.

Durch die Aufteilung einer Tonfolge auf zwei Spuren steht zwar eine Spur weniger für andere Einsätze zur Verfügung, das akustische Ergebnis gleicht diesen Verlust aber allemal aus. Wenn Sie zusätzlich noch einen Stereoeffekt erzielen wollen, sollten Sie eine Melodie so aufteilen, daß sie auf zwei akustisch »gegenüberliegenden« Spuren - etwa 1 und 3 - landet.

Synthesizer

Einige Tips zum eingebauten Synthesizer von Sonix.

- Grundschiwingung: Mit dem Sonix-Synthesizer können Sie die Wellenform der Grundschiwingung freihand mit der Maus zeichnen. Sollten Sie von dieser hervorragenden Eigenschaft Gebrauch machen, ist zu beachten, daß die Welle möglichst grobe, klare Formen hat. Sie verleihen damit dem Grundklang die

nötige Prägnanz. Detailreiche Wellen bringen meist keine Klangverbesserung und machen den Sound verschwommen.

- Vibrato: schnelle, zyklische Veränderung der Frequenz. Der Effekt läßt sich mit Hilfe des LFO realisieren, dessen Wellenform idealerweise ein Sinusschwingung sein sollte. Der Schalter im LFO-Fenster muß auf »On« stehen, die Einstellung von Speed sollte im oberen Drittel angesiedelt sein. Mit »LFO« im Freq-Fenster regeln Sie die Stärke des Vibrato.

- Tremolo: schnelle, zyklische Veränderungen der Lautstärke. Der Effekt wird mit »LFO« im Fenster Amplitude geregelt. Die anderen Einstellungen sind analog zum Vibrato.

- Wha-Wha-Effekt: Dieser von Gitarristen gern verwendete Effekt läßt sich durch zyklisches Verändern der Grenzfrequenz des Filters mit Hilfe des LFO erreichen. Der Schalter im LFO-Fenster sollte auf »On« stehen, »Speed« wie gewünscht. »LFO« im Filter-Fenster regelt den Einfluß des LFO auf die Cutoff-Frequenz und damit die Stärke des Effekts.

Der LFO orientiert sich an der mit »Freq« eingestellten Grenzfrequenz des Filters. Daher sollte der Regler »Freq« keine Extremwerte annehmen, da sonst für den LFO kein Spielraum mehr bleibt und der Effekt behindert wird. Die Wellenform für den LFO ist idealerweise eine Sinusschwingung.

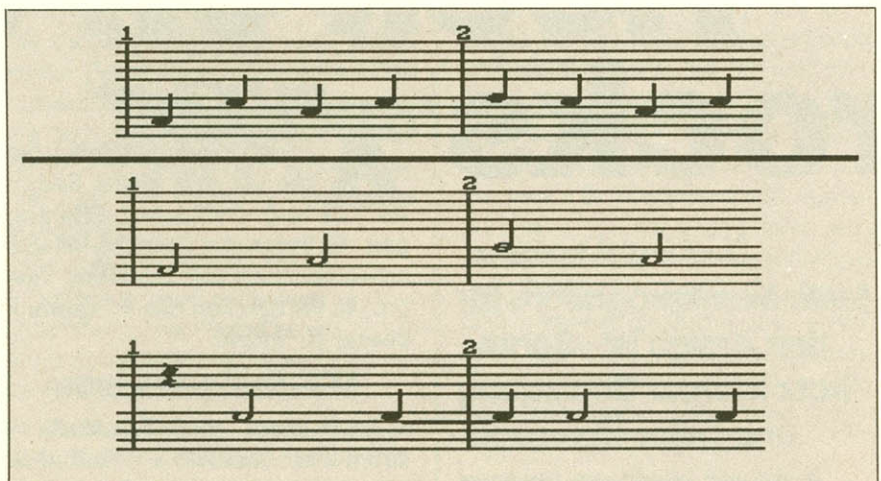


Bild 1. So teilen Sie eine Melodie auf zwei Spuren auf

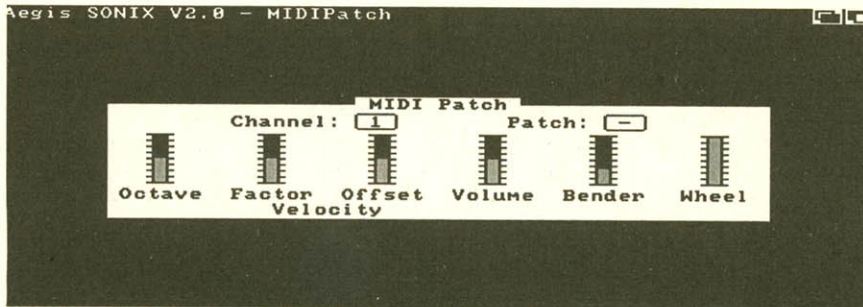


Bild 2. »Octave« des MIDI-Patch besitzt verborgene Qualitäten

Editieren

Die Schalter für die Spuren von Sonix besitzen drei Aktivierungszustände. Sehr hilfreich beim Editieren ist jener Modus, in dem der Schalter blau eingefärbt ist. In diesem Zustand wird die Spur beim Abspielen nicht nur in halber Lautstärke gespielt, der Modus hat den Vorteil, daß zum Editieren einer Stimme auch die restlichen Spuren angezeigt werden können: Wollen Sie bei Spur 1 Veränderungen vornehmen, schalten Sie diese in den »schwarzen« Modus, bei den restlichen Spuren (wo dies gewünscht ist) aktivieren Sie den Zustand »blau«. Sie können Spur 1 editieren und sehen zur besseren Orientierung zusätzlich die anderen Spuren.

MIDI

Über eine fantastische und zugleich recht unbekanntere Fähigkeit verfügt das MIDI-Patch. Neben der Möglichkeit, MIDI-Geräte von Scores aus anzusprechen, kann es Sie auch unterstützen, wenn Sie »eigenhändig« auf der Klaviatur Ihres Keyboards spielen.

Das Geheimnis verbirgt sich hinter dem Regler »Octave« des MIDI-Patch (Bild 2). Während er normalerweise dazu dient, die Oktave festzulegen, mit der Noten eines Score vom MIDI-Gerät gespielt werden, hat er auch noch eine andere Funktion. Laden Sie dazu im »Instruments«-Modus das MIDI-Patch. Stellen Sie über ein MIDI-Interface die übliche Verbindung zwischen dem Amiga und Ihrem MIDI-Keyboard her. Wenn Sie jetzt auf der Klaviatur des Keyboards spielen, hat sich im Grunde

nichts verändert. Das ist aber nur so lange der Fall, bis Sie die Stellung des Reglers »Octave« verändern. Dann nämlich begleitet Sonix Ihr Spiel auf der Klaviatur in anderen Tonlagen. Jedesmal, wenn Sie einen Ton anschlagen, erzeugt der Amiga zusätzlich – je nach Stellung von »Octave« – denselben Ton eine bzw. zwei Oktaven höher oder tiefer. Ein interessanter Effekt, der die Klangfülle erheblich verbessern kann. »Bender« sollte dabei auf Mittelposition stehen, »Wheel« üblicherweise in der Nullstellung.

Samples

Ein Tip für alle, die die Qualität selbst gesampelter Instrumente verbessern wollen. Für die meisten Fälle genügt es, von einem Musikinstrument einen Ton in mittlerer Tonhöhe zu digitalisieren und dann von einem entsprechenden Programm (etwa »Audiomaster«) ein mehroktaviges Instrument erzeugen zu lassen. Bei bestimmten Instrumenten (etwa einem Klavier) geht aber mit diesem Verfahren die Klangauthentizität verloren. Das hat folgenden Grund: Ein »natürliches« Instrument (nehmen wir wieder das Klavier) besitzt in seinen verschiedenen Tonlagen eine unterschiedliche Klangcharakteristik. Ein Ton in den unteren Oktaven eines Klaviers (etwa C) unterscheidet sich von einem C4 also nicht nur in seiner Frequenz. Ein deutlicher Unterschied ergibt sich auch in der Klangfarbe.

Wird daher nur ein Ton aus der gesamten Bandbreite eines Klaviers gesampelt, müssen alle anderen Töne dieser und der darüber und darunter liegenden Oktaven berechnet, d.h. durch Frequenzerweiterung, erzeugt werden. Bei dieser rein mathematischen Methode wird – und kann auch gar nicht – das unterschiedliche

Klangbild in verschiedenen Tonlagen nicht berücksichtigt: Das Klavier klingt also nicht mehr wie ein Klavier im »richtigen Leben«.

Der theoretisch günstigste Weg, dieses Übel zu beseitigen, wäre nun, jeden einzelnen Ton eines Klaviers separat zu digitalisieren. Daß dieses Verfahren allein schon aus Speicherplatzgründen zum Scheitern verurteilt ist, leuchtet ein. Ein guter Kompromiß ist, von jeder Oktave einen Ton zu digitalisieren. Dann brauchen beim Abspielen nur mehr die fehlenden Töne der jeweiligen Oktave rekonstruiert zu werden. Eine erheblich verbesserte Klangtreue ist das Ergebnis.

Das Digitalisieren je eines Tones pro Oktave ist technisch kein Problem. Schwierig wird es, diese fünf Oktaven zu einem Sonix-Instrument zusammenzufassen. »Audiomaster« unterstützt diese Methode leider nicht. Trotzdem gibt es einen Weg, und zwar mit der Unterstützung eines PD-Programms: »Stic V1.0« von »Ruhr 7«.

Wenn Sie von einem Instrument jeweils einen Ton pro Oktave in mittlerer Tonhöhe sampeln und mit »Stic« zu einem Instrument zusammenfassen, werden Sie erstaunliche Verbesserungen in der Klangqualität Ihrer Instrumente erreichen. Nähere Hinweise für die Arbeit mit »Stic« entnehmen Sie der Anleitung, die auf der PD-Diskette beigefügt ist. *ag*

In eigener Sache

Der Verfasser dieses Beitrags ist Mitautor des kürzlich bei Markt & Technik erschienenen Buches »Sonix Workshop«. Das Werk setzt sich umfassend mit »Sonix« auseinander und deckt alle Bereiche des Programms ab. Zentraler Punkt ist das umfassende Tutorium, das sich der praktischen Arbeit mit Sonix widmet und auch die erforderliche Musiktheorie nicht zu kurz kommen läßt. Der komplette Referenzteil ist eine unentbehrliche Hilfe im Umgang mit Sonix. Andere Abschnitte beschäftigen sich mit Fehlerbehandlung, Zusatz-Hard-/Software, Kniffen für Profis (natürlich ausführlicher als im Rahmen des vorliegenden Artikels möglich) etc. Das beiliegende Notenheft unterstützt die Entwicklung des im Tutorium entstehenden Musikstücks.

Martin Jobst/Horst Mitmansgruber: Sonix Workshop, Markt & Technik 1990, ISBN 3-89090-897-7, 39 Mark

Trick für Trick

von Winfried Dietmayer

Alle folgenden Ausführungen beziehen sich auf die Version 3.02 des Superbase-Professional-Entwicklerpakets und wurden auch mit dieser getestet.

Verwendung des eigenen Editors für DML-Programme

Der Programm-Editor von Superbase Professional stellt zwar die wichtigsten Funktionen für das Entwickeln von Programmen zur Verfügung, doch wer möchte nicht seinen bevorzugten Editor zum Programmieren einsetzen? Das stellt bei einem multitaskingfähigen Computer wie dem Amiga kein Problem dar, da man Superbase und Editor gleichzeitig laufen lassen kann. Doch wie gestaltet man das wiederholte Laden des fertigen Programms in Superbase, um es auszutesten? Folgende Vorgehensweise hat sich als optimal herausgestellt (wir setzen voraus, daß sowohl Superbase als auch der zu verwendende Editor bereits gestartet wurden):

1. Belegen Sie eine Funktionstaste mit dem Befehl »RUN "Pfad/Programmname"«.

Beispiel: Ihr Programm hat den Namen »Buchliste« und befindet sich im Verzeichnis »dh2: Buchbestand«. Wir wollen die Funktionstaste »F2« mit dem Befehl belegen.

Wählen Sie im »Einstell.«-Menü den Punkt »Funktionstasten/Edieren« an. Klicken Sie auf »F2« und aktivieren Sie den Cursor in der Eingabezeile. Geben Sie nun »RUN "dh2: Buchbestand/Buchliste"« <RETURN> ein. Bestätigen Sie Ihre Belegung, indem Sie »OK« anklicken.

2. Speichern Sie das Programm im Editor als ASCII-File ab, verlassen den Editor aber nicht!

ein Hit

DML ist eine sehr mächtige Programmiersprache für Superbase. Aber auch sie kann nicht alle Wünsche der Programmierer befriedigen. Lesen Sie, wie diese Wünsche doch noch in Erfüllung gehen.

3. Aktivieren Sie das Superbase-Fenster oder den Superbase-Screen. Jetzt können Sie mit der Funktionstaste Ihr Programm vom DML-Interpreter ausführen lassen.

4. Bei notwendigen Programmänderungen nehmen Sie diese in Ihrem Editor vor und fahren wieder mit 2. fort.

Beachten Sie bitte, daß der Superbase-Editor das Programm im »Token-Format« abspeichert, d.h. jedes Befehlswort, z.B. »SELECT NEXT«, wird als »Kennziffer« abgespeichert. Es ist also nicht möglich, ein im Superbase-Editor erstelltes und mit dem Menüpunkt »Programm/Speichern« abgespeichertes Programm mit einem Fremdeditor zu bearbeiten. Hierzu muß man das Programm zuerst mit dem Befehl »SAVE "Pfad/Programmname", TEXT« als ASCII-Datei abspeichern. Sie können diese Anweisung entweder über den Menüpunkt »Programm/Befehl« eingeben oder Sie belegen auch hierfür eine Funktionstaste. Jetzt läßt sich das Programm in Ihren Editor laden. Befindet sich jedoch an jedem Zeilenende das Steuerzeichen »^M«, so versteht Ihr Editor nur die Zeilenschaltung mit dem

ASCII-Zeichen »10« (Linefeed). Superbase nimmt aber die PC-übliche Zeilenschaltung »10,13«, »Linefeed« und »Carriage Return« vor. Um dies auf Amiga-Standard zu ändern, muß in den Tooltypes des Superbase-Programm-Piktogramms die Option »RETURN=OFF« eingestellt werden. Dies erreicht man über den Menüpunkt »Info« der Workbench. Die Änderung wirkt jedoch erst nach dem nächsten Neustart von Superbase.

Führt Superbase ein im ASCII-Format vorliegendes Programm aus, so übersetzt es zuerst die ASCII-Zeichen in ein Token-Format. Dieser Konvertierungsprozeß kostet vor allem bei großen Programmdateien viel Zeit. Sie sollten deshalb ein fertiggestelltes Programm mit Hilfe des Superbase-Editors im Token-Format abspeichern. Die Ladezeit verringert sich so um ca. 40 Prozent.

Automatisches Verschlüsseln von ».sbp«-Dateien

Bei Superbase Professional besteht die Möglichkeit den Sourcecode eines DML-Programms zu verschlüsseln, um vor Manipulationen oder allzu großer Neugier geschützt zu sein. Der Befehl hierzu lautet »PROTECT "Dateiname"«. Leider muß hierzu die Programmdatei im Token-Format vorliegen. Versuchen Sie niemals eine Programmdatei im ASCII-Format mit »PROTECT« auf

die oben geschilderte Art zu verschlüsseln. Es hat die sofortige Zerstörung der Datei zur Folge!

Eine Verschlüsselung einer ASCII-Datei ist nur auf folgende Art und Weise möglich:

Setzen Sie in jede zu verschlüsselnde Datei den Befehl »PROTECT« (ohne Argument!!) an den Anfang oder besser in die zweite Zeile, da beim Einladen eines verschlüsselten Programms in den Superbase-Editor der Inhalt der ersten Zeile angezeigt wird, sofern es sich um eine »REM«-Zeile handelt. Die Programmdatei wird dann beim ersten Aufruf mit »RUN« oder »CHAIN« verschlüsselt. Wohlgemerkt, das Programm ist erst dann verschlüsselt, wenn es mindestens einmal ausgeführt wurde! Bedenken Sie weiterhin, daß die Verschlüsselung nicht rückgängig zu machen ist, erstellen Sie also vorher eine Sicherungskopie. Diese Methode der Verschlüsselung kann sehr mühsam sein, wenn es gilt viele Programm-Module zu verschlüsseln. Deshalb haben wir ein kleines Utility entwickelt, das diese Aufgabe übernimmt (siehe Listing 1 »Do_Protect.sbp«). Speichern Sie hierzu alle DML-»sbp«-Dateien, die Sie verschlüsseln wollen, in einem eigenen Verzeichnis ab und geben dieses Verzeichnis (kompletter Pfad) dann bei der Programmfrage »Verzeichnis für Sourcmodule?« ein. Bei der nächsten Programmfrage »Verzeichnis für Protectmodule?« geben Sie ein Verzeichnis Ihrer Wahl an. Die Verschlüsselung kann bei vielen oder sehr großen Dateien einige Zeit in Anspruch nehmen. Sie finden dieses hilfreiche Tool auf unserer Programmservice-Diskette.

Simulation eines rudimentären »Data Dictionary«

Leider bietet auch Superbase kein »Data Dictionary« an, in dem alle Dateidefinitionen, Konsistenzbedingungen, Views und Verknüpfungen einer Datenbank abgelegt werden können. Zumindest hat man die Möglichkeit, über den Menüpunkt »System/Status/Datei« sich die Felderdefinition einer Datei und eventuell vorhandene Konsistenzbedingungen (»Formel«) anzeigen zu las-

sen. Meistens besteht eine Datenbank jedoch aus mehr als aus einer Datei. Um sich halbwegs einen Überblick zu verschaffen, muß man für jede Datei den oben genannten Menüpunkt anwählen. Was nur dann funktioniert, wenn man vorher die entsprechende Datei geöffnet (»Projekt/Öffnen/Datei«) hat. Bei einer Datenbank, die z.B. aus 15 Dateien besteht, ein äußerst umständliches und zeitraubendes Verfahren. Deshalb haben wir auch hierzu ein kleines Utility entwickelt, daß die Definitionen aller in einem Verzeichnis befindlichen Dateien auf dem Drucker ausgibt: »Data_Dic« (finden Sie auf der Programmservice-Diskette zum Heft). Achten Sie bitte darauf, daß Ihr Drucker eingeschaltet und »online« ist, bevor Sie das Programm starten.

Error-Logbuch

DML bietet eine brauchbare Methode zur Fehlerbehandlung: Die »ON ERROR GOTO/GOSUB«-Anweisung. Bei der Applikationsentwicklung ergibt sich dennoch ein zusätzliches Problem: Ist das Pro-

Fehleranalyse

gramm erst einmal beim Endanwender installiert und treten dann Fehler auf, so kann nachträglich meist nicht mehr genau festgestellt werden, um welchen Fehler es sich handelt. Genau dies wäre aber zur Behebung der Fehlerursache von großem Nutzen. Abhilfe kann hier der Einbau eines »Error-Logbuchs« schaffen, in dem der aufgetretene Fehler und wann er aufgetreten ist, festgehalten wird. Wie realisiert man dies in der Praxis? Zuerst legen wir eine Datei an, die die Fehlerinformationen speichern soll, nennen wir sie »Error«. Die Dateidefinition könnte folgendermaßen aussehen:

Feldname	Attribute	Format	Ort	Formel
Fehlernr	NUM;IXD	999.	0	0
Fehlertext	TXT	50	1	0
Modul	TXT	20	2	0
Datum	DAT;IXD	dd mmm yyyy	3	0
Zeit	TIM	hh:mm	4	0

Die Felder haben folgende Bedeutung: »Fehlernr« enthält die von DML erzeugte interne Fehlernummer, »Fehlertext« den korrespondierenden Text, »Modul« den Namen des Programms, in dem der Fehler aufgetreten ist, »Datum« und »Zeit« das

Systemdatum bzw. die Systemzeit. In jedem Modul, das zu einer Applikation gehört, sieht die Integration des Logbuchs so aus (unterstellter Programmname: »buchneu«):

```
ON ERROR GOTO bug
.
.
bug:
OPEN FILE "Error"
BLANK
Fehlernr = ERRNO
Fehlertext = ERR$( ERRNO )
Modul = "buchneu"
Datum = TODAY
Zeit = NOW
STORE
REQUEST "Programmfehler: " +
LEFT$(ERR$( ERRNO ),33),
"Bitte Entwickler informieren!"
,O,a%
CLOSE FILE "Error"
RESUME cleanup
.
.
cleanup:
CLOSE ALL
RUN "prog:hauptmenu"
```

Die »ON ERROR GOTO bug«-Anweisung legt fest, welche Marke im Fehlerfalle angesprungen wird und sollte möglichst am Anfang eines Moduls stehen, da vor ihr keine Fehlerbehandlung möglich ist. Tritt ein Fehler auf, so werden alle Befehle des Modulteils »bug« ausgeführt: Sie öffnet die Datei »Error« und erzeugt einen neuen Datensatz. Beachten Sie bitte, daß »ERRNO« eine Systemvariable ist, die die letzte Fehlernummer enthält, und »ERR\$()« eine Funktion, die den entsprechenden Fehlertext hierzu liefert. Mit »STORE« wird der Datensatz in die »Error«-Datei gespeichert und mit »REQUEST« der Benutzer informiert, daß ein Fehler aufgetreten ist. Die Anwei-


```

1 REM `Do_Protect' V.1.2.2 (c) by Eikel + Dietmayer 19.04.1990
2 REM -----
3 REM Verschlueseln aller Pgm.dateien
4 REM V.1.2.2 Verschlueseln durch Einfuegen von `Protect' in
5 REM die zweite Zeile des Sourcecode's
6 REM -----
7
8 REM Hoffen wir, dass wir nie mehr als 50 Dateien haben...
9 DIM sb_Datei$(50)
10 REM und kein Pgm > 2000 Zeilen....
11 DIM P_Datei$(2000)
12 PANEL OFF
13
14 start:
15 ON ERROR GOTO check
16 Source$=" ": Schutz$=""
17 SET PAGING ON
18 DISPLAY;
19 CLS
20 HOME
21 ? @25,2 "Verschl"+CHR$(252)+"sseln von Quellmodulen"
22 ? @25,3 "=====

```

```

23 NEWLINE 2
24 ? @10 " Bitte kopieren Sie alle Programmdateien, die Sie
    ver-"
25 ? @10 "schl"+CHR$(252)+"sseln wollen in ein eigenes Verzeich-
    nis - am besten"
26 ? @10 " in die Ramdisk - um eine optimale Geschwindigkeit zu
    er-"
27 ? @10 "reichen."
28
29 REM Frage Quell- und Zielverzeichnis ab
30 dialog:
31 WHILE TRIM$(Source$) = ""
32 HOME
33 ASK @5,12 "Verzeichnis f"+CHR$(252)+"r Sourcemodule? " &32;
    Source$
34 WEND
35
36 dialog1:
37 WHILE TRIM$(Schutz$) = "" OR Schutz$ = Source$ 38 HOME
39 ASK @5,14 "Verzeichnis f"+CHR$(252)+"r Protectmodule? " &32;
    Schutz$
40 ? Schutz$

```

sung »RESUME cleanup« setzt die Bearbeitung bei der Marke »cleanup« fort, die – nach dem unvorhergesehenen Fehler – mit »run prog:hauptmenu« zum Hauptmenü zurückkehrt.

Durch das Auswerten der »Error«-Datei wird jeder Fehler festgehalten und kann damit zur Fehlerbehebung herangezogen werden.

Die multitaskingkonforme Menüprogrammierung

Superbase bietet die Möglichkeit einfach eigene Pull-Down-Menüs zu definieren. Der Applikationsprogrammierer steht nun vor dem »Problem« auszuwerten, ob und welchen Menüpunkt der Benutzer ausgewählt hat. Die klassische Lösung dieses Problems besteht darin, daß man in einer Endlosschleife abfragt, was der Benutzer ausgewählt hat. In DML sieht das ungefähr so aus:

```
MENU CLEAR
```

```
REM Menüdefinition
```

```
MENU 1,0,1, "Hauptpunkt 1"
```

```
MENU 1,1,1, "Unterpunkt 1"
```

```
.
```

```
.
```

```
MENU 3,0,1, "Hauptpunkt 3"
```

```
REM Endlosschleife zur Auswahlabfrage
```

```
auswahl:
```

```
MENU ON spalte%, zeile%
```

```
IF spalte% <> 0 THEN
  IF spalte% = 1 THEN
    ON zeile% GOSUB up1,
up2 ...
  END IF

```

```

.
.
IF spalte% = 3 THEN
  ON zeile% GOSUB up3,
up4...
  END IF
ELSE
  GOTO auswahl
END IF

```

Solange kein Menüpunkt ausgewählt wird, ist die numerische Variable »spalte%=0« und der »ELSE«-Zweig wird ausgeführt: »GOTO auswahl«. In einem Singletasking-Betriebssystem reicht dieser Ansatz aus, da es ausgeschlossen ist, daß noch ein weiteres Programm Prozessorzeit beansprucht. Nicht so beim Amiga. Will man »nebenbei« noch andere Aufgaben erledigen, so führt diese Endlosschleife zu einem merklichen Performance-Verlust des Gesamtsystems: Die Menüsleife »frißt« unnötig Rechenzeit, die man für die anderen Programme gut gebrauchen könnte. Deshalb stellt DML eine vorbildliche Möglichkeit zur Verfügung, das Message-System des Amigas auszunutzen: den »WAIT«-Befehl. In unserem Zusammenhang ist der »WAIT MENU«-Befehl interessant. Er deaktiviert (im Amiga-Jargon: Der Task wird in den »waiting state« versetzt) den Superbase-Program-Task solange, bis eine Menüauswahl erfolgt ist und nimmt dann die Arbeit wieder auf. Solange

kein Menü angewählt wird, verbraucht er keine Rechenzeit. Unser DML-Codefragment sieht dann, multitaskingkonform programmiert, folgendermaßen aus:

```
auswahl:
MENU ON spalte%, zeile%
```

```
REM Warte auf Anwahl eines Menüpunktes
WAIT MENU
```

```

IF spalte% = 1 THEN
  ON zeile% GOSUB up1,
up2 ...
  END IF

```

```

.
.
IF spalte% = 3 THEN
  ON zeile% GOSUB up3,
up4...
  END IF

```

Beachten Sie, daß der Task solange auf der »WAIT MENU«-Anweisung stehen bleibt, bis eine Auswahl erfolgt. Bei einer Anwahl erfolgt die Auswertung dann nur einmal. Es ist kein Schleifenkonstrukt mehr notwendig. pe

Schweigen ist Gold,

SILVER macht mehr

von Marco Vitolini-Nandini

Die zahlreichen Manipulationsmöglichkeiten zur Einstellung der Objekteigenschaften sind bei »Turbo Silver« sicherlich die herausragenden Merkmale. Dabei lassen sich neben den Materialfarben auch die Spiegelstärke und Transparenz für jede Grundfarbe einzeln definieren. Soll ein Objekt durchsichtig erscheinen, kann man zusätzlich einen Lichtbrechungskoeffizienten einstellen. Aber damit nicht genug. Oberflächenrauheit läßt sich ebenso bestimmen, wie die Stärke des Glanzlichts, was die dargestellten Objektmaterialien wie Glas, Plastik oder Metall noch realistischer macht. Neben diesen Features besteht noch die Möglichkeit, ein Objekt mit einer Textur zu überziehen. Statt einer einzigen Objektfarbe kann man dem Körper beliebige Musterungen, ja sogar Bilder zuordnen. In »Turbo Silver« existieren hierfür zwei unterschiedliche Texturarten, die eine als IFF-Mapping bezeichnet, die ein beliebiges IFF-Bild auf einen Körper projiziert. Die andere als mathematische Funktion, die vielfältige Musterungen auf das Objekt überträgt.

IFF-Mapping:

In »Turbo Silver« läßt sich jedes Bild (außer Extra-Halfbrite-Bilder) jedem Objekt zuordnen. Hierbei bleiben alle Objekteigenschaften wie Reflektion oder Transparenz vollständig erhalten. Bis zu acht verschiedene Bilder lassen sich pro Szene einsetzen. Um nun die Bilder den ausgewählten Objekten zuzuordnen, stehen vier verschiedene Projektionsarten zur Verfügung.

»Flat X & Z« legt das Bild flach von vorne auf das Objekt.

»Wrap X«, »Wrap Z« und »Wrap X & Z« wickeln das Bild an den jeweiligen Achsen um den Körper. Speziell bei runden Objekten oder Rotationskörpern sollte man diese Einstellung wählen, da sonst das Bild an den Objektseiten zu stark verzerrt erscheint. Die einzelnen Schritte, um ein IFF-

Bild einem Körper zuzuordnen, werden bereits ausgiebig anhand eines Beispiels im Handbuch von »Turbo Silver« erläutert.

Die andere Texturart, die auf mathematischen Funktionen beruht, wird hingegen nur sehr oberflächlich beschrieben. Waren es in der Programmversion 3.0 noch zwei Texturen – Bricks (Ziegelstein) und Checks (Schachbrettmuster) – existieren seit dem Update zur neuen Version 3.0 SV fünf weitere, Grid (Gittermuster), Marble (Marmor), Wood (Holz), Angular (Farbverlauf) und Disturbed (3D-Interferenzmuster). Da sich die Update-Dokumentation über die neuen Texturen ausschweigt, sind viele Experimente angesagt, um diese mehr oder weniger wunschgemäß einsetzen zu können.

Da die neue »Turbo Silver«-Version über einen Objekteditor im Interlace-Modus verfügt, benötigt man vor dem Programmstart mindestens 420 KByte freies Chip-RAM, was viele Benutzer, die noch nicht über 1 MByte Chip-RAM verfügen, davon abhält, sich mit dem Update anzufreunden. In vielen Fällen wandert die 3.0 SV-Version wieder in den Schrank und die neuen Texturen somit auch. Diese lassen sich aber ebenso im alten »Turbo Silver« ver-

wenden. Man muß sie sich nur auf die entsprechende Programmdiskette zu den alten Texturen kopieren.

Schritt für Schritt wollen wir nun die Texturen mit ihren dazugehörigen Parametern kennenlernen.

CHECKS (Schachbrettmuster)

Eine der einfachsten Texturen ist die Schachbrettmusterung. Hiermit lassen sich quadratische Felder beliebiger Größe auf ein Objekt übertragen. Im »PARAMETERS«-Requester des »Textur«-Menüs können Sie sieben Eingaben tätigen:

»Check Size« bestimmt die Größe eines einzelnen Schachfeldes. Bevor man hier eine Eintragung vornimmt, sollte man sich über die Größe des mit dieser Textur zu überziehenden Körpers informieren. Am einfachsten geht dies mit der Funktion »Coordinates ON« aus dem »View«-Menü. Fahren wir also mit dem Koordinatenkreuz der Maus die Ausmaße des Objekts ab, können wir die Anfangs- und Endkoordinaten des Körpers in einem Extra-Fenster ablesen. Nun sollte man sich noch überlegen, wie viele Felder sich über eine Achse des Objekts erstrecken sollen. Diese Zahl teilt man mit der vorher

errechneten Körpergröße und erhält somit den Wert für die Größe eines quadratischen Feldes, die bei »Check-Size« im »Parameters«-Requester eingetragen werden muß. Die nächsten drei Parameter (»RGB Red«, »RGB Green«, »RGB Blue«) bestimmen die eine Farbe der Schachbrettmusterung (die andere Farbe wird im »Attribute«-Requester im »Settings«-Menü des Objekts eingestellt). Hier sind Werte zwischen 0 und 255 für die drei Grundfarben anzugeben. Um diese Farbe nicht blind eintragen zu müssen, kann man sie probeweise im »Attribute«-Requester einstellen und die Werte aufschreiben. Bleiben noch die drei letzten Parameter, die die Schachbrettmusterung im dreidimensionalen Raum verschieben können. Werden hier alle drei Werte auf Null belassen, befindet sich im Koordinatenursprung des Objekts (»Axis«-Punkt) genau eine Ecke der einzelnen Quadrate. Will man, aus welchen Gründen auch immer, diese verschieben, lassen sich hier Koordinaten eingeben, die das Muster um deren Betrag im Raum verschieben (Ähnlich dem »TRANS«-Befehl im »Settings«-Menü bei den Objekten). Abschließend bleibt noch zu sagen, daß diese Textur auf flachen Objekten am besten wirkt. Möchte man sie trotzdem auf Kugeln oder ähnlichen Körpern abbilden, entstehen durch die Dreidimensionalität der Textur (unerwünschte?) Musterungen. Um diese möglichst gering zu halten, sollte der Wert »Check Size« immer ein Vielfaches von 32 sein.

GRID (Gitter)

Ähnlich der »Checks«-Textur erzeugt diese ebenfalls quadratische Felder, die aber durch eine Art Gitternetz voneinander getrennt sind. Man kann den Gitterlinien eigene Materialeigenschaften zuordnen, und dies ist sicherlich die Stärke dieser Textur. So ließe sich beispielsweise das Gitter vollkommen transparent, also unsichtbar, machen, womit man lauter einzelne im Raum schwebende Quadrate erhielte. Die Parameter im einzelnen:

»Grid Size« legt den Abstand von einer Gitterlinie bis zur nächsten fest (incl. der Größe des Quadrats).

»Line Size« bestimmt dagegen die Stärke einer einzelnen Gitterlinie.

Wird der »Line Size«-Wert gleich oder größer als der »Grid-Size«-Wert gewählt, erscheint der Körper vollständig in der für die Gitterlinien gewählten Materialeigenschaft. Diese wird in den nächsten 14 Parametern eingestellt. Dabei entsprechen alle Eingabefelder denen des gewohnten »Attribute«-Requesters. Vielleicht sollte man einer schon bekannten Materialeigenschaft die Werte aufschreiben, um sie dann in die Felder einzutragen.

BRICKS (Ziegelstein)

Ebenfalls mathematisch definierbar ist eine Art Gitterstruktur, die sich durch Verschieben der Struktur nach jedem Gitterfeld, sowie einer entsprechenden Farbgebung, wie eine Ziegelsteinmauer aussieht. Hierbei symbolisieren die Gitterfelder die einzelnen Ziegel und die Gitterlinien den dazwischenliegenden Mörtel.

Die Farbe der Ziegelsteine wird im »Attribute«-Requester des Objekts festgelegt. Die Mörtelfarbe, also die Farbe des Gitters, wird in den ersten drei Parametern der Textur bestimmt. Die nächsten drei Eingaben bestimmen die Breite, Tiefe und Höhe der einzelnen Ziegelsteine, inklusive einer Mörtelbreite. Der Parameter »Mortar Size« ist für die Dicke des Mörtels zuständig. Hier sollten möglichst Werte eingesetzt werden, die einer Ziegelsteinwand ähnlich erscheinen. So wären die Ausmaße eines Ziegels beispielsweise Breite=30, Tiefe=15, Höhe=10 und

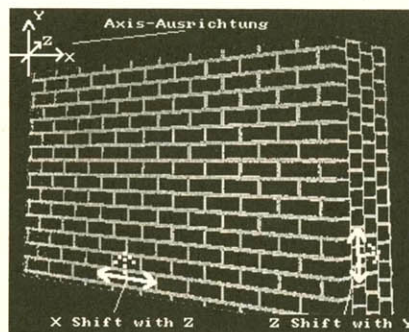


Bild 1. Die Bricks-Texture verlangt ein gutes räumliches Vorstellungsvermögen

Mörtel=2. In weiteren drei Parametern läßt sich die Mauerstruktur definieren. Ein Maurer würde sicherlich niemals alle Ziegel exakt übereinanderlegen, sondern diese in jeder neuen Reihe um einen gewissen Wert nach links oder rechts versetzen. Genau dies verfolgt der achte Parameter »X Shift with Z«, also den Versatz nach links oder rechts mit zunehmender Höhe. Angenommen, wir wollten jede zweite Ziegelreihe genau übereinanderlegen, so müssen wir hier die Hälfte des Werts von »XSize« (also 15) eingeben. Der nächste Parameter »X Shift with Y« definiert die Verschiebung einer Ziegelsteinreihe nach links oder rechts in Abhängigkeit mit der Tiefe. Sollte das Objekt also dicker als eine Ziegelsteintiefe sein, so sollte man auch von diesem Wert Gebrauch machen. Falls die Wand doch dicker als eine Ziegelsteintiefe sein sollte, so kann man mit dem Parameter »Z Shift with Y« eine weitere Verschiebung, nämlich der nach oben und unten pro Ziegelsturm festlegen. Hier wäre beispielsweise die Hälfte des Werts »Z-Size« (also 6) sinnvoll. In Bild 1 werden diese Verschiebungsparameter noch einmal verdeutlicht. Bleiben noch drei Parameter übrig. Hiermit läßt sich, wie bei der »Checks«-Textur, der Ursprung der Struktur im dreidimensionalen Raum verschieben. Dies kann nötig werden, wenn sich die Axis des Körpers so im Raum befindet, daß sich eine Wandseite genau im Mörtelbereich befindet und somit keine Ziegelstruktur sichtbar wird. Sollte eine Wand stark »verbröseln« wirken, befindet sie sich genau an der Grenze zwischen Mörtel und Ziegelstein. Hier sollte man den entsprechenden Offset um einige Zehntelwerte verschieben. Weiterhin sollte man beachten, daß sich die Axis des Körpers gleichwertig zu den Raumachsen verhält. (X=Breite, Y=Tiefe und Z=Höhe). Verwendet man beispielsweise eine »SURFACE« als Objekt, muß diese erst um 90° um die X-Achse gedreht werden, damit sie aufrecht in der Szene steht. Dabei vertauscht sich natürlich auch die Y-Achse mit der Z-Achse, was bei der Parameterwahl der »BRICKS«-Textur zu beachten ist.

MARBLE (Marmor)

Eine Textur ganz anderer Art ist die Marmor-Textur. Hier werden Maserungen einer bestimmten Farbe wie Adern in die Grundfarbe des Körpers eingelegt. Diese laufen mehr oder weniger geradlinig entlang der Y-Achse in die Tiefe der Szene. Die ersten drei Parameter des Marmors bestimmen die Farbe dieser Maserungen. Der nächste Wert, »Layer Spacing«, bestimmt den Abstand der einzelnen Maserungsstreifen an der X-Achse. Je kleiner dieser Wert gewählt wird, desto näher liegen diese beieinander.

Der Parameter »Exponent« bestimmt die Stärke dieser Streifen. Je kleiner dieser Wert wird, desto breiter werden die Maserungen und umgekehrt. Der Parameter »Shift X« dürfte wohl keine Schwierigkeiten mehr bereiten. Hiermit läßt sich die Maserung an der X-Achse entlang nach links oder rechts verschieben. Interessant ist sicherlich der vorletzte Parameter, »Variance«, der mit seinen möglichen Werten zwischen 0 und 1 die Marmorierung zunehmend unregelmäßiger erscheinen läßt. Abschließend läßt sich mit einer Zufallszahl (»Random Seed«) die Variance weiter verstärken.

WOOD (Holz)

Die Holz-Textur ist mit dem Marmor stark verwandt. Anstatt einzelne, parallel an der X-Achse liegende Maserungen zu erzeugen, werden hier konzentrische Kreise an der X-Z-Achse gebildet. Diese ähneln somit den Jahresringen eines Holzstücks. Auch hier gelten die ersten drei Parameter der Maserungsfarbe. Die Grundfarbe des Körpers wird natürlich im »Attribute«-Fenster definiert. Der »Ring Spacing«-Parameter ist dem »Layer Spacing« des Marmors verwandt. Auch hier wird der Abstand zwischen den einzelnen Maserungen eingestellt. Hier wirkt sich der Wert allerdings auch auf die Z-Achse aus. »Exponent«, »Variation« und »Random Seed« sind mit den gleichlautenden Parametern der Marmor-Textur identisch. Ebenso die »Shift...«-Parameter, mit der zusätzlichen Möglichkeit, die Holzstruktur

mit einem »Shift-Z«-Parameter in der Höhe zu verschieben.

ANGULAR (Farbenpracht)

Einen ganz neuen Weg wird mit der »ANGULAR«-Textur beschritten. Mit ihr hat der Anwender die Möglichkeit, ganze Farbverschiebungen auf der Objektoberfläche vorzunehmen. Insgesamt können sechs verschiedene Farben definiert werden. Diese befinden sich immer an einem Objektrand (ganz oben, ganz unten, ganz vorne, ganz hinten, ganz links und ganz rechts vom Objekt). Diese Farben vermischen sich dann über den ganzen Körper hinweg.

Farbkaleidoskop

Somit lassen sich fantastische Effekte erzielen, zumal eine Animation mit solch einem Objekt (oder gar nur seiner Axis) für wunderschöne animierte Farbverläufe sorgt. Die 15 einzugebenden Parameter repräsentieren die Rot-, Grün- und Blauwerte jeder einzelnen Farbe. Diese sind mit »Start-« und »End-« Kennzeichnung jeder Achse versehen. Auffällig ist die fehlende »Start-X«-Farbe. Diese wird aber im Attributes-Requester des Objektes eingestellt. Sie steht für die links-außen-Farbe des Körpers. Die »End-X«-Parameter setzen die Farbe rechts-außen des Objekts fest. Des Weiteren steht »Start-Y« für ganz vorne, »End Y« für ganz hinten, »Start Z« für den untersten Teil und »End Z« für den obersten Teil des Objekts – natürlich nur bei gleicher Ausrichtung der Objektachse zur Raumachse. Will man weniger als sechs Farben einsetzen, so lassen sich auch gleiche Farbwerte in den einzelnen Parametern setzen.

DISTURBED (Verzerrt)

Die schönste Textur bildet den Abschluß unserer Reise durch »Turbo Silver«-World. Sie ist eine der komplexesten und auch am schwierigsten gezielt einzusetzenden Texturen. Die Ergebnisse entschädigen aber für den Aufwand und die Mühe. Es lassen sich nämlich dreidimensionale, wellenartige Strukturen auf eine Oberfläche legen. Die anzugebenden Parameter wirken mit den im »Attribute«-Requester gemachten Einstellungen für »Specular« und »Hardness« zusammen. Ebenso können mehrere, unterschiedlich positionierte und farbige Lichtquellen

das Erscheinungsbild dieser Textur verändern. Sehen wir uns dazu die Funktionsweise dieser Textur an. Trifft ein Lichtstrahl auf ein Objekt, so wird ein bestimmter Winkel zur Lichtquelle errechnet, der dann, im Zusammenspiel mit den Materialeigenschaften des Körpers, die gewünschte Schattierung erzeugt. Kommt nun die »DISTURBED«-Textur hinzu, wird dieser errechnete Winkel zyklisch in veränderbarer Stärke verstellt. Das Ergebnis läßt den Körper nicht mehr glatt, sondern geriffelt aussehen. Der Vorteil liegt auf der Hand: Mußte man bisher solche Oberflächen mühsam modellieren, kann man nun diese Textur verwenden. Je nach Einstellung der »Specular«- und »Hardness«-Regler, kann man die zusätzlich hervorgerufenen Highlights verstärken oder abschwächen. Nun aber zu den Parametern:

Mit »Amount« läßt sich die Intensität, mit der die Textur auf die Oberfläche wirken soll, einstellen. Hier sind Werte zwischen »Null« und »Eins« zugelassen, wobei »Null« keine und »Eins« die volle Intensität hervorruft. Der Parameter »Wavelength«, die Wellenlänge, bestimmt den Abstand zwischen zwei aufeinanderfolgenden Tälern oder Bergen der Wellenstruktur. Die Wellenphase läßt sich durch den dritten Parameter, »X Separation«, definieren. Man könnte diese Funktion mit einer Verschiebung auf der X-Achse gleichsetzen. Zu guter Letzt bleibt noch die »Amplitude«, also die Wellenhöhe. Auch hier sind wieder nur Werte zwischen »Null« und »Eins« zugelassen, wobei »Null« keine Wellenhöhe und »Eins« maximale Wellenhöhe bedeutet. Mit diesen Texturen lassen sich in »Turbo Silver« Bilder voller Kreativität und neuem Look erstellen (siehe Bild 2). Andere Programme gleichen Genres geraten somit immer mehr ins Hintertreffen.

Wir hoffen, daß nun auch Sie mit den kreativsten Werkzeugen von »Turbo Silver« umgehen können und diese Funktionen in Ihre neuen Schöpfungen einfließen lassen. Let's texture! pe

von Walter Friedhuber

Wer sich für Farborgien begeistern kann und schon die Grundprinzipien der Farblehre kennenlernen wollte, der wird mit der folgenden Übung auf beste bedient. Insgesamt sechs Lampen werden dazu eingesetzt, um Bild 1 zu erstellen.

Starten Sie die Übung in einem leeren Tri-View. Aktivieren Sie das DOWN-Fenster, indem Sie in das Innere der oberen Fensterleiste klicken.

Schalten Sie die Koordinatenanzeige ein:

```
EDIT-COORDINATES
```

Stellen Sie die Facettenfarbe auf ein mittleres Grau ein.

```
SCULPT 3D:  EDIT -- COLOR
SA4D      :  EDIT -- MODIFY --
              FACES
```

Verschieben Sie alle drei Farbräder (RGB) in die Mitte des Feldes. Schließen Sie den Requester mit »OK«. Damit liegt die Farbe für die »CIRCLE« (Kreis-)Form fest.

Lesen Sie die Standardform »Kreis« ein:

```
EDIT -- ADD -- CIRCLE
```

Es erscheint ein Requester, in dem Sie die Anzahl der Seiten eingeben sollen, aus denen sich das Polygon zusammensetzt (SA4D-Meldung: »How many points around circle (3 or more?)«, SCULPT 3D-Meldung: »How many subdivisions (3 or more)«).

Löschen Sie die vorgegebene Zehn und ersetzen Sie diese durch eine Vier. Drücken Sie dann die <RETURN>-Taste.

SCULPT benutzt die Frage dazu, um feststellen zu können, welche Art Vieleck (Polygon) Sie generieren möchten. Würden Sie die Minimalangabe von drei Seiten akzeptieren, erhalten Sie ein Dreieck. Bei Eingabe einer Vier wird demnach ein Viereck kreiert usw. Wird hingegen eine Zehn eingetippt, bildet SCULPT ein kreisförmiges Gebilde. Diese Eigenschaft der »CIRCLE«-Form kann dazu benutzt werden, auch komplizier-

Bild 1. Ein Lichtermeer auf einem gebohnerten Parkett – mit SCULPT kein Problem

te Gebilde aufzubauen. Wie aber uns schwer zu erkennen ist, besteht die generierte Form nur aus einer Anzahl von verbundenen Eckpunkten. Facetten sind keine vorhanden. Aus diesem Grund stellt SCULPT ein Kommando zur Verfügung, das es gestattet, eine Form auch nachträglich mit Facetten zu füllen.

Schalten Sie die Eckpunkte des Vierecks ein:

```
EDIT -- SELECT -- ALL
```

Befehlen Sie

```
EDIT -- DO -- FILL
```

Das »FILL«-Kommando arbeitet nur, wenn Eckpunkte angeschaltet, diese mit Kanten verbunden sind und auf derselben Ebene liegen. Zusätzlich müssen alle aktivierten Kanten eine oder mehrere geschlossene Schleifen bilden. Eine Schleife ist ei-

Im SCULPT Rampen licht

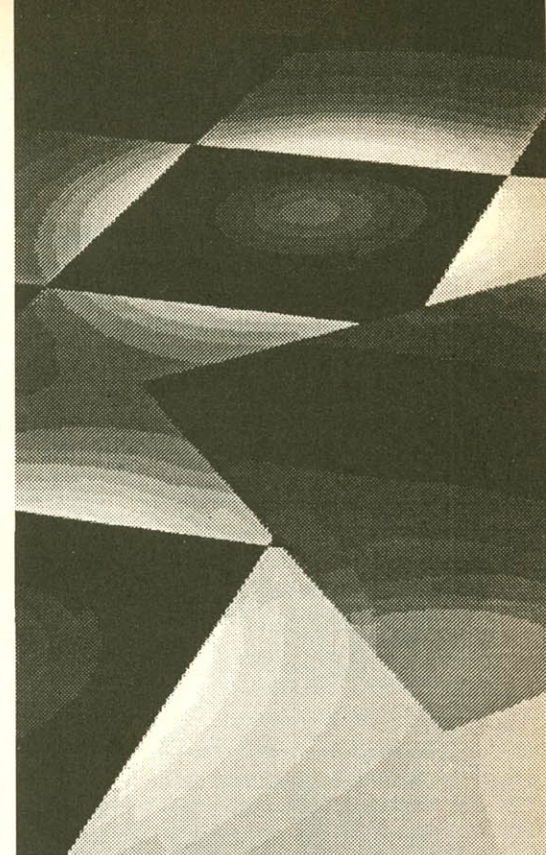
ne geschlossene Form, wie etwa unser Viereck oder ein Kreis.

Diese Schleifen dürfen sich nicht überschneiden. »FILL« erzeugt neue Ecken, so daß das Innere der Figur (der Schleife) in Dreiecke (Fazetten) unterteilt wird. Liegt nun aber eine Schleife innerhalb einer anderen, größeren Schleife, so wird diese nicht gefüllt und es entsteht ein Loch. Ein Effekt, der nützlich ist, wenn aus einem kreisförmigen Objekt ein Rad entwickelt wird. Dazu müssen Sie lediglich einen größeren Kreis definieren (»SUBDIVISION 10«), darin einen kleineren anordnen und »EDIT-DO-FILL« befehlen.

Positionieren Sie nun das Fadenkreuz auf den Target-Punkt des Vierecks (Zentrum) im »DOWN«-Fenster (Draufsicht). Aktivieren Sie durch einen Mausklick das »GRABBER«-Gadget und verschieben Sie den Cursor auf die Koordinaten

```
East   :    3800
North  :    3800
Up     :         0
```

Schalten Sie den »GRABBER« durch einen Klick mit der rechten Maustaste aus.



Im SCULPT

SCULPT 3D zählt neben Turbo-Silver zu den weit verbreitetsten 3D-Programmen. Von Einsteigern hört man immer wieder, daß Ihnen der Umgang mit farbigen Lichtquellen Probleme macht. Dem kann abgeholfen werden.

Diese Aktion ist erforderlich, um das Viereck auf dem später einzufügenden Schachbrettmuster (»GROUND-CHECKERED«) positionieren zu können.

Fixieren Sie dann den Zielpunkt der Kamera:

OBSERVER -- TARGET

In SCULPT 3D schaltet daraufhin die Koordinatenanzeige sofort auf 0 zurück, da der Targetpunkt gleichzeitig den absoluten Nullpunkt des Koordinatenfensters repräsentiert.

Jetzt wird das »Tri-View« zentriert. Klicken Sie dazu auf das »CENTER TRI-VIEW«-Gadget. Das »Tri-View« bewegt sich und der Cursor wird in dessen Mittelpunkt plaziert.

Zur Bestimmung der Größe, der im Schachbrettmuster generierten Quadrate, klicken Sie dreimal das »CONTRACT TRI-VIEW«-Gadget an.

Rufen Sie »WORLD - GROUND« auf und tippen Sie zweimal in das »GROUND TYPE«-Fenster, bis dort »CHECKERED« angezeigt wird. Die obere Farbe wird auf Weiß gesetzt, wozu Sie die drei Regler R, G und B auf den äußersten rechten Rand des Feldes ziehen. Dann klicken Sie die zweite (untere) Farbe im Farbanzeigefenster an und verstellen die Regler so, daß ein dunkelgrauer Farbton angezeigt wird. Dunkelgrau erhalten Sie, indem Sie alle drei Regler (RGB) im ersten Drittel des Feldes anordnen. Klicken Sie abschließend auf das Schachbrettsymbol (rechts von der »CHECKERED«-Anzeige). Mit »OK« wird der Requester geschlossen.

Da aus dem von uns festzulegenden Blickwinkel kein Himmel sichtbar sein wird, rufen wir mit »WORLD - SKY« den zuständigen Requester auf und klicken so lange in das Innere des »SKY COLOR«-Fensters, bis »NONE« angezeigt wird. Mit »OK« verzweigen wir zurück ins »Tri-View«.

Im nächsten Schritt werden wir das Quadrat so auf der Grundlinie (dem Horizont) plazieren, daß es nicht im Boden verschwindet. Zur Durchführung verwenden wir wieder den sog. Linealtrick, der sich in der-

artigen Situationen besonders bewährt. Vergrößern wir aber zunächst das »Tri-View«. Dreimal das »EXPAND TRI-VIEW«-Gadget anklicken.

Der Linealtrick

Wo liegt das Problem? Ganz einfach: Sobald Sie eine Objektverschiebung mit Hilfe des »GRABBER«-Werkzeuges ausführen, kann es passieren, daß sich auch die Lage des Objektes auf der x-Achse ändert, was ja in unserem Fall nicht erwünscht ist. Selbst wenn Sie noch so genau arbeiten, kann die beschriebene Situation eintreten. Ideal wäre es, wenn ein Werkzeug zur Verfügung stehen würde, das es erlaubt, je nach Bedarf, ein Objekt exakt in der Vertikalen (y-Achse) oder Horizontalen (x-Achse) zu verschieben. Leider kennt SCULPT kein entsprechendes Tool für derartige Situationen, so daß wir zu einem Trick greifen müssen. Wenn alle Punkte eines Objekts selektiert sind, gestaltet sich der Vorgang problemlos.

Das Fadenkreuz wird (in unserem Beispiel) an die äußerste linke oder rechte Kante des »Tri-View« gesetzt und durch einen Mausklick fixiert. Dann aktivieren Sie das »GRABBER«-Gadget, versetzen den Cursor direkt in den linken Tri-View-Rahmen und klicken die linke Maustaste an.

Befindet sich das Fadenkreuz oberhalb der Objektgrundlinie, wird sich das Modell hochbewegen ohne von seiner horizontalen Position abzuweichen.

Wenn Sie nun den Cursor innerhalb der Leiste nach oben oder unten bewegen - je nachdem, wie es die Aufgabenstellung erfordert - und nach jeder Lageänderung die linke Maustaste anklicken, kann das Objekt vertikal verschoben und auf der Grundlinie positioniert werden.

Zu beachten ist:

Wenn nicht alle Punkte des Objektes selektiert wurden, wird die Lageänderung nur für den oder die selektierten Eckpunkte ausgeführt.

Sobald sich der Cursor im Fensterahmen befindet und entsprechend der oben beschriebenen Methode in Aktion tritt, müssen Sie darauf achten, daß Sie dabei nicht versehentlich eines der im Rahmen untergebrachten Gadgets aktivieren!

Das Lineal kann nur dann in Funktion treten, wenn sich das Fadenkreuz entweder im linken oder rechten Fensterrand (vertikale Objektbewegung) oder im unteren »Tri-View«-Rand befindet (horizontale Verschiebung). Eine Plazierung in der oberen Leiste löst keine Aktion aus.

Alles klar? Okay, dann stellen Sie das Fadenkreuz so dicht als möglich an den linken oder rechten Rand des »NORTH«-Fensters und aktivieren das »GRABBER«-Gadget.

Klicken Sie dann in das Innere des linken oder rechten Fensterrandes, so daß sich das Viereck um etwa einen Rasterpunkt über den Horizont hebt (ca. 1 mm). Schalten Sie den »GRABBER« aus und klicken Sie die rechte Maustaste an.

Aktivieren Sie das »EXPAND TRI-VIEW«-Gadget und klicken Sie es siebenmal an. Das schafft Platz für die folgende Festlegung des Beobachterstandortes (»OBSERVER-TARGET«) und für die Plazierung der Lampen.

Verschieben Sie den Cursor im »NORTH«-Fenster an die Koordinaten

East : 9500
North : -22000
Up : 16300

und überprüfen Sie diese Position in einem der anderen Fenster. Abweichungen von ± 10 Prozent spielen keine Rolle.

Befehlen Sie »OBSERVER - LOCATION«

Damit bei der Positionierung der Lampen genauer gearbeitet werden kann, klicken Sie viermal das »CONTRACT TRI-VIEW«-Gadget an.

Setzen Sie die Farbe der ersten Lampe auf Violett:

```
SCULPT 3D: EDIT --LAMPS -  
            - COLOR  
SA4D      : EDIT --MODIFY -  
            - LAMPS
```

Die Regler »S« und »V« werden dazu an den rechten Fensterrand geschoben und der »H«-Regler im letzten Viertel der Anzeige plaziert. Eine

Kontrolle ist über das Farbanzeige-
fenster möglich. Abschließend »OK«
anklicken.

Im »NORTH«-Fenster wird der
Cursor auf die Position

UP : 2600

gesetzt. Im »DOWN«-Fenster plazie-
ren Sie den Cursor auf den obersten
Punkt des Quadrates.

Jetzt kann der Standort der ersten
Lampe fixiert werden:

```
SCULPT 3D: EDIT -- LAMPS
             -- CREATE
SA4D      : EDIT -- ADD
             -- LAMP
```

Die zweite Lampe wird im
»DOWN«-Fenster auf den untersten
Punkt des Quadrates gestellt, wobei
Sie darauf achten sollten, daß sich
die Höhe (»UP«) im »NORTH«-
Fenster nicht verändert. Ist das si-
chergestellt, befehlen Sie

```
SCULPT 3D: EDIT -- LAMPS
             -- CREATE
SA4D      : EDIT -- ADD
             -- LAMP
```

Legen Sie die Lampenfarbe auf
Violett fest:

```
SCULPT 3D: EDIT -- LAMPS
             -- COLOR
SA4D      : EDIT -- MODIFY
             -- LAMPS
```

»R«- und »B«-Regler kommen an
den äußersten rechten Rand, der
»G«-Regler wird nach links gezogen:

```
SA4D      : SET anklik-
             ken, dann OK.
SCULPT 3D: OK anklicken,
             dann
             EDIT -- MODIFY
             --INDICATED LAMP COLOR
```

Die Farbe der dritten Lampe wird
auf Gelb festgelegt (»EDIT - LAMPS
- COLOR«, bzw. »EDIT - MODIFY -
LAMPS«). Den »R«- und »G«-Regler
ziehen Sie voll nach rechts, der »B«-
Regler verbleibt am linken Fenster-
rand. Mit »SET« (SA4D) bzw. »EDIT -
MODIFY - INDICATED LAMP CO-
LOR« (SCULPT 3D) wird die neue
Farbe übergeben.

Lampe 3 wird an den nachstehen-
den Koordinaten untergebracht
(»DOWN«-Fenster):

```
East      : -3300
North     : 11500
und mit EDIT -- LAMPS -- CREATE
           (SCULPT 3D)
```

```
bzw.      EDIT -- ADD
           -- LAMP (SA4D)
```

an dieser Position aufgehängt.

Die vierte Lampe wird türkis einge-
färbt. Rufen Sie den Requester auf
und stellen Sie den »G«- und »B«-
Regler nach rechts und den »R«-
Regler ganz nach links. Die Lampe
wird dann an den folgenden Koordi-
naten plaziert:

```
East      : 11200
North     : 11500
```

und wird dort fixiert (»EDIT - LAMPS
CREATE«, bzw. »EDIT - ADD -
LAMP«).

Auch die Farbe der fünften Lampe
soll Türkis sein. Rufen Sie den
»LAMP COLOR-Requester« auf.

```
SCULPT 3D: EDIT -- LAMPS --
           COLOR
SA4D      : EDIT -- MODIFY --
           LAMPS
```

Schieben Sie den »R«-Regler an
den linken und die Regler »G« und
»B« an den rechten Rand des Regler-
feldes.

```
SCULPT 3D: OK anklicken,
             dann
             EDIT -- MODIFY --
             INDICATED LAMP COLOR
SA4D      : SET anklicken,
             dann OK.
```

Lampe 5 wird an den nachstehen-
den Koordinaten untergebracht
(»DOWN«-Fenster):

```
East      : 11200
North     : -3500
und mit
EDIT -- LAMPS --
CREATE (SCULPT 3D)
bzw.: EDIT -- ADD
       -- LAMP (SA4D)
```

an dieser Position aufgehängt.

Die sechste Lampe kommt an die
Koordinaten

```
East      : -3400
North     : -3500
```

und wird dort fixiert (»EDIT - LAMPS
CREATE«, bzw. »EDIT - ADD -
LAMP«).

Die Lampenfarbe wird auf Gelb
eingestellt (Rot- und Grün-Regler
rechts, Blau-Regler links) und an das
Programm übergeben.

Damit ein möglichst realistischer
Leuchteffekt zustande kommt, muß
von automatischer auf manuelle Be-
lichtung umgeschaltet werden:

```
OBSERVER -- EXPOSURE -- MANUAL
```

Die Frage nach der neuen Pro-
zentzahl (»Percentage of standard
exposure?«) beantworten Sie mit
300. Dadurch wird das gesamte Bild
aufgehellt. Wenn Sie »OK« anklicken
oder die <RETURN>-Taste drük-
ken, schließt sich der Requester.

Im nächsten Schritt wird das »Di-
ttering« abgeschaltet. Mit dieser
Antialiasing-Option kann normaler-
weise ein optimaler Übergang an
den Farbkanten eingestellt werden.
Da wir aber in unserem Fall harte Ab-
stufungen benötigen, wird der Wert
von 100 Prozent auf 0 Prozent korri-
giert:

```
OBSERVER -- ANTI ALIASING --
DITHERING
```

Tippen Sie im Verlauf-Requester
(»Dithering Percentage«) eine 0 ein
und schließen Sie den Requester
durch Drücken der <RE-
TURN>-Taste.

Eine weichere Glättung der Facet-
ten erreicht man, wenn man den
Antialiasing-Schalter auf »GOOD«
(Gut) umlegt. Dieser Wert genügt
vollauf, um das Bild zu testen:

```
OBSERVER -- ANTI--ALIASING --
GOOD
```

Sehen wir uns das Ergebnis der
Arbeit mal an. Schalten Sie in den
»SNAPSHOT«-Modus und wählen
Sie als Bildformat »SMALL«:

```
OBSERVER -- MODE -- SNAPSHOT
OBSERVER -- IMAGE SIZE --SMALL
OBSERVER -- START oder F8
```

Wenn der Test erfolgreich verlau-
fen ist, kann zur Endberechnung
übergegangen werden. Den
»SNAPSHOT«-Modus können Sie
weiterverwenden. Die restlichen Da-
ten sollten Sie, wie folgt, ändern:

```
OBSERVER -- IMAGE SIZE --JUMBO
OBSERVER -- ANTI --
ALIASING -- BEST
OBSERVER -- MODE --
INTERLACE
```

```
OBSERVER --
START oder F8 drücken
```

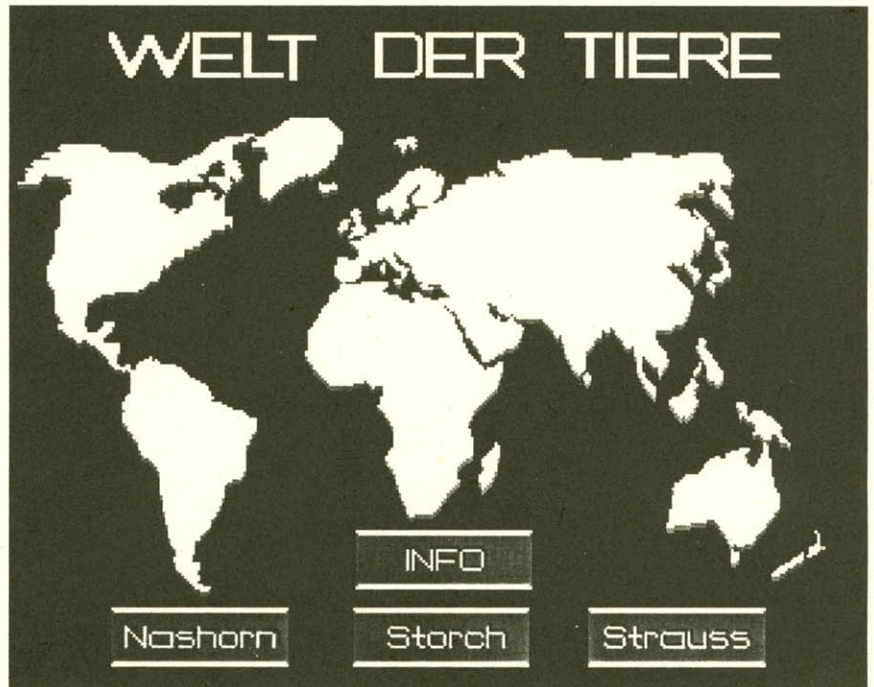
Die Entwicklungszeit beläuft sich
auf ca. 2:20 Stunden. Während die-
ser Zwangspause können Sie be-
reits neue Objekte und Szenen ent-
werfen. Dank Multitasking! *pe*

Interaktive Programmabläufe zu realisieren, gehört zu den Stärken von Deluxe Video III. Am Beispiel eines Bilderlexikons soll gezeigt werden, wie man derartige Aufgaben lösen kann.

Datentransfer

Deluxe Paint

ruft Deluxe Video...



von Walter Friedhuber

Um Deluxe Video III (in der Folge mit DVideo bezeichnet) zur Gestaltung interaktiver (mausgesteuerter) Abläufe oder zum Aufbau vertonter Videoclips einsetzen zu können, bedarf es einiger Vorarbeiten, die man besser mit Deluxe Paint III (kurz DPaint genannt) erledigt. Lesen Sie, welcher Teil des künftigen Videos wo herzustellen ist, wie Hindernisse beim Transfer der Daten überwunden werden können und sich die jeweiligen Stärken beider Programme kombinieren lassen.

REALISIERUNG EINER INTERAKTIVEN PROGRAMMSTEUERUNG

Wie Bild 1 zeigt, werden wir eine mausgesteuerte Anwendung realisieren, mit deren Hilfe es auf Knopfdruck möglich ist, die einzelnen Teile des Lexikons anzuwählen, deren Inhalte zu betrachten und eine Rückkehr ins Startbild (das Hauptmenü) abzuwickeln.

Zur Verwirklichung des Projekts, das auch auf 1-MByte-Systemen

nachgebaut werden kann, sind zahlreiche Grafiken, animierte Pinsel und Brushes notwendig (finden Sie alle auf unserer Programmservice-Diskette).

Bevor wir näher darauf eingehen, noch kurz ein paar Worte über den Handlungsablauf:

DIE PROGRAMMLOGIK:

Sobald ein Benutzer den Knopf »INFO« anklickt, verzweigt das Programm in eine Szene, in der Tierarten angezeigt und mit Texten unterteilt wurden. Bild 2 zeigt dies in schematisierter Form, wobei die Textposition durch Linien angedeutet wird.

EIN FALL FÜR ZWEI...

Jede dieser Info-Tafeln wird mit einem frei definierbaren Überblendeffekt angezeigt, bleibt für wenige Sekunden sichtbar, bis das Folgebild aufgerufen wird. Beim letzten Bild erhält der Benutzer die Möglichkeit, ins Hauptmenü (das Startbild) zurückzukehren.

Dort stehen drei weitere Icons, die jeweils unterschiedlich visuell aufgebaute Szenen starten, deren »Programmierung« jedoch weitgehend identisch ist. Zu Beginn der Szene »Nashorn«, »Storch« und »Strauss« wird mit Hilfe einer Animation die

Weltkarte perspektivisch ausgeblendet. Die entsprechende Manipulation wurde mit DPaint realisiert. Nachdem die Karte im Nichts verschwunden ist, kann DVideo das Einfädeln des ersten Hintergrundbildes übernehmen. Über dessen Oberfläche wird anschließend ein animierter Pinsel (AnimBrush) »gezogen«. Ist dies geschehen, erfolgt die Ausgabe eines Icons, dessen Anklicken mit der linken Maustaste eine Rückkehr zum Startbild provoziert, von wo aus die beiden restlichen Szenen aufgerufen werden können.

Innerhalb der insgesamt fünf Szenen-Skripten (»Start« —> Menübild, »Slideshow« —> Grafik und Texte, »Nashorn«, »Storch« und »Strauss« —> Animationen) widmen wir den Möglichkeiten der »STEUERN«-Spur (Control) unsere ganze Aufmerksamkeit. Effekte wie »WARTEN« (Wait), »GEH ZU« (GoTo) und »MARKE« (Label) werden detailliert behandelt.

Bild 1. Das Titelbild können Sie frei gestalten

VORBEREITENDE TÄTIGKEITEN:

Alle aufgelisteten Video-Bestandteile werden in DPaint angefertigt:

GRAFIKEN: Insgesamt benötigen Sie vier Hintergrundbilder, die im Lores-Modus von DPaint (32 Farben) erstellt werden sollten. Sie müssen dazu nicht erst unsere Vorlagen abmalen, sondern können ohne weiteres beliebige eigene Motive verwenden:

Grafik 1: Startbild (Menütafel)

Grafik 2 bis 4: Hintergrund für Animationen und Infotafeln

PINSEL:

Fünf rechteckige Pinsel zur Darstellung der Menü-Icons, die mit den Texten »ZURÜCK«, »INFO«, »NASHORN«, »STORCH« und »STRAUSS« versehen sein sollten, sowie einen Pinsel, der zur Beteilung der Infothek eingesetzt wird (WELT DER TIERE), sollten angelegt werden. Als Dateinamen verwenden Sie die Bezeichnungen »Titel«, »Knopf-Zurück«, »Knopf-Info«, »Knopf-Nashorn« usw.

Die Grafiken Nummer 2 bis 4 (Animationshintergründe) werden zusätzlich als Pinsel ausgeschnitten und abgespeichert. Wobei nur etwa zwei Drittel der Gesamtfläche benötigt werden, da das fehlende untere Drittel durch den Hires-Text ergänzt wird. Im Beispiel haben wir dafür die Dateinamen »Kombi-Nashorn«, »Kombi-Storch« und »Kombi-Strauss« vergeben. Innerhalb dieser Pinsel sollten Standbilder der im Text erläuterten Figuren einkopiert werden (z.B. das erste Bild des jeweils einzuspielenden AnimBrushes).

Den Hires-Text fertigen Sie im DPaint-Format 640 x 512 an. Er darf, wie schon erwähnt, nicht mehr als 30 Prozent des Bildschirms einnehmen. Zweckmäßigerweise wird im Bildschirmformat-Menü des Malprogramms die Farbzahl auf zwei Töpfe reduziert! Damit steht zwar nur eine Schriftfarbe zur Verfügung, mindert aber den Speicherbedarf!

Alle drei Textteile müssen separat als Pinsel aufgenommen und im Brush-Verzeichnis der Datendiskette abgelegt werden. Verwenden Sie da-

zu die Namen »Text-Nashorn«, »Text-Strauss« und »Text-Storch«.

ANIMBRUSHES:

Zur Verwirklichung der 1-MByte-Version genügt es völlig, wenn Sie nur einen animierten Pinsel verwenden. Vergessen Sie aber nicht, dessen Farben an den Hintergrund anzupassen. Laden Sie dazu die Hintergrundillustration, dann den animierten Pinsel und befehlen Sie anschließend »Pinsel - Farbwechsel - Anpassen«. Zuletzt speichern Sie das Bild im Unterverzeichnis »AnimBrush« der Datendiskette ab.

ANIMATION:

Bei der mit DPaint erstellten Animation sollte es sich ebenfalls um eine Lores-Datei handeln. Von der Verwendung des Overscan-Formats sollten Benutzer, die lediglich 1 MByte zur Verfügung haben, tunlichst Abstand nehmen.

Damit hätten wir alle Elemente unseres Videos. Bevor wir endlich starten können, sollten Sie alle Daten auf einer speziell vorbereiteten Diskette ablegen.

deoclips herangezogenen Daten in einem speziellen Script.

Auf einer formatierten Diskette bringen Sie folgende Unterverzeichnisse an:

```

Anims -- zur Ablage von
    >> Anim Opt-5 <<-Dateien
    (DPaint-Filme)
AnimBrushes -- animierte Pinsel
    aus DPaint III
Pictures -- Grafiken in
    beliebigem Format
    (HAM, Lores...)
Brushes -- selbst definierte
    Pinsel (DPaint III)
    
```

AUFBAU DER FALLSTUDIE:

Das Hauptprogramm »DVMaker« sollte sich in Laufwerk df0: befinden. Ihre Datendiskette, die bereits alle im Film benötigten Einzeldateien enthält, wird in das externe Laufwerk df1: eingelegt.

Starten Sie »DVMaker«. Nachdem DVideo geladen wurde, landen Sie im »Video Script«, der obersten Hierarchiestufe des Programms. Von hier aus werden alle getätigten Aktionen

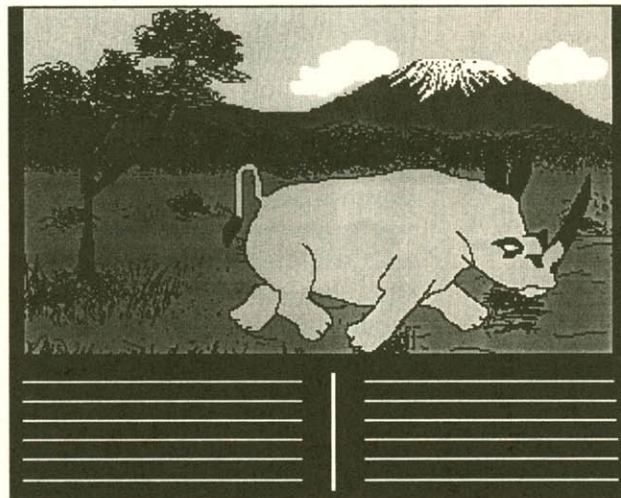


Bild 2. Im Info-Bild »Nashorn« muß nur noch der Hires-Text eingeblendet werden

EINRICHTEN EINER DATENDISKETTE:

Beim Arbeiten mit DVideo spielen Datendisketten eine große Rolle. Diese sollten nicht nur alle vom Programm angesprochenen Unterverzeichnisse, sondern auch das komplette, für die jeweilige Aktion benötigte Grafik-, Musik- und Animationsmaterial enthalten. DVideo sammelt nämlich alle zum Aufbau eines Vi-

zentral gesteuert. Mit den dort sichtbaren Spuren und aufgesetzten Szene-Effekten kann man sich einen Überblick verschaffen, aus wie vielen Teilszenen ein Film besteht und zu welchem Zeitpunkt diese abgespielt werden.

Ein »Video Script« besteht immer aus minimal zwei Spuren: Der »SICHT«- und der »VIDEO«-Spur. Während von der »SICHT«-Spur Daten über das Gesamtvideo verwaltet werden (Bildschirmformat, Seitengröße, Overscan, Interlace-Darstellung...), dient die darunter liegende

»VIDEO«-Spur zur Aufnahme von »Szene«-Effekten.

Klicken Sie den »Szene«-Effekt auf der »Video«-Spur zweimal an.

DVideo verzweigt in das leere Szene-Skript und erwartet nun, daß Sie hier die Bauteile Ihres Films anordnen.

Rufen Sie mit gedrückter, rechter Maustaste das am oberen Bildschirmrand versteckte »Szene«-Menü auf und aktivieren Sie dort den Menüpunkt »Verändern«.

Übrigens: Derartige Menüaufrufe (Befehlsfolgen) werden wir in der Folge jeweils in Kurzform beschreiben, was für unser Beispiel dann so aussieht: »Szene — Verändern«.

DVideo blendet daraufhin den »Szene«-Requester ein. Löschen Sie den vorgegebenen Namen und tragen Sie statt dessen »START« ein. Nach Schließen des Fensters mit »Ok« ist die Umbenennung des Szenen-Skripts abgeschlossen.

Schalten Sie mit »Optionen - Experte« auf den erweiterten Programm-Modus um. Dadurch werden alle im Beispiel verwendeten Effekte zur Verfügung gestellt.

Bauen wir zunächst das Startbild auf (Bild 3). Ziehen Sie eine BILD-Spur ins leere Skriptfeld und laden Sie aus dem »Bild«-Requester die vorgesehene Illustration (Menütafel), der wir im aktuellen Beispiel den Namen »Weltkarte« gegeben haben. Übernehmen Sie diese mit »Wählen« in Ihr Video.

Setzen Sie einen »POSITION«-Effekt auf und klicken Sie im »Was und Wo«-Requester sofort den »Ok«-Knopf an. Der Zeitzeiger des Effekts wird auf 0:00.00 gestellt. Dann holen Sie einen »SCHAU«-Effekt und stellen diesen auf 00.00.10.

Innerhalb dieser Grafiken müssen nun, wie es Bild 1 zeigt, der Titel und die vier vorbereiteten Icons (Knöpfe) »INFO«, »NASHORN«, »STORCH« und »STRAUSS« plaziert werden. Setzen Sie dazu eine »PINSEL«-Spur in Ihr Script ein und laden Sie die Datei »Knopf-Info«: Wieder wird ein »POSITION«-Effekt dazu verwendet, den Standort des Pinsels festzulegen. Nachdem Sie den Standort bestimmt und das Menü mit »Ok« verlassen haben, verstellen Sie den Zeitzeiger auf 00:00.00. Ein »SCHAU«-Effekt, auf 00:00.10 ange-

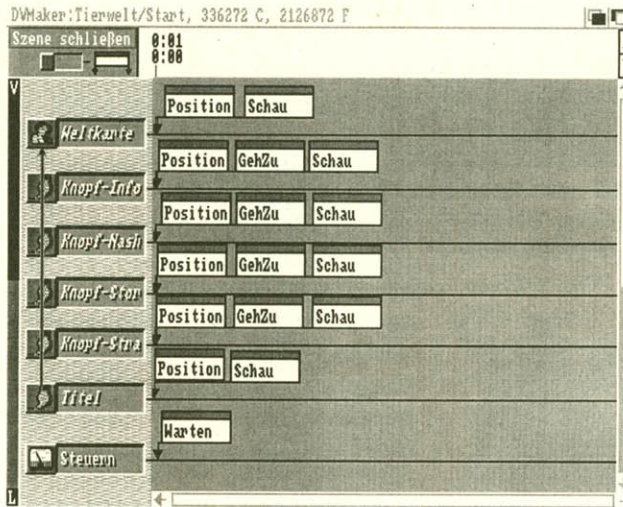


Bild 3. Die unten angelegte »STEUERN«-Spur regelt den Programmablauf

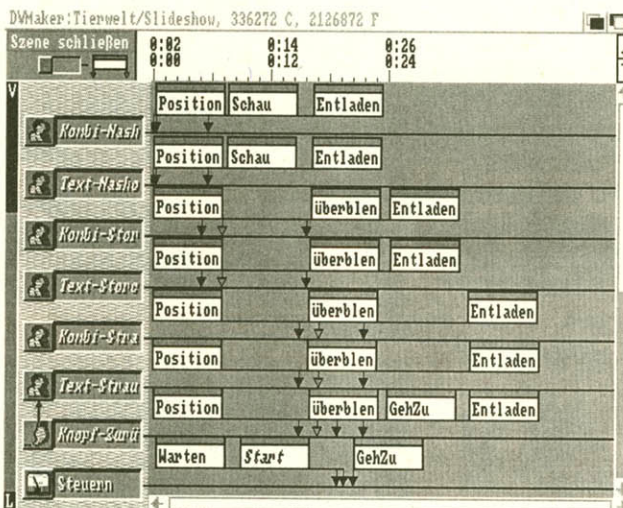


Bild 4. Auch umfangreiche Skripts sind für Deluxe Video III kein Problem

setzt, bringt den Knopf zur Anzeige. Wie Sie sich erinnern, soll dieser Pinsel beim Anklicken mit der linken Maustaste die Verzweigung in ein anderes, noch zu erstellendes Szenen-Skript (namens »SLIDESHOW«) auslösen. Um dies realisieren zu können und die Anklickfläche auf exakt diesen Pinsel zu beschränken, wird ein »GEHZU«-Effekt auf die »Knopf-Info«-Spur aufgesetzt. Im gleichnamigen Requester wird der künftige Szenenname des »Sprungziels«: »SLIDESHOW« eingetragen. Der Zeitzeiger muß auf die Marke 00:00.05 weisen.

Das Feld »Video« (im »Gehe zu«-Requester) sollte nicht aktiviert werden. Nur im ausgeschalteten Zustand wird auf das im »NAME«-Feld eingegebene Sprungziel verzweigt. Kombiniert man einen »GEHZU« mit

einem »WARTEN«-Effekt, ermöglicht DVideo Verzweigungen nicht nur innerhalb eines Skripts, sondern auch den Aufruf externer Szenen, die in anderen Skripten abgelegt wurden! Dazu ist allerdings eine spezielle »STEUERN«-Spur nötig, die wir gleich anlegen werden.

Wie aus Bild 3 hervorgeht, müssen nun die drei Icons (»Knopf-Nashorn«, »Knopf-Storch« und »Knopf-Strauss«) auf identische Weise angebracht werden. Einzig und allein die beim »GEHZU«-Effekt angefragten »Sprungadressen«, die Szenennamen, müssen entsprechend geändert werden (»NASHORN«, »STORCH«, »STRAUSS«). Zusammenfassend hier noch mal die Positionen der einzelnen Zeitzeiger:

POSITION -Effekt: 00:00.00
 GEHZU -Effekt: 00:00.05
 SCHAU -Effekt: 00:00.10

Noch hat es keinerlei Sinn, das »START«-Skript abzuspielen. Eine »STEUERN«-Spur muß her (Bild 3)

und darauf ein »WARTEN«-Effekt abgelegt werden, der auf 00:00.20 zu positionieren ist. Der »WARTEN«-Effekt enthält zwei Möglichkeiten: Wird der zugehörige Requester ohne Inhaltsänderung mit »Ok« geschlossen, pausiert das Programm mit seinem Ablauf solange. Bis ein entsprechendes, durch einen »GEHZU«-Effekt markiertes Icon angeklickt wird. Klickt man aber das quadratische Feld »Wartezeit« an, kommt der - im daneben angeordneten Feld - ausgewiesene Sekundenwert (Voreinstellung 60) zum Tragen.

Für unser Beispiel darf das »Wartezeit«-Feld nicht aktiviert werden, da die Fortsetzung des Programmablaufs mausgesteuert erfolgen soll!

Neben dem bereits angesprochenen »GEHZU«-Effekt, der Sie zum Anfang einer anderen Szene oder zu einer mit »MARKE« gekennzeichneten Spur führt (erklären wir später), läßt sich die Steuerung des Geschehens auch an einen Joystick übergeben oder via Tastatur (»KEY TO«-Effekt) abhandeln.

Damit wäre die Definition des ersten Skripts »START« abgeschlossen und Sie können mit »Projekt - Szene spielen« einen ersten Probelauf durchführen. Ein Anklicken der Icon-Felder hat zu diesem Zeitpunkt allerdings noch keinen Sinn, da die entsprechenden »Einsprungadressen« (Szenen-Skripten) erst noch definiert werden müssen.

Drücken Sie deshalb — zur Rückkehr ins Skript — die ESC-Taste.

Klicken Sie »Szene schließen« an und legen Sie, unmittelbar neben der bereits vorhandenen »START«-Szene einen neuen »Szene«-Effekt an, den Sie in »SLIDESHOW« umbenennen (»Szene — Verändern« befehlen).

Die »SLIDESHOW«-Szenerie (siehe Bild 4) gestaltet sich bereits sehr umfangreich. Beginnen Sie damit, eine »BILD«-Spur einzusetzen. Aus dem Inhaltsverzeichnis Ihrer Datendiskette muß nun die im Brushes-

Unterverzeichnis (!) abgelegte Pinseldatei »Kombi-Nashorn« ausgewählt werden. »Kombi-Nashorn« ist das verkleinerte, als Pinsel gespeicherte Lores-Hintergrundbild, das anschließend mit dem Hires-Textpinsel kombiniert werden soll (Bild 2). Ein »POSITION«-Effekt (00:00.00) legt die Lage des Pinsels auf der Sichtspur fest, die mit der Option »Vorschau Wo« des »Was und Wo«-Requesters zu bestimmen ist. Der

Effektehascherei

Grafikpinsel wird an den oberen Bildschirmrand geschoben, zentriert und mit »Ok« abgestempelt. Ein »SCHAU«-Effekt (00:00.15) bringt das Bild auf den Monitor. Bei dieser Spur wird abermals eine »BILD«-Spur angelegt. Dann diese mit der Hires-Datei »Text-Nashorn« gefüllt, die Sie aus dem »Bild«-Requester (Unterverzeichnis Brush) selektieren. Wenn Sie jetzt aber mit »Vorschau« den Textpinsel betrachten, sehen Sie, daß dieser in Lores angezeigt wird und somit unbrauchbar ist. Keine Panik!

Ändern wir das: Klicken Sie »Info« an und sehen Sie sich den »Raster Info«-Requester genauer an. In seiner unteren Hälfte enthält er drei »Knöpfe«, die mit »Modus«, »HIRES« und »LACE« gekennzeichnet sind. Wenn Sie der Reihe nach alle drei Icons selektieren, haben Sie bereits alles getan. Da DVideo sich nicht darum kümmert, welches Bildschirmformat ein Pinsel hat, läßt sich mit diesen Knöpfen so ziemlich alles realisieren. Schließen Sie den Requester mit »Ok« und versuchen Sie nochmals den »Vorwahl«-Schalter. Jetzt müßte alles in Ordnung sein. Befehlen Sie abschließend »Wählen«. Die schon obligatorischen Effekte »POSITION« (00:00.00) und »SCHAU« (00:00.15) dürfen natürlich nicht fehlen, wobei im »Was und Wo«-Requester des »POSITION«-Effekts der Textpinsel an den unteren Bildschirmrand gezogen wird.

Ein »ÜBERBLENDUNGEIN«-Effekt (Startzeiger auf 0:05.00 und Endezeiger auf 0:07.05) ersetzt den bisher benutzten »SCHAU«-Effekt. Aus dem »Überblendeffekt«-Requester kann das gewünschte Einblendmuster ausgewählt und mit »Ok« in die Szene übernommen werden.

Da wir aus Speicherplatzgründen die »BILD«-Spuren zum Positionieren der Textpinsel mißbraucht haben, kann die Lage der beiden Bildteile nur durch Abspielen des Videos mit »Projekt — Szene spielen« überprüft werden. Positionskorrekturen lassen sich anschließend durch zweimaliges Anklicken der »POSITION«-Effekte durchführen.

Nach demselben Schema vollzieht sich auch die Belegung der »BILD«-Spuren »Kombi-Storch«, »Text-Storch« und »Kombi-Strauss«, »Text-Strauss«. Lediglich die Einblendzeiten (ÜBERBLENDUNGEIN-Effekte) und POSITION-Markierungen müssen der veränderten Situation angepaßt werden. Schließlich soll eine gewisse Zeit verstreichen, bevor ein Bild vom anderen abgelöst wird. Hier die entsprechenden Zeiten:

Spur: "Kombi-Storch"

ÜBERBLENDUNGEIN Start: 0:05.00

Ende: 0:07.05

POSITION Start: 0:05.00

Spur: "Text-Storch"

ÜBERBLENDUNGEIN Start: 0:05.00

Ende: 0:07.05

POSITION Start: 0:05.00

Spur: "Kombi-Strauss"

ÜBERBLENDUNGEIN Start: 0:15.00

Ende: 0:17.00

POSITION Start: 0:15.00

Spur: "Text-Strauss"

ÜBERBLENDUNGEIN Start: 0:15.00

Ende: 0:17.00

POSITION Start: 0:15.00

Wie Sie aus der Aufgabenstellung wissen, soll nach Einblendung der beiden letzten Pinselkombinationen ein benutzergesteuerter Rücksprung in die Szene »START« und dort zur Bildspur »Weltkarte« (dem Menübild) vollzogen werden. Dies läßt sich realisieren, in dem eine »PINSEL«-Spur unterhalb der zuletzt angebrachten »BILD«-Spur (»Text-Strauss«) plaziert wird. Dann die Datei »Knopf-Zurück« eingelesen und die Spur - neben dem »POSITION«- und »ÜBERBLENDUNGEIN«-Effekt, mit einem »GEHZU«-Effekt versehen wird. Dieser »GEHZU«-Effekt erhält den Namen START (»Gehe zu«-Requester, Feld »Name«). Bevor wir mit

der Erklärung fortfahren, hier noch schnell die Positionen:

PINSEL-Spur: Dateiname

"Knopf-Zurück"

POSITION Start: 0:15.00

ÜBERBLENDEIN Start: 0:15.00

Ende: 0:17.00

GEHZU Start: 0:18.45

Der Effekt »GEHZU« »Start« darf hier nicht dazu benutzt werden, einen automatisch ausgeführten Sprung ins gleichnamige Szenen-Skript zu provozieren, da dies – laut Aufgabenstellung – nur vom Benutzer veranlaßt werden soll.

Deshalb werden wir die Ausführung der »GEHZU«-Anweisung mit Hilfe einer »MARKEM«, die wir auf der »STEUERN«-Spur anbringen, umleiten und durch Setzen eines »WARTEN«-Effektes solange außer Kraft setzen, bis der Benutzer das »ZURÜCK«-Icon anklickt. Ist dies geschehen, wird ein weiterer, auf der »STEUERN«-Spur angelegter »GEHZU«-Effekt, der den Namen des Sprungziels (»Weltkarte«) aufweist, die sich im »START«-Skript befindet, zum Aufruf des Hauptmenüs (Bild 1) verwendet.

Mit Hilfe einer »MARKE«, die nichts als ein namentlich benanntes Ziel darstellt, könnten Sie kreuz und quer innerhalb eines Szenen-Skripts umherspringen, nicht aber in ein anderes Skript wechseln. Dazu muß die »GEHZU«-Option zum Einsatz kommen.

Definieren wir abschließend die »STEUERN«-Spur. Setzen Sie zuerst einen »WARTEN«, dann einen »MARKE«- und zuletzt einen »GEHZU«-Effekt auf diese Spur. Kümmern Sie sich noch nicht um die Zeitzeiger, dazu kommen wir gleich.

Der »MARKEM«-Effekt wird mit dem Namen »START«, der »GEHZU«-Effekt »WELTKARTE« belegt. So, aber nun zum Setzen der Zeitmarkierungen, die in diesem Fall äußerst kritisch sind.

Der »WARTEN«-Effekt (»STEUERN«-Spur) muß exakt mit der Startzeit des »GEHZU«-Effekts auf der

»PINSEL«-Spur (»Knopf-Zurück«) synchronisiert werden (0:18.45). Aber das ist nicht das einzige, was zu beachten ist: Die »MARKE« (»Start«) muß zeitlich zwischen dem »WARTEN«- und »GEHZU«-Effekt (0:20.30) angeordnet werden (z.B. 0:19.30). Beachten Sie diese »Regeln«, kann nichts mehr schiefgehen!

Bevor wir uns den ersten Teil unseres künftigen Videos ansehen, muß man sich allerdings überlegen, ob der vorhandene Speicher dazu überhaupt ausreicht. Leider: 1 MByte reicht nicht. Treffen wir also einige Vorkehrungen, die Platz schaffen. Der im »Experte«-Modus zur Verfügung gestellte Effekt »ENTLADEN« ist genau das richtige Mittel. »ENTLADEN« wirft zum definierten Zeitpunkt alles aus dem Pool. Ergebnis: Ein leerer Bildschirm und aufgefrischte Speichervorräte. Der Haken an der Sache ist, daß der Zeitpunkt, zu dem dies erfolgen soll, genau bedacht werden muß. Sinnvoll ist es, den Speicher dann zu entleeren, wenn kurz zuvor das Folgebild (-animation) geladen wurde. Würden Sie den Effekt zu früh ansetzen, blicken Sie im wahrsten Sinne des Wortes in die Röhre – der Bildschirm verfärbt sich schwarz.

Und hier die Zeitpunkte für sämtliche »ENTLADEN«-Effekte der aktuellen Szene:

Spur BILD "Kombi-Nashorn"

Start: 0:05.30

Spur BILD "Text-Nashorn"

Start: 0:05.30

Spur BILD "Kombi-Storch"

Start: 0:15.35

Spur BILD "Text-Storch"

Start: 0:15.35

Spur BILD "Kombi-Strauss"

Start: 0:21.30

Spur BILD "Text-Strauss"

Start: 0:21.30

Spur PINSEL "Knopf-Zurück"

Start: 0:21.30

Sehen wir uns jetzt an, ob die Sache funktioniert: Wählen Sie »Projekt - Video spielen«.

Anmerkung: Auf keinen Fall dürfen Sie »Projekt - SZENE spielen« zur Kontrolle benutzen, da DVideo sonst die Ausgangssituation, das Sprungziel »Weltkarte« in der Szene »START«, nicht mehr herstellen könnte!

Nachdem Sie das mit »INFO« bezeichnete Knöpfchen des Hauptmenüs mit der linken Maustaste angeklickt haben, verzweigt DVideo ins Szenen-Skript »SLIDESHOW«. Handelt dort der Reihe nach alle Spuren und Effekte ab, bis es – durch den »GEHZU«-Effekt (»Start«) – die »MARKE« »Start« auf der »STEUERN«-Spur erreicht hat. Der zeitlich zum »GEHZU« synchron angesetzte »WARTEN«-Effekt läßt einen Mauszeiger am Bildschirm erscheinen. Nun pausiert DVideo solange, bis Sie diesen Pfeil auf das Icon »Zurück« richten und die linke Maustaste anklicken. Sofort spult das Programm auch den letzten auf der »STEUERN«-Spur angebrachten Effekt »GEHZU« (»Weltkarte«) ab und verzweigt, wie dort befohlen, ins erste Bild (»Weltkarte«) der Szene »START« zurück. Da auch dort ein »WARTEN«-Effekt vorhanden ist, wird der Ablauf solange unterbrochen, bis eines der anderen Icons angeklickt wird.

Alles klar?

Sehen wir uns also an, wie die noch fehlenden Szenen »Nashorn«, »Storch« und »Strauss« aufgebaut werden können.

Verzweigen Sie (»Szene schließen«) ins Videoskript des Programms und legen Sie auf der dort plazierten »Video«-Spur einen weiteren »Szene«-Effekt ab. Geben Sie ihr den Namen »Nashorn« (Befehl: »Szene - Verändern«).

Landen Sie im leeren Szenen-Skript »NASHORN«. Bild 5 zeigt, was dort zu machen ist. Verfolgen wir kurz den Ablauf: Die »ANIM«-Spur (!), belegt mit der DPaint-Datei »Weltkarte-Aus«, steuert den Ablauf der Animation, die mit »SPIELANIM« sichtbar gemacht wird. »SPIELANIM« blendet den »Animation spielen«-Requester ein, in dem die Felder »ZYKLUS« und »WIEDERHO-

LEN 1« sowie »ZEITABSTAND 2« orangefarben markiert sein sollten (Voreinstellung). Ändern Sie hier nichts. Verlassen Sie den Requester mit »Ok«.

Versetzen Sie den Startzeiger auf 0:00.10. Der Endezeiger kann nicht verstellt werden (Fehler 304), da die Dauer der Wiederholung im »Animation spielen«-Fenster mit 1 festgelegt wurde. Aus diesem Grund ermittelt DVideo diese Zeit selbst.

Es wäre aber nun wenig sinnvoll, die voluminöse Animationsdatei weiterhin im Speicher zu belassen. Ein »ENTLADEN«-Effekt auf 0:10.20 wirft die Datei aus dem Speicher, bevor

den kann. Da derartige Gags natürlich den Speicher über Gebühr belasten, sorgen am Ende angebrachte »ENTLADEN«-Effekte, daß die Animationen sofort wieder eliminiert werden. So, nun aber zu den Details:

Plazieren Sie unterhalb der »ANIM«-Spur eine »BILD«-Spur und weisen Sie ihr die gewünschte Bild-datei (im Beispiel »Afrika-Steppe«) zu. Legen Sie darauf einen »LADEN«-Effekt ab (0:00.00) und leiten Sie mittels eines »ÜBERBLENDEIN«-Effektes die Anzeige der Hintergrundgrafik ein. Da dieser Zeitpunkt genau mit dem Ende des »SPIELANIM«-Effekts überein-

vom linken zum rechten Bildschirmrand erstreckt. Der »BEWEGZU«-Effekt sollte außerhalb des linken, sichtbaren Bildschirmbereichs beginnen und genauso aus dem Sichtbereich verschwinden (Option »Vorschau Wo« im »Was und Wo«-Requester). Der »ENTLADEN«-Effekt kommt auf 0:10.20 zum Stehen.

Um dem Benutzer die Möglichkeit zu verschaffen, am Ende des Vorgangs wieder in das Hauptmenü zu verzweigen, wird die mehrmals praktizierte »Pinsel-/Steuern-Spur«-Kombination eingesetzt.

Ziehen Sie dazu zuerst eine »PINSEL«-Spur und bestücken Sie diese mit der Datei »Knopf-Zurück«. Legen Sie den »POSITION«-Effekt auf 0:06.45 (entspricht dem Ende des »SPIELANIM«-Effekts) und suchen Sie sich im »Was und Wo«-Requester eine geeignete Position, an der Sie das Icon ablegen können (Option »Vorschau Wo«).

Auf derselben Zeitmarke (0:06.45) wird ein »SCHAU«-Effekt aufgesetzt und durch einen »GEHZU«-Effekt (0:08.25) ergänzt, der mit dem Namen »START« versehen wird.

Ein »ENTLADEN«-Effekt (0:10.20) sorgt zuletzt dafür, daß der belegte Speicher wieder freigegeben wird.

Abschließend bringen Sie eine »STEUERN«-Spur an, setzen den »WARTEN«-Effekt zeitgleich zum darüberliegenden »GEHZU«-Pendant (0:08.25), fügen die Ansprung »MARKE« namens »START« ein (0:09.00) und definieren mit einem »GEHZU« (Name »Weltkarte«, Start 0:10.00) die benötigte Rücksprungadresse im Szenen-Skript.

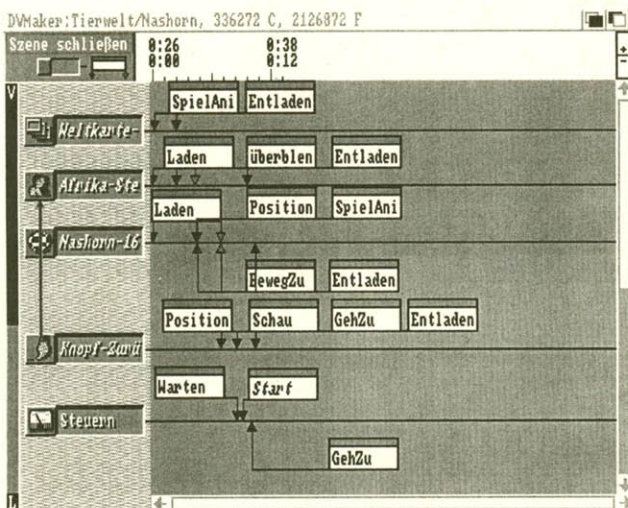
Jetzt kann mit »Projekt - Video spielen« der neue Teilzweig (NASHORN) getestet werden.

Mehr ist bei einem Speichervolumen von 1 MByte RAM allerdings nicht drin! Sichern Sie jetzt Ihr Video unter einem beliebigen Namen (»Projekt - Sichern als«).

Wer mehr Speicher zur Verfügung hat, findet auf der Programmservice-Diskette zu diesem Heft auch eine 3-MByte-Version dieser Präsentation.

Das war's. Nach diesem Schema lassen sich beliebig viele Szenen-Skripte definieren. In der Praxis werden Sie nur vom vorhandenen RAM-Vorrat gebremst. pe

Bild 5. Die Szene »Nashorn« sieht sehr verwirrend aus, ist aber bis ins letzte durchdacht



das Skript verlassen wird. Diese Zeitangabe muß keineswegs zwingend eingehalten werden, wichtig ist nur, daß sie nicht vor dem geplanten Ende der Videovorführung plaziert wird.

Wie geht es weiter?

Nach dem Abspielen der Animation soll mittels eines Überblendungseffektes ein Hintergrundbild (BILD-Spur »Afrika-Steppe«) eingeblendet werden, über das ein AnimBrush zu bewegen ist. Nun taucht aber ein Problem auf: Es sieht nicht gut aus, wenn nach dem Ende der Startanimation eine Zwangspause eingelegt wird, weil DVideo mit dem Laden des Hintergrundbildes und des AnimBrushes beschäftigt ist. Daher werden beide Spuren mit einem »LADEN«-Effekt eröffnet, der die betreffenden Dateien in das RAM schaufelt. Die der Animation vorgelagerte Position (0:00.00) sorgt dafür, daß nach Ablauf der Startanimation sofort im Geschehen fortgefahren wer-

stimmt, wird der Startzeiger auf 0:02.20 verschoben, der Endposition des SPIELANIM-Effektes auf der darüberliegenden Spur. Den Endezeiger verschieben Sie auf 0:04.10. Auch dieses Bild wird zum Zeitpunkt 0:10.20 aus dem Pool gepustet. Ein entsprechender »ENTLADEN«-Effekt übernimmt diese Aufgabe.

Die nächste Spur (ANIMPINSEL) wird mit der »AnimBrush«-Datei »Nashorn-16« (oder wie immer Sie Ihre Datei benannt haben) gefüllt, erhält den besprochenen »LADEN«-Effekt (0:00.00) und wird unter Beihilfe eines »POSITION«-Effektes (0:04.20) außerhalb des Sichtbereiches untergebracht. Ein »SPIELANIM«-Effekt (Start: 0:04.25, Ende: 0:06.45), der zeitlich synchron mit einem »BEWEGZU«-Effekt gekoppelt wird, sorgt für einen Ablauf, der sich

BiPF - Speicher-Check nach Bitplanes

Der Grafik- räuber



Großartige Supergrafik
zeichnet die meiste Software
für den Amiga aus.
Solche Bilder auszuson-
dern, war bislang
schwierig und aufwendig.
»BiPF« findet die Bit-
planes eines Programms
und speichert die
Bitmap als IFF-Bild.

links AMIGA rechts> der Speicherinhalt des Computers im Chip-Memory bis auf einen geringen Bruchteil unverändert erhalten bleibt. »BiPF« findet Grafikdaten in diesem Speicherabschnitt und definiert diese als speicherbare Bilddatei.

Chip-RAM »durchkämmen«

Um das Prinzip eines derartigen »Grafik-Verleihs« zu verstehen, sollte sich der Anwender von »BiPF« mit dem grundlegenden Aufbau und den Möglichkeiten der Grafik-Hardware beim Amiga auskennen. Das Programm bietet jedoch derart leicht zu bedienende Funktionen, daß sich auch der Einsteiger nicht zu scheuen braucht, mit »BiPF« zu arbeiten. Im Textkasten finden Sie die wichtigsten Begriffe im Umgang mit diesem Programm erläutert.

Um eine bestimmte Grafik aus einem Programm auszusondern, müssen Sie dieses zunächst laden. Wenn der gewünschte Bildschirm erscheint, lösen Sie beim Amiga einen RESET aus. »BiPF« sollte das erste und einzige Programm sein, das Sie danach laden und starten, um so wenig Speicher wie möglich zu verändern.

Lassen Sie sich von der wirren Pixelwüste nach dem Programmstart nicht täuschen: »BiPF« hat seine Arbeit bereits aufgenommen und bietet Ihnen ein Fenster in den Speicher des Amiga. Sie sehen auf dem Bildschirm das Chip-RAM ab Adresse 0. Wenn Sie nun die Maus bewegen, verändern sich einige Punkte im Bild. Klar - weil nämlich sämtliche Bits der Speicherzellen als Bildpunkte (Pixel) erscheinen - auch wenn diese mit der Grafik nichts zu tun haben und z.B. lediglich die aktuellen Mauskoordinaten abfragen.

Auf Grafik-Safari

Damit Sie sich mit der Arbeitsweise von »BiPF« besser vertraut machen, können Sie zur Übung den Screen der Workbench bzw. des Amiga-DOS im Speicher suchen

diesen als IFF-File auf Diskette speichern.

Nachdem »BiPF« gestartet wurde, müssen Sie die Leertaste drücken. Das Programm merkt sich damit die aktuelle Speicherposition und findet jederzeit dahin zurück.

Mit den Pfeiltasten des Ziffernblocks rechts auf der Tastatur oder den Cursortasten können Sie den Speicher bewegen:

<CRSR aufwärts> oder <8>:
Bildschirm nach oben verschieben

<CRSR abwärts> oder <2>:
Bildschirm nach unten

<CRSR links> oder <4>:
Bildschirm nach links verschieben

<CRSR rechts> oder <6>:
Bild wird nach rechts verschoben

Möchten Sie ein kontinuierliches Scrollen erreichen, müssen Sie die entsprechende Taste gedrückt halten. Werden die genannten Tasten zusammen mit <SHIFT> betätigt, geht das Scrollen des Bildschirms in allen vier Richtungen doppelt so schnell. Um einen ganzen Bildschirmbereich vorwärts oder rückwärts zu überspringen, müssen Sie im Ziffernblock die Taste <9> (Pg Dn) oder <3> (Pg Up) drücken.

Während Sie sich im Speicher des Amiga umsehen, wird der Speicherinhalt von »BiPF« nicht verändert. Haben Sie von diesem Ausflug in die Speicherlandschaft genug, dient die Tastenkombination <SHIFT 5> zur Rückkehr an die Ausgangsposition, die Sie mit der SPACE-Taste festgelegt hatten.

Wie finden Sie den Workbench-Screen? Drücken Sie zunächst die Taste <R> für »Hires«. Betreiben Sie Ihre Workbench im Interlace-

von Stefan Winterstein

Einige kommerzielle Programme des Amiga (z.B. »Turboprint II«, »Grabbit«) bieten zwar die Möglichkeit, den aktuellen Bildschirm zu speichern. Da, wo es am lohnendsten wäre, vor allem bei Grafiken aus Spielen, sind diese aber oft erfolglos.

Der »BitPlaneFinder« (BiPF) spürt nahezu alles auf, was eigene oder fremde Software an Grafik bietet (Bilder, Animationssequenzen, BOBs, Sprites usw.). Diese »ausgesonderten« Speicherbereiche lassen sich als IFF-Grafik speichern und mit jedem Grafikprogramm weiterverarbeiten, das der IFF-Norm entspricht.

Die Funktionsweise von »BiPF« beruht auf der Tatsache, daß nach einem RESET mit der bekannten Tastenkombination <CTRL AMIGA

Modus, muß zusätzlich die Taste <1> betätigt werden. Es ist bekannt, daß der Datenbereich des Workbench-Screen keine feste Adresse besitzt. Sie müssen ihn daher mit der Taste <3> auf dem Ziffernblock solange suchen, bis der entsprechende Speicherbereich auf dem Bildschirm erkennbar ist. Höchstwahrscheinlich muß der Workbench-Screen noch zurechtgerückt werden. Für diese Feineinstellung sollten Sie die erwähnten Zahlen- bzw. Cursortasten benutzen. Haben Sie alles ordnungsgemäß erledigt, sehen Sie die bekannte Workbench vor sich, allerdings nur zweifarbig (schwarzweiß). Der Grund: Sie haben zunächst lediglich die erste Bitplane gefunden.

Bitplane 2 liegt normalerweise ganz in der Nähe der ersten, meist direkt dahinter. Zum Auffinden der anderen Grafik-Bitplanes dienen Ihnen erneut Zahlentasten, diesmal in der oberen Reihe der alphanumerischen Amiga-Tastatur:

- <2> Bitplane 2
- <3> Bitplane 3
- <4> Bitplane 4
- <5> Bitplane 5
- <6> Bitplane 6

Die jeweils gewählte Bitplane ist nach dem Druck auf die entsprechende Zahlentaste aktiv und könnte mit den erwähnten Tasten (Cursor- und Ziffernblock) passend in die darunterliegenden Bitplanes eingefügt werden. Das Programm kennt dafür eine elegantere und weitaus komfortablere Methode: das Justieren mit der Taste <HELP>.

Wenn Sie die Taste <2> und anschließend <HELP> drücken, erscheint die Workbench mit korrekt justierter Bitplane 2: in vier Farben. Liegen die einzelnen Bitplanes zu einer Grafik direkt hintereinander im Speicher (dies trifft in 95 Prozent aller Fälle zu), berechnet »BiPF« anhand der eingestellten Bildgröße die exakten Positionen der anderen Bitplanes. Ein zeitraubendes »Justieren von Hand« mit den Ziffernblocktasten bzw. denen des Cursors entfällt dadurch.

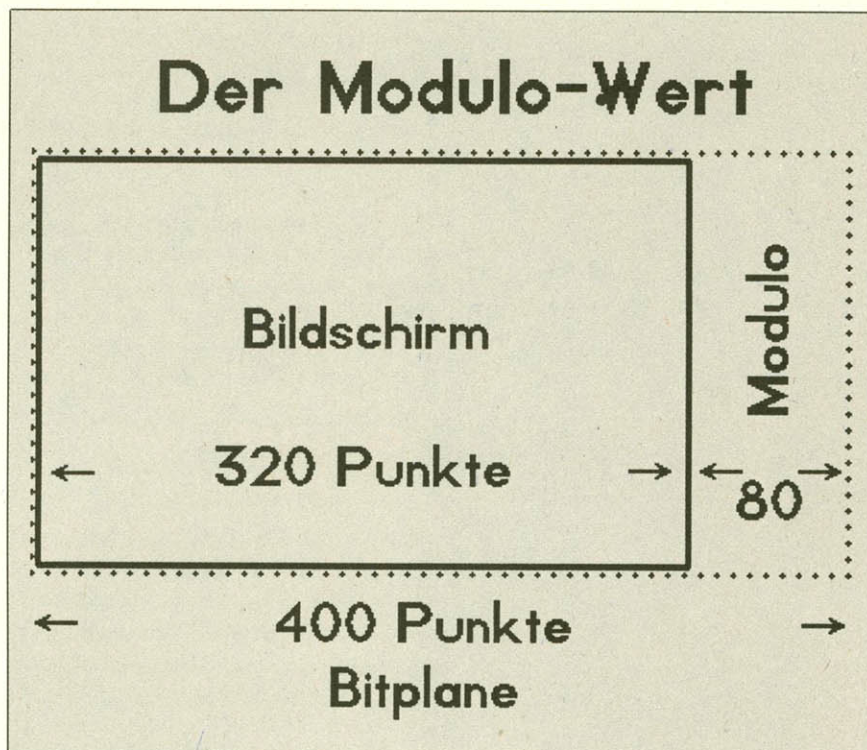


Bild 1. Der Modulo-Wert »80«: Die horizontale Länge einer Bitplane differiert um 80 Pixel gegenüber der Bildschirmdarstellung

Bewegung aller Bitplanes

Mit welchen Tasten die einzelnen Bitplanes über den Bildschirm bewegt werden, wurde bereits erläutert. Das Programm bietet eine weitere nützliche Funktion: das Bewegen aller gefundenen Bitplanes (von 1 bis 6) als Gesamtgrafik. Dazu ist nötig, ein wenig mit dem Modulo-Wert zu »spielen« (Bild 1). Drücken Sie auf der alphanumerischen Tastatur die Tilde-Taste ganz links oben neben der Eins. Damit schaltet das Programm in den »Merge-Modus«. Hierin wirken die Scroll-Bewegungen auf alle Planes, nicht nur auf die aktivierte. Die einzelnen Bitplanes werden wie eine gesamte Bitmap behandelt. Wenn Sie durch erneuten Druck auf <R> in den Lores-Modus umschalten, können Sie die aktuelle Grafik zwar noch erkennen, allerdings ist diese etwas durcheinander geraten. Drücken Sie jetzt die Taste <J> auf dem Ziffernblock genau 20mal. Der

Modulo-Wert erhöht sich dabei Schritt für Schritt, die Grafik sieht immer wirrer und unordentlicher aus – nach dem 20. Tastendruck erscheint die Workbench im Lores-Modus. Die Erklärung dieses Phänomens ist einfach: Die Zeilenlänge einer Hiresgrafik beträgt 640 Bildpunkte, die Bildschirmzeile ist jedoch nur 320 Pixel lang. Demnach gleicht der Modulo-Wert »20« (20 Words = 40 Byte = 320 Bit oder Pixel) diese Differenz zwischen Hires und Lores exakt aus. Mit der Taste </> auf dem Ziffernblock schalten Sie die Modulo-Funktion wieder ab. Erneuter Druck auf die Taste <'> deaktiviert den Merge-Modus. Tabelle 1 enthält alle Tastaturfunktionen des Programms.

Grafik speichern

Während mit »BiPF« gearbeitet wird, bleibt der Mauszeiger abgeschaltet. Trotzdem läßt sich das »BiPF«-Window jederzeit mit einem Mausklick deaktivieren. Das Programm schaltet automatisch zur Workbench zurück. Sie können mit anderen Programmen weiterarbeiten, »BiPF« bleibt im Background-Modus aktiv.

Es ist ein Kinderspiel, mit »BiPF« Grafiken in Amiga-Programmen zu finden. Nicht minder einfach ist es,

die gefundenen Grafik-Screens zu speichern. Dazu dient die Tastenkombination <SHIFT *>, wobei als Sternchen-Zeichen die Taste rechts oben im Ziffernblock gedrückt werden muß (nicht in der alphanumerischen Tastatur). »BiPF« schaltet zur Original-Workbench zurück. Sie können im entsprechenden Window den Filenamen des Bildes angeben. Gegebenenfalls muß der betreffende Pfad definiert werden. Auf eine File-Select-Box wurde vom Programm-autor bewußt verzichtet, da »BiPF« mit so wenig Chip-RAM wie möglich auskommen soll. Falls Sie aus irgendwelchen Gründen den Speichervorgang nicht durchführen wollen, drücken Sie die rechte Maustaste.

Tips und Hinweise

Optimale Ergebnisse lassen sich erzielen, wenn Ihr Amiga mit zusätzlichem Speicher ausgerüstet ist. Das Betriebssystem überschreibt in diesem Fall bedeutend weniger Chip-Memory. »BiPF« benötigt lediglich 1,5 KByte des Speichers. Wenn Sie während des RESET die externen Laufwerke abschalten, wirkt sich dies noch positiver auf den Speicher-verbrauch im Chip-RAM aus.

Es lohnt sich, nach BOBs Ausschau zu halten. Meist sind diese grafischen Gebilde recht schmal. Deshalb findet man sie nur mit einer geringen Bildbreite oder einem geeigneten (negativen) Modulo-Wert. Es lassen sich ganze Animations-Phasen, z.B. einer Figur, aufspüren. Diese können als Einzelbilder gespeichert und mit entsprechender Software (z.B. »DPaint III«) erneut zu einer Animation verbunden werden.

Faustregel: Sind Grafiken breiter als der sichtbare Bildschirm, benötigen diese einen positiven Modulo-Wert, schmälere Bitmaps einen negativen. Bei einem hohen Modulo-Wert liegen die Bitplanes nebeneinander im Speicher statt untereinander.

Die Features des Programms auf einen Blick:

- Unterstützung aller Grafikmodi inklusive HAM, EHB und Overscan,

Tastaturfunktion

Tastenkombination	Funktion
Scroll-Funktionen (Ziffernblocks rechts oder Cursorstasten): <8>, <2> <SHIFT 8>, <SHIFT 2> <4>, <6> <SHIFT 4>, <SHIFT 6> <9>, <3> <SPACE> <SHIFT 5>	
scrollt Bildzeile auf- oder abwärts Scrollen beschleunigen scrollt Bildzeile nach links bzw. rechts zur danebenliegenden Bitplane springen (bei Modulo) Bildschirm aufwärts/abwärts (Pg Up, Pg Dn) speichert die aktuelle Adresse der Bitplanes springt zur gespeicherten Position	
Bitplane-Funktionen (Zahlentasten alphanumerische Tastatur): <1> bis <6> <SHIFT 1> bis <SHIFT 6> <6> <'> <HELP> <SHIFT HELP>	
Die mit der entsprechenden Zahlentaste gewählte Bitplane wird aktiviert und läßt sich verschieben. Ist die Anzahl der eingeschalteten Bitplanes kleiner als die Nummer der aktivierten, schaltet das Programm die fehlenden dazu. gewählte Gesamtzahl der Bitplanes einsetzen. Beispiel: <SHIFT 3> aktiviert die Bitplanes 1 bis 3 gemeinsam. zusätzliches Einschalten des HAM-Modus Merge-Modus ein/aus Im Merge-Modus werden alle eingeschalteten Bitplanes bewegt. Ausgehend von der ersten werden die Adressen der nachfolgenden Bitplanes berechnet. Diese müssen dazu direkt hintereinander im Speicher stehen. Voraussetzung: Die Bildgröße ist passend eingestellt. berechnet die Adressen der vorhergehenden Bitplanes	
Grafikmodi: <R> <I> <H> <E>	
Umschalten zwischen Lores und Hires Interlace-Modus ein/aus HAM-Modus ein Extra-Halfbrite-Modus ein	
Größe des Bildschirms (Tasten im Ziffernblock): <ENTER> <+> <SHIFT +> <-> <SHIFT -> <. > <SHIFT .> <0> <SHIFT 0>	
schaltet um zwischen PAL- und NTSC-Zeilenaufösung erhöht die Bildhöhe um eine Zeile um zwei Zeilen erniedrigt die Bildhöhe um eine Zeile um zwei Zeilen erhöht die Bildbreite um 16 Pixel setzt die Breite auf 320/640 Bildpunkte erniedrigt Bildbreite um 16 Pixel setzt die maximale Bildbreite 384/768 Pixel Beim Verändern der Bildhöhe zentriert »BiPF« das Bild auf dem Monitor. Durch Änderung der Bildbreite wird die Grafik unkenntlich. Mit dem geeigneten Modulo-Wert läßt sich dies kompensieren. Overscan-Grafiken werden erst bei der richtigen Bildbreite sichtbar.	
Modulo-Modus (Tasten im Ziffernblock): <I> <SHIFT I> <J> <SHIFT J> </>	
Modulo-Wert um 16 Punkte erhöhen um 64 Punkte Modulo-Wert um 16 Punkte reduzieren um 64 Punkte Modulo auf »0« setzen	
Sonstige Tastaturfunktionen: <X>, <C> <Y>, <A> <S> <SHIFT *>	
justiert das Bild horizontal (schnellerer Ablauf mit der SHIFT-Taste) justiert das Bild vertikal (schnellerer Ablauf mit der SHIFT-Taste) erneutes Zentrieren des Bildes speichert die Grafik im IFF-Format. Während das File-Gadget aktiv ist, kann die Save-Funktion mit der rechten Maustaste verlassen werden. Bringt das Programm die Fehlermeldung »Save failed«, konnte die Grafik aus einem der folgenden Gründe nicht gespeichert werden: - Eine Datei mit dem angegebenen Namen ließ sich nicht öffnen, - es war nicht genügend Speicherplatz im RAM vorhanden, - DOS-Error (»Diskette voll« usw.) <ESC> Programm verlassen	

Tabelle 1. Auf einen Blick: Tastaturfunktionen des »BitPlaneFinder«

- beliebige Bildschirmgrößen (von 16 x 2 bis 768 x 560 Punkten),
- einfache Handhabung des Modulo-Registers,
- multitaskingfähig (Background-Modus),
- speichert gepackte IFF-ILBM-Files.

Einschränkungen

Bei Grafiken, die gleich am Anfang des Chip-Memory abgelegt wurden, ist »BiPF« machtlos. Dieser Speicherbereich wird nach einem RESET teilweise überschrieben. Ebenso gibt es einige Spiele, die bei einem RESET den kompletten Speicher löschen. In so einem Fall findet »BiPF« gähnende Leere vor. Außerdem gehen die Originalfarben einer Grafik in jedem Fall verloren. Das Programm arbeitet deshalb mit einer neutralen Palette von Grautönen, die bei den meisten Bildern in etwa den ursprünglichen Helligkeitswerten entsprechen. Mit der »Palette«-Funktion verschiedener Grafikprogramme des Amiga kann die Farbgebung anschließend jederzeit verändert bzw. angepaßt werden.

»BiPF« wurde mit dem C-Compiler »Aztek-C 3.6« entwickelt und arbeitet auf der untersten Ebene der Grafikdarstellung. Das Programm öffnet keinen Screen, sondern benutzt einen View-Port in einem eigenen View. Die Adressen der Pages werden direkt in der Struktur der Bitmap geändert. Die Darstellung des geänderten View (Copperlisten) übernimmt die Funktion »newview()«. Da »BiPF« so kompakt wie möglich bleiben sollte, wurde für »puts()« ein DOS-Ersatz sowie zwei Dummy-Funktionen verwendet. Das Executable ist nur knapp 11 000 Byte lang.

Der Sourcecode zu »BiPF« besteht aus vier Teilen:

BiPF.c (Listing 1) enthält die grundlegenden Prozeduren,

BiPF.h (Listing 2) beinhaltet die Definitionen,

BiPF_KEY.c (Listing 3) übernimmt die Ausführung der Tastenkommmandos,

BiPF_IFF.c (Listing 4) enthält die IFF-Save-Routine, die Sie auch in eigenen Programmen verwenden können.

BiPF-Glossar

Auflösung: Die Anzahl der Punkte, aus denen ein Bild in horizontaler und vertikaler Richtung besteht. Für europäische Amigas beträgt die Standardauflösung 320 x 256 (Loses) und 640 x 256 Pixel (Hires).

Bitmap: die Kombination aller Bitplanes einer Grafik.

Bitplane: Bit-, Bildpunktebene. Jedes Bild, das der Amiga darstellt, setzt sich aus bis zu sechs solcher Ebenen zusammen. Sie liegen »hintereinander« und bilden gemeinsam die Bitmap. Diese Punktemuster legen auch die Farbe eines Bildpunktes fest. Die Farbmöglichkeiten mehrerer Bitplanes:

1 Bitplane	2 Farben
2 Bitplanes	4 Farben
3 Bitplanes	8 Farben
4 Bitplanes	16 Farben
5 Bitplanes	32 Farben
6 Bitplanes	(64 Farben)

werden nur für die Spezialmodi HAM und EHB verwendet.

BOB: »Blitter-Object«, beweglicher Teil einer Grafik (z.B. laufende Figur, Raumschiff usw.).

Chip-Memory: der untere Teil des RAM im Amiga. Grafikdaten können nur dort abgelegt werden.

EHB-(Extra-Half-Bright-)Modus: Hier werden 64 Farben pro Bild verwendet. Wie eine HAM-Grafik besteht es aus sechs Bitplanes und verträgt sich demnach nicht mit dem Hires-Modus.

HAM-(Hold-AND-Modify-)Modus: eine spezielle Art der Grafikdarstellung, die bis zu 4096 Farben gleichzeitig auf den Bildschirm bringt. Die HAM-Bitmap besteht ebenfalls aus sechs Bitplanes, ist also sehr speicheraufwendig.

Hires: Kurzform für »High Resolution« (hohe Auflösung). Die horizontale Auflösung beträgt 640 Pixel. Es lassen sich nur vier Bitplanes und 16 unterschiedliche Farben verwenden.

Interlace: der berühmte Flimmermodus: Interlace verdoppelt die Anzahl der Zeilen (400 bzw. 512 Zeilen).

Loses: Kurzform für »Low Resolution« (niedrige Auflösung). Eine horizontale Grafikzeile besteht aus 320 Bildpunkten.

Modulo: Der Modulo-Wert ist ein Korrekturwert, der die »virtuelle« Länge einer Bildzeile beeinflusst. Im Speicher sind Grafiken zeilenweise als Bitplanes abgelegt. Im Normalfall sind die Zeilen ebenso lang wie die dargestellte Zeile am Bildschirm (Modulo = 0). Der Modulo-Wert steht für die Anzahl von Bildpunkten, um welche die Bildzeile im Speicher länger ist als deren Darstellung auf dem Monitor. Ein Beispiel: Eine Bitplane ist 400 Pixel breit, der Bildschirm stellt jedoch nur 320 Bildpunkte dar. Um das Bild korrekt zu zeigen, ist Modulo = 80 nötig. Würde man kein Modulo setzen, begänne die zweite Bildschirmzeile exakt da, wo die erste aufhört: bei Pixel Nr. 321 in der ersten Bildschirmzeile. Daß Bitplanes auf dem Monitor oft als undefinierbare »Pixelwüsten« erscheinen, liegt an einem falsch gesetzten Modulo-Wert. Bedingt durch die Hardware kann ein Modulo-Wert lediglich in 16-Pixel-Schritten durchgeführt werden. Da negative Werte vom Computer akzeptiert werden, lassen sich auch Bilder darstellen, die schmaler als der Bildschirm sind.

NTSC: amerikanische Fernsehnorm (200 Zeilen)

Overscan: Bei diesem Modus werden auch die Bildschirmränder genutzt. Auflösungen bis 384 x 290 (Loses) bzw. 768 x 580 Bildpunkten (Hires) sind möglich.

PAL: europäische Fernsehnorm. Im Gegensatz zu ihren amerikanischen Kollegen können PAL-Amigas 256 Zeilen darstellen. Leider nutzen dies viele Spiele nicht aus. Am unteren Bildrand bleibt meist ein freier Balken.

Nach dem abschließenden Abtippen der Batch-Datei **BiPF.bat** (Listing 5) kann das lauffähige Programm »BiPF« erzeugt werden. Starten Sie den Generierungsvorgang mit folgender Befehlseingabe:

```
execute BiPF.bat
```

Achten Sie darauf, daß sich die vorher genannten Dateien (Listing 1 bis 4) auf der Arbeitsdiskette befinden, auf der das endgültige Programm »BiPF« erzeugt werden soll.

Was man sonst noch mit »entlehnen« Grafiken anfangen kann, außer diese in eigenen Programmen zu verwenden? Denkbar sind beispielsweise Disketten-Label, Wand-Poster oder T-Shirt-Aufdrucke. Na? Da fällt doch auch Ihnen noch was ein? *ag/bl*

Achtung:

In letzter Minute erreichte uns eine Version, die auch auf dem Amiga 1000 lauffähig ist. Diese Version ist nur auf der Programm-Service-Diskette enthalten. Die fehlenden Tasten des Ziffernblocks haben nun Entsprechungen:

Taste im ZB	Alternativtaste
[ß
]	Accent-Taste
/	\
*	Backspace
—	+
ENTER	—
.	·
0	/


```

125 1W6      }
126 2X0     }
127 BM      newview(w,h,d)          /* displays a new View */
128 3h      SHORT w,h;              /* If called with width != 0, s
ets new size */
129 vE      UBYTE d;
130 2V      {
131 an4      if (w != 0) {           /* display size was changed */
132 lD8      InitBitMap (bm, d, w + modulo, h); /* Planepointer
untouched */
133 xy      vp.DWidth = w;
134 To      vp.DHeight = h;
135 aD      vp.DxOffset = (((MAXWIDTH << (H_SCALE)) - w) >>
1)
136 9k0      - (xcorrect << (H_SCALE));
137 IT8      vp.DyOffset = (((MAXHEIGHT << (L_SCALE)) - h) >
> 1)
138 S80      - (ycorrect << (L_SCALE));
139 Fk4      }
140 OJ      MakeVPort(&v,&vp);
141 FS      MrgCop (&v);
142 dW      LoadView (&v);
143 Jo0     }
144 uJ      avoid_copper_chrash()    /* prevents MakeVPort from maki
ng nonsense */
145 yb      {                       /* when in LACE.
*/
146 Kt4      LoadView (oldview);     /* This bug was haunting me for
days ! */
147 Y1      WaitTOF ();
148 hV      freeclists();
149 Pc      InitView (&v);
150 zI      v.ViewPort = &vp;
151 Ds      v.Modes = vp.Modes;
152 Sx0     }
153 gA      freeclists()             /* Frees the copperlists when t
oggeling LACE*/
154 pL      {                       /* Only call when old View is a
ctive ! */
155 IE4      FreeVPortCopLists (&vp);
156 Y1      FreeCprList (v.LOFCprList);
157 qe      if (MOD_LACE)
158 Tg8      FreeCprList (v.SHFCprList);
159 Z40     }
160 CV      newplaneaddress (inc, key, repeat) /* Handles scrolling reas
onably quick */
161 oTY      /* (IDMP-messages too slo
w) */
162 V10     register LONG inc;       /* amount to change adres
ses */
163 qe      UBYTE key;              /* which key code are we
dealing with */
164 PJ      BOOL repeat;            /* key repeat option on/o
ff */
165 b4      {
166 ev      register UBYTE *cia, ciacode;
167 4k      register SHORT i;        /* loop */
168 LM4      cia = 0xbfec01;         /* CIA-address for keycodes */
169 08      ciacode = 0;             /* keycodes from keyboard */
170 hB8      if (repeat && SHIFT) { /* select cia-register fo
r RAWKEY code */
171 XFC      ciacode = (key << 1) ^ 0xff;
172 mH8      }
173 9z      do {
174 dVC      if (MOD_MERGE) {
175 gQG      i = vdepth;
176 Gd      for (i--; i>=0; i--) {
177 1NK      bm -> Planes[ i ] += inc;
178 t5      bm -> Planes[ i ] &= availchip;
179 tOG      }
180 uPC      }
181 gw      else {
182 Z8G      bm -> Planes [actiplane] += inc; /* new p
lane-address */
183 59      bm -> Planes [actiplane] &= availchip;
184 yTC      }
185 3s      newview (0,0,0);
186 Zb8      } while(*cia == ciacode); /* repeat if correct
ciacode only */
187 5u      newview (0,0,0);
188 2X0     }
189 Zp      savepic()
190 OT      {
191 cO      USHORT id;              /* For File-Gadget */

```

```

192 KD4      SetWindowTitles(win, "BiPF: Save picture to",-1);
193 9L      LoadView (oldview);
194 br      OnGadget(&filegad,win,NULL);
195 X7      ModifyIDCMP(win,GADGETUP | MOUSEBUTTONS | INACTIVEWINDO
W | CLOSEWINDOW);
196 8s      ActivateGadget(&filegad,win,NULL);
197 gp      ON_SPRITE
198 lL      clearidcmp();
199 0o      WaitPort (win->UserPort);
200 6n      message = GetMsg(win->UserPort); /* simply use first
incoming message */
201 nS      class = message -> Class;
202 Cn      code = message -> Code;
203 hw      id = ((struct Gadget *) message -> IAddress) -> Ga
dgetID;
204 PQ      ReplyMsg(message);
205 xF8      switch (class)
206 GJ      {
207 BNC      case CLOSEWINDOW :
208 ZEG      do_it = FALSE;
209 IR      break;
210 qEC      case MOUSEBUTTONS:
211 J7G      if (code == MENUDOWN)
212 LUK      break;
213 OSC      case GADGETUP:
214 y5G      if ((id==MYGAD) && (gadinfo.NumChars > 0))
215 Ps      {
216 BeK      SetWindowTitles(win, "BiPF: Saving pict
ure ...",-1);
217 Lz      OffGadget(&filegad,win,NULL);
218 Ww      if ((write_if (name, &vp)) == NULL)
219 Tw      {
220 YUO      SetWindowTitles(win, "BiPF: Sorry,
SAVE failed !",-1);
221 kB      Delay (100);
222 a5K      }
223 b6G      }
224 Xg      break;
225 d88      }
226 CX4      WaitBOVP (&vp);
227 sH      OFF_SPRITE
228 71      SHIFT = FALSE;
229 XB      OffGadget(&filegad,win,NULL);
230 iJ      SetWindowTitles(win,TITLE,-1);
231 zz      ModifyIDCMP(win,IDCMP);
232 KT      newview (0,0,0);
233 lG0     }
234 wJ      main ()                 /****** Entry - Point *****/
235 Jc      {
236 Qh8      openstuff ();
237 vO5      /* CHIP-MEM-size (e.g. 0x7ffff) used for masking plane
pointers */
238 qP8      availchip = (SysBase -> MaxLocMem) -1;
239 RT      std_height = GfxBase -> NormalDisplayRows;
240 OP      vheight = std_height;
241 sy      vwidth = WIDTH;
242 eO      vdepth = DEPTH;
243 RZ      modulo = actiplane = 0;
244 im      xcorrect= XCORRECT;
245 nt      ycorrect= YCORRECT;
246 To      makeview ();             /* Display our View */
247 LB      do {
248 nbC      WaitPort (win->UserPort);
249 wI      while ((message = GetMsg(win->UserPort)) != NU
LL)
250 yR      {
251 bGG      class = message -> Class;
252 Ob      code = message -> Code;
253 CD      ReplyMsg(message);
254 k2      switch (class)
255 3W      {
256 6YK      case RAWKEY :
257 ynO      handlekey (code); break;
258 2QK      case MOUSEBUTTONS :
259 hvM      if (code == SELECTDOWN) {
260 k5O      WaitBOVP (&vp);
261 Qp      OFF_SPRITE
262 ZG      newview(0,0,0);
263 zT      SetWindowTitles(win, TITLE,-1);
264 G1M      }
265 CL      break;
266 JnK      case INACTIVEWINDOW :
267 LXO      LoadView (oldview);
268 py      ON_SPRITE

```



```

269 kb      SetWindowTitles(win,
270 lQR      "BiPF: Click to re-activate.",-1
           );
271 IRO      break;
272 EQK      case CLOSEWINDOW :
273 g00      (do_it = FALSE); break;
    
```

```

274 NmG      } /* of switch */
275 YWC      } /* of while */
276 mt8      } while (do_it);
277 3n      closeeverything ();
278 640 }    /* of main() */
(C) 1990 M&T
    
```

Listing 1. Die grundlegenden Prozeduren von »BiPF«

Programmname: BiPF.h

Computer: A500, A1000, A2000 mit Kickstart 1.2 & 1.3

Sprache: C

Bemerkung: siehe Text

Programmautor: Stefan Winterstein

```

1 h10 /******
2 My /*      BiPF.h      */
3 TT /*      */
4 HO /* Header-File for BiPF */
5 l5 /******
6 TO #define TITLE      "BiPF 1.2 (c) by ACE "
7 Sa #define REV      33L
8 J1 #define DEPTH      1
9 P5 #define WIDTH      320
10 ou #define MAXWIDTH      384 /* max Overscan (n*16) */
    /
11 lq #define MAXHEIGHT      290 /* max Overscan */
12 sG #define XCORRECT      30 /* default display offset
    s */
13 4Q #define YCORRECT      20
14 fy #define MYGAD      4711 /* filegadget-ID */
15 Oy #define L_SCALE      MOD_LACE /* Correction factors */
16 9h #define H_SCALE      MOD_HIRES
17 dm #define puts(a) Write(Output(), a, (long) strlen(a)); \
18 bjG Write(Output(), "\n", 1L);
    
```

```

19 y40 /* Jim's stuff */
20 KP #define MAKE_ID(a, b, c, d)\
21 Ok8 ( ((long)(a)<<24) + ((long)(b)<<16) + ((long)(c)
    <<8) + (long)(d) )
22 RJ0 /* these are the IFF types I deal with */
23 sJ #define FORM MAKE_ID('F', 'O', 'R', 'M')
24 IS #define ILBM MAKE_ID('I', 'L', 'B', 'M')
25 lo #define BMHD MAKE_ID('B', 'M', 'H', 'D')
26 rq #define CMAP MAKE_ID('C', 'M', 'A', 'P')
27 5k #define CAMG MAKE_ID('C', 'A', 'M', 'G')
28 Ne #define BODY MAKE_ID('B', 'O', 'D', 'Y')
29 CA #define DUMP      0
30 5X #define RUN      1
31 9j #define MINRUN      3
32 4y #define MAXRUN      128
33 b3 #define MAXDAT      128
34 LW #define MAXCOL      32
35 y5 union bytes4 { char b4_name[4]; long b4_type; };
36 V6 struct iff_chunk { union bytes4 iff_type; long iff_length; }
    ;
37 nv struct form_chunk { union bytes4 fc_type; /* == FORM */
38 HbK long fc_length; union bytes4 fc_subtype;
    };
39 ev0 struct BitMapHeader { UWORD w, h; UWORD x, y;
40 NyE UBYTE nPlanes; UBYTE masking; UBYTE compressio
    n; UBYTE pad1;
41 kn UWORD transparentColor; UBYTE xAspect, yAspect
    ;
42 bx WORD pageWidth, pageHeight;
43 OwD };
(C) 1990 M&T
    
```

Listing 2. Hier werden die Definitionen festgelegt

Programmname: BiPF_key.c

Computer: A500, A1000, A2000 mit Kickstart 1.2 & 1.3

Sprache: C

Compiler: Aztek-C 3.6a

Aufrufe: cc +l +s +x3 BiPF_key.c

Bemerkung: siehe Text

Programmautor: Stefan Winterstein

```

1 4y0 #include <exec/types.h>
2 gX #include <graphics/gfx.h>
3 aj #include <graphics/gfxbase.h>
4 g3 #include <graphics/gfxmacros.h>
5 Q4 #include <graphics/view.h>
6 wN #include "BiPF.h" /* get #defines */
7 5n extern struct BitMap *bm;
8 IQ extern struct View *oldview, v;
9 qF extern struct ViewPort vp;
10 JR extern USHORT actiplane;
11 Zj extern SHORT vwidth, vheight, modulo, xcorrect, ycorrect;
12 l1 extern UBYTE vdepth;
13 MQ extern BOOL MOD_HIRES, MOD_LACE, MOD_HAM, MOD_EHB, MOD_MER
    GE, SHIFT, do_it;
14 Zg PLANEPTR savepos [6]; /* save plane pointers */
15 SI handlekey (key)
16 Dv USHORT key;
17 Dg {
18 u9 LONG coa; /* change_of_adress */
19 dn SHORT delta; /* used with HELP */
20 AE register SHORT i; /* loop */
21 4e4 switch (key)
22 l1 {
23 l58 case 30 : /* up */
    
```

```

24 Ck case 77 : /* up */
25 j8G coa = -((vwidth + modulo) >> 3); /* Ch
    ange_of_Adress */
26 He newplaneaddress (coa,key,TRUE);
27 MV break;
28 Yk8 case 62 : /* down */
29 n4 case 76 : /* down */
30 ocG coa = ((vwidth + modulo) >> 3);
31 Mj newplaneaddress (coa,key,TRUE);
32 Ra break;
33 KR8 case 78 : /* right */
34 dK case 47 :
35 gOG if (SHIFT)
36 voK coa = vwidth >> 3; /* plane to the
    right */
37 QDG else
38 PUK coa = -2; /* 2 bytes */
39 RZG newplaneaddress (coa,key,FALSE);
40 Zi break;
41 9c8 case 79 : /* left */
42 FK case 45 :
43 o8G if (SHIFT)
44 3fK coa = - (vwidth >> 3); /* plane to
    the left */
45 YLG else
46 iyK coa = 2; /* 1 word */
47 dZG newplaneaddress (coa,key, FALSE);
48 hq break;
49 X28 case 31 : /* KP-9 */
50 9jG coa = ((vwidth + modulo) >> 3) * vheight
    ;
51 i7 if (MOD_MERGE)
52 6bK coa *= vdepth;
53 fnG newplaneaddress (coa,key,FALSE);
54 nw break;
55 X18 case 63 : /* KP-3 */
56 v6G coa = - ((vwidth + modulo) >> 3) * vheight;
    if (MOD_MERGE)
57 oD
    
```



```

58 ChK          coa = vdepth;
59 ltG          newplaneaddress (coa,key,FALSE);
60 t2           break;
61 nm8         case 46 :          /* SH-KP- 5   jump to position
*/
62 WJG          if (SHIFT) {
63 scK          i = vdepth;
64 SO           for (i--; i >= 0; i--) {
65 lrO           bm -> Planes [i] = savepos [i];
66 4ZK         }
67 9y          newview (0,0,0);
68 6bG         }
69 2B          break;
70 r38         case 64 :          /* SPACE   set position */
71 zQG          OFF_DISPLAY
72 Mb          Delay(3);
73 qS          ON_DISPLAY
74 3n          i = vdepth;
75 9N          for (i--; i >= 0; i--) {
76 nKO         savepos [i] = bm -> Planes [i];
77 FkG         }
78 BK          break;
79 gv8         case 6 :          /* activate 6th plane */
80 yEC         if (MOD_HIRES)
81 2rG         break;          /* no action if HIR
ES*/
82 9sC         actiplane = 5;          /* else continue */
83 Bw         if (vdepth == 6)
84 PAG         break;          /* ok if EHB or HAM
on */
85 cXe         /* else continue
*/
86 hTC         if (MOD_LACE)
87 kZG         avoid_copper_chrash();
88 tq8         case 37:          /* H am on
*/
89 d7C         MOD_HAM = TRUE;
90 a6         MOD_EHB = FALSE;
91 5o         if (MOD_HIRES) {
92 DEG         MOD_HIRES = FALSE;
93 rT         vwidth >>= 1;
94 W1C         }
95 FH         setvmode (HIRES | EXTRA_HALFBRITE, FALSE);
96 Rg         setvmode (HAM, TRUE);
97 ry         vdepth = 6;
98 WL         newview (vwidth, vheight, vdepth);
99 Wf         break;
100 m78        case 5:
101 JZC        if (MOD_HIRES)
102 PVG        break;          /* no action */
103 Vde        /* else continue */
104 tC8        case 1 : case 2 : case 3 : case 4 :
105 xrC        actiplane = key;          /* Planes [ 0-4 ] */
106 HE         actiplane--;
107 iV         if (SHIFT) {
108 sLG         vdepth = key;          /* activate n plane
s */
109 Yr         MOD_HAM = MOD_EHB = FALSE;
110 QL         setvmode (HAM | EXTRA_HALFBRITE, FALSE);
/* HAM OFF*/
111 nIC        }
112 p7         else /* set # of planes * according to
actiplane */
113 XpG         vdepth = ((vdepth < key) ? key : vdepth);
114 9vC         if (MOD_LACE)
115 C1G         avoid_copper_chrash();
116 odC         newview (vwidth, vheight, vdepth);
117 ox         break;
118 f78        case 0 :          /* ^ merge mode
*/
119 HvC         MOD_MERGE = !(MOD_MERGE);
120 mD         OFF_DISPLAY
121 90         Delay(3);
122 dF         ON_DISPLAY
123 u3         break;
124 Ym8        case 70 :          /* DEL   equalize pl
ane pointers */
125 EDC         for ( i = actiplane ; i < vdepth; i++)
126 9YD         bm->Planes [i] = bm->Planes[actiplane > 0
? actiplane-1 : 0];
127 A6C         newview (0,0,0); /* display size unchanged */
128 z8         break;
129 Cp8        case 95 :          /* HELP   calculate subsequen

```

```

t plane pointers */
130 SJC         delta = (vwidth >> 3); /* plane aside */
131 AY         if (modulo == 0)
132 OoG         delta == vheight;
133 GaC         if (SHIFT)
134 6PG         delta = -(delta); /* reverse */
135 VeC         for ( i=1; i < vdepth; i++)
136 uNK         bm -> Planes [i] = bm -> Planes[0] +
delta * i;
137 KGC         newview (0,0,0); /* display size unchanged */
/
138 9I         break;
139 Xj8        case 19 :          /* R esolution */
140 38C         MOD_HIRES = !(MOD_HIRES);
141 3f         if (MOD_HIRES) { /* HiRes on */
142 MuG         vwidth <<= 1;
143 6P         MOD_HAM = MOD_EHB = FALSE;
144 Gi         setvmode (HAM | EXTRA_HALFBRITE, FALSE); /*
HAM/EHB OFF*/
145 I5         if (vdepth > 4) {
146 HAK         actiplane = 3;
147 6k         vdepth = 4;          /* only 4 planes
in HIRES */
}
148 OtG        }
149 PuC        else
150 P2         vwidth >>= 1;
151 nPG         setvmode (HIRES, MOD_HIRES);
152 NUC         newview (vwidth, vheight, vdepth);
153 PE         break;
154 PY         case 23 :          /* I   Lace on/off */
155 828         avoid_copper_chrash();
156 rgC         MOD_LACE = !(MOD_LACE);
157 5k         if (MOD_LACE)
158 rd         vheight <<= 1; /* double vheight */
159 MfG        else
160 PCC         vheight >>= 1; /* halve vheight */
161 KKG         setvmode (LACE, MOD_LACE);
162 8DC         newview (vwidth, vheight, vdepth);
163 ZO         break;
164 ZI         case 18 :          /* E xtra-Halfbrite */
165 y28         if (MOD_HIRES) {
166 i1C         MOD_HIRES = FALSE;
167 QRG         vwidth >>= 1;
168 4g         }
169 jEC         MOD_HAM = FALSE;
170 Tr         MOD_EHB = TRUE;
171 My         setvmode (HIRES | HAM, FALSE);
172 rP         setvmode (EXTRA_HALFBRITE, TRUE);
173 hQ         vdepth = 6;
174 6D         newview (vwidth, vheight, vdepth);
175 la         break;
176 lu         case 67 :          /* ENTER */
177 Y98         vheight = ((vheight < (256 << L_SCALE)) ? 256
: 200) << (L_SCALE);
178 KwC         newview (vwidth, vheight, vdepth);
179 pe         break;
180 py         case 94 :          /* KP+ vheight + */
181 6v8         vheight += ((vheight >= (MAXHEIGHT << (L_SCA
LE))) ? 0 : ((SHIFT) ? 4 : 1));
182 NAC         newview (vwidth, vheight, vdepth);
183 ti         break;
184 t2         case 74 :          /* KP- vheight - */
185 G78         vheight -= ((vheight <= 4) ? 0 : ((SHIFT) ? 4
: 1));
186 77C         newview (vwidth, vheight, vdepth);
187 xm         break;
188 x6         case 60 :          /* KP- vwidth + */
189 t78         if (SHIFT)
190 BVC         vwidth = MAXWIDTH << (H_SCALE);
191 upG         else
192 viC         vwidth += ((vwidth >= ((MAXWIDTH) << (H_S
CALE))) ? 0 : 16);
193 xoG         newview (vwidth, vheight, vdepth);
194 4tC         break;
195 4D         case 15 :          /* KP-0 vwidth - */
196 NV8         if (SHIFT)
197 iOC         vwidth = WIDTH << (H_SCALE);
198 UfG         else
199 2pC         vwidth -= ((vwidth <= 16) ? 0 : 16);
200 qeG         newview (vwidth, vheight, vdepth);
201 BOC         break;
202 BK         case 90 :          /* [ Modulo - */
203 mC8         modulo -= (SHIFT ? 64 : 16);
204 sDC

```



```

205 F4      newview (vwidth, vheight, vdepth);
206 F0      break;
207 tK8     case 91 :          /* ]      Modulo + */
208 o7C     modulo += (SHIFT ? 64 : 16);
209 J8      newview (vwidth, vheight, vdepth);
210 JS      break;
211 tv8     case 58 :          /* - /      Modulo 0 */
212 Q7      case 92 :
213 BPC     modulo = 0;
214 OD      newview (vwidth, vheight, vdepth);
215 OX      break;
216 Gt8     case 49 :          /* y      y-offset - */
217 36C     ycorrect -= (SHIFT ? 2 : 1);
218 SH      newview (vwidth, vheight, vdepth);
219 Sb      break;
220 XY8     case 32 :          /* a      y-offset + */
221 v6C     ycorrect += (SHIFT ? 2 : 1);
222 WL      newview (vwidth, vheight, vdepth);
223 Wf      break;
224 Jf8     case 50 :          /* x      x-offset + */
225 x7C     xcorrect += (SHIFT ? 2 : 1);
226 aP      newview (vwidth, vheight, vdepth);
227 aj      break;
228 up8     case 51 :          /* c      x-offset - */
229 DfC     xcorrect -= (SHIFT ? 2 : 1);
230 eT      newview (vwidth, vheight, vdepth);

```

```

231 en      break;
232 Kc8     case 33 :          /* s      reset x-y-offsets */
/
233 XbC     xcorrect= XCORRECT;
234 ci      ycorrect= YCORRECT;
235 jY      newview (vwidth, vheight, vdepth);
236 js      break;
237 VQ8     case 93 :          /* SHIFT + KP - * Save Picture */
*/
238 xHC     if (SHIFT)
239 zVG     savepic();
240 nwC     break;
241 Rq8     case 96 :          /* either SHIFT-key pressed */
242 9v      case 97 :
243 9CC     SHIFT = TRUE; break;
244 SC8     case 224:         /* either SHIFT-key released */
245 50      case 225:
246 KNC     SHIFT = FALSE; break;
247 qT8     case 197 :        /* ESC */
248 GxC     (do_it = FALSE);
249 w5      break;
250 588     default : break;
251 3Y4     }
252 4Z0     }
(C) 1990 M&T

```

Listing 3. Die Ausführung der Tastenkommandos

Programmname: BiPF_iff.c

Computer: A500, A1000, A2000 mit Kickstart 1.2 & 1.3

Sprache: C

Compiler: Aztek-C 3.6a

Aufrufe: cc +l +s +x3 BiPF_iff.c

Bemerkung: siehe Text

Programmautor: Stefan Winterstein

```

1 4y0 #include <exec/types.h>
2 gX #include <graphics/gfx.h>
3 O2 #include <graphics/view.h>
4 DO #include <stdio.h>
5 kG #include "BiPF.h"
6 jh static USHORT
7 gD pack_row (file, source, size)
8 GW FILE *file;
9 4A char *source;
10 NO SHORT size;
11 7a {
12 js char c, lastc = '\0';
13 9Y short mode = DUMP;
14 ts short nbuf = 0;          /* number of chars in buffer */
15 DX short rstart = 0;      /* buffer index current run sta
rts */
16 LH short putsz;
17 17 char buf[MAXDAT+10];    /* [MAXDAT*3/2] not necessary
*/
18 Zw putsz = 0;
19 fL buf[0] = lastc = *source++; /* so have valid lastc */
20 NN nbuf = 1;
21 hy size--;                /* since one byte eaten. */
22 c0 for (; size; --size)
23 Jm4 {
24 Oh buf[nbuf++] = c = *source++;
25 4o switch (mode)
26 Mp8 {
27 gW case DUMP:
28 WuC if (nbuf > MAXDAT)
29 PsG {
30 Cpk if (putc(nbuf-2, file) == EOF)
31 yt0 return(0);
32 KRK if (write(buf, nbuf-1, 1, file) != 1)
33 Ov0 return(0);
34 tIG putsz += nbuf;
35 ps buf[0] = c;
36 dd nbuf = 1;
37 rr rstart = 0;
38 Xg break;

```

```

39 d8 }
40 xLC if (c == lastc)
41 b4G {
42 qJ if (nbuf - rstart >= MINRUN)
43 d6K {
44 KH if (rstart > 0)
45 f8O {
46 nRS if (putc(rstart-1, file) == EOF)
return(0);
if (fwrite(buf, rstart, 1, file)
!= 1)
return(0);
putsz += rstart+1;
}
mode = RUN;
}
else if (rstart == 0)
mode = RUN;
}
else
rstart = nbuf - 1; /* first of run */
break;
case RUN:
if ((c != lastc) || (nbuf - rstart > MAXRUN))
{ /* output run */
if (putc( -(nbuf - rstart - 2), file) =
= EOF)
return(0);
if (putc( lastc, file) == EOF)
return(0);
putsz += 2;
buf[0] = c;
nbuf = 1;
rstart = 0;
mode = DUMP;
}
break;
} /* of switch */
lastc = c;
} /* of for */
77 ue0 switch (mode)
78 Cf8 {
79 WM case DUMP:
80 ubG if (putc(nbuf-1, file) == EOF)
81 mh0 return(0);
82 GLG if (fwrite(buf, nbuf, 1, file) != 1)
83 oj0 return(0);
84 kKG putsz += nbuf+1;
85 IRC break;
86 rG8 case RUN:
87 zoG if (putc( -(nbuf - rstart - 1), file) == EO
F)
return(0);
88 to0 if (putc( lastc, file) == EOF)
89 GxC return(0);
90 vq0

```



```

91 t8G          putsize += 2;
92 PYC          break;
93 V08          }
94 5W0 return(putsize);
95 X2          }
96 1V          static LONG          /* MUST return LONG, not int */
97 lQ          pack_bitmap(file, bm, len)
98 iy          FILE *file;
99 D6          struct BitMap *bm;
100 nr          WORD len;
101 Z2          {
102 fA          USHORT i, j, row_length;
103 MJ          ULONG plane_offset, compressed_length;          /* MUST BE A LONG
G */
104 RG          compressed_length = 0;
105 co          plane_offset = 0;
106 c6          for (i=0; i<bm->Rows; i++)
107 f88          {
108 Sd          for (j = 0; j < bm->Depth; j++)
109 hAC          {
110 Tv          if ( (row_length =
111 ARJ          pack_row(file, bm->Planes[j] + plane_of
fset, len) ) == 0)
112 kDK          {
113 ID          return(0);
114 qL          }
115 dSC          compressed_length += row_length;
116 sN          }
117 hn8          plane_offset += bm-
BytesPerRow;          /* consider
modulo */
118 uP          }
119 gXO          if ( compressed_length & 1) /*check to see odd length */
120 sL8          {
121 1M          if ( puts( 0, file) == EOF) /*pad with 0 to make it
even */
122 RMC          return(0);
123 n48          compressed_length++;
124 OV          }
125 IQO          return(compressed_length);
126 2X          }
127 1A          SHORT write_iff(name, vport)          /* <<<<< ENTER HER
E >>>>> */
128 Zs          char *name;
129 lz          struct ViewPort *vport;
130 2V          {
131 FV          FILE *file;
132 yh          struct form_chunk chunk;
133 ic          struct iff_chunk ichunk;
134 Xp          struct BitMapHeader header;
135 Ha          struct ColorMap *cm;
136 Lc          register struct BitMap *bits;
137 jj          ULONG color, pic_length, bits_size, modes;
138 BT          SHORT i;
139 Oa          register short j;
140 hI          register short row_offset;
141 4M          UBYTE c;
142 VH          if ((file = fopen(name, "w") ) == 0) {
143 mh8          return(0);
144 Kp0          }
145 Ob          bits = vport -> RasInfo -> BitMap;
146 84          cm = vport -> ColorMap;
147 9U          chunk.fc_type.b4_type = FORM;          /* say its a F
ORM ILBM */
148 r3          chunk.fc_subtype.b4_type = ILBM;          /* chunk.fc_length is s
et later */
149 ag          if (fwrite(&chunk, sizeof(chunk), 1, file) != 1)
150 to8          return(0);
151 uIO          ichunk.iff_type.b4_type = BMHD;          /* here comes a Bi
tMapHeader */
152 iy          ichunk.iff_length = sizeof(header);
153 yQ          if (fwrite(&ichunk, sizeof(ichunk), 1, file) != 1)
154 xs8          return(0);
155 HXO          header.masking = 0;          /* initialize the BitMa
pHeader */
156 iH          header.padi = 0;
157 qK          header.transparentColor = 0;
158 NO          header.compression = 1;          /* to cut a long story
short ... */
159 7G          header.xAspect = 5;          /* well, almost ... */
160 HM          header.yAspect = 11;
161 vy          header.pageWidth = vport -> DWidth;
162 po          header.pageHeight = vport -> DHeight;
163 3A          header.w = vport -> DWidth;          /* save as displayed,
not BitMap */

```

```

164 ok          header.h = vport -> DHeight;          /* =====
===== */
165 10          header.x = vport -> DxOffset;
166 AZ          header.y = vport -> DyOffset;
167 Y1          header.nPlanes = bits -> Depth;
168 z1          if (fwrite(&header, sizeof(header), 1, file) != 1)
169 C78          return(0);
170 9IO          ichunk.iff_type.b4_type = CAMG;          /* utter a CAMG - Chunk
for ViewModes */
171 hG          ichunk.iff_length = 4;          /* 1 LONG = 4 BYTES */
172 Oc          if (fwrite( &ichunk, sizeof(ichunk), 1, file) != 1)
173 GB8          return(0);
174 p30          modes = vport->Modes;          /* LONG */
175 EU          if (fwrite( &(modes), 4, 1, file) != 1)
176 JEB          return(0);
177 aw0          ichunk.iff_type.b4_type = CMAP;          /* squirt out the color
map */
178 wF          ichunk.iff_length = MAXCOL*3;
179 Oq          if (fwrite(&ichunk, sizeof(ichunk), 1, file) != 1)
180 NI8          return(0);
181 Ee0          for (j=0; j < MAXCOL; j++){          /* ColorTable
OxOrgb */
182 114          color = (SHORT) GetRGB4(cm, j);
183 uY          c=((color) & 0x0F00) >> 4);          /* red */
184 9T          if (fwrite(&c, 1, 1, file) != 1)
185 SNC          return(0);
186 Ou4          c=((color) & 0x00F0);          /* green */
187 CW          if (fwrite(&c, 1, 1, file) != 1)
188 VQC          return(0);
189 wd4          c=((color) & 0x000F) << 4);          /* blue */
190 FZ          if (fwrite(&c, 1, 1, file) != 1)
191 YTC          return(0);
192 6b0          }
193 OX          ichunk.iff_type.b4_type = BODY;          /* BODY */
194 U6G          /* ichunk.iff_length is set later */
195 e60          if (fwrite(&ichunk, sizeof(ichunk), 1, file) != 1)
196 dY8          return(0);
197 sa0          if ( ( bits_size = pack_bitmap(file, bits, ((vport->DWidth)
>>3))) == 0)
198 fa9          return(0);
199 ch0          pic_length= sizeof(struct BitMapHeader) + (MAXCOL * 3) + 8
+ bits_size +
200 ZTC          (4 * sizeof(struct iff_chunk));
201 D3d          /* adapt, if more ch
unks added !*/
202 sd0          if (fseek(file,41,0) != NULL)          /* Set FORM-len
gth in file */
203 kf4          return(0);
204 dJO          if (fwrite(&pic_length,4, 1, file) != 1)          /* write LONG *
/
205 mh8          return(0);
206 Um0          if (fseek(file,-(bits_size)-4,2) != NULL)          /* set BODY-len
gth in file */
207 oJ4          return(0);
208 vh0          if (fwrite(&bits_size,4, 1, file) != 1)          /* write LONG *
/
209 ql8          return(0);
210 yf0          fclose(file);
211 ma          return(1);          /* phew, that's it !*/
212 Qv          }
(C) 1990 M&T

```

Listing 4. Die IFF-Save-Routinen

Programmname:	BiPF.bat
Computer:	A500, A1000, A2000 mit Kickstart 1.2 & 1.3
Sprache:	Amiga-DOS
Bemerkung:	Batch-Datei zum Generieren von BiPF
Programmautor:	Stefan Winterstein

```

1 2W0 ;#
2 9N ;#          BiPF.bat
3 15 ;#          Execute it to generate BiPF
4 5Z ;#
5 8G cc BiPF.c +1 +x3 -s
6 p1 cc BiPF_iff.c +1 +x3 -s
7 Wr cc BiPF_key.c +1 +x3 -s
8 8L ln +q BiPF BiPF_key.o BiPF_iff.o -lc32
(C) 1990 M&T

```

Listing 5. Diese Batch-Datei hilft beim Erzeugen eines lauffähigen Programms

»DISK-UTILITY« — der CLI-Ersatz

von Thomas Hauser

Wer kennt nicht »Disk-Master«, ein typisches Hilfsprogramm für alle, die mit dem CLI auf Kriegsfuß stehen. Aber »DISK-UTILITY« kennen Sie noch nicht. Kein Wunder – schließlich stellen wir Ihnen dieses tolle Programm jetzt das erste Mal vor. Betrachten Sie einmal Bild 1. Was, Sie sehen keinen großen Unterschied zu »Disk-Master«? Aber es gibt viele kleine und einen großen Unterschied: »Disk-Master« müssen Sie kaufen, »DISK-UTILITY« können Sie abtippen (ist aber nur sinnvoll, wenn Sie »GFA-Basic« besitzen) oder sowohl den Quelltext als auch eine compilierte Version mit der zu diesem AMIGA-Sonderheft 13 erschienenen Leserservice-Diskette erwerben.

Zunächst kopieren Sie bitte die Dateien »DISK__UTILITY.Def« und »DISK__UTILITY.Bef« von der Leserservice-Diskette in das »S«-Directory Ihrer Boot-Diskette. Machen Sie zuerst das »DISK-UTILITY«-Directory zum aktuellen und geben Sie dann im CLI ein:

```
copy DISK__UTILITY.Def to SYS:
  s/DISK__UTILITY.Def
copy DISK__UTILITY.Bef to SYS:
  s/DISK__UTILITY.Bef
```

Wer »GFA-Basic« hat, hat sicher keine Probleme, das Hauptprogramm (Listing 1) aufzurufen. Für die User der compilierten Version gilt: Entweder Sie klicken das Programm-Icon auf der Workbench zweimal an und starten so »DISK-UTILITY« oder Sie geben im CLI ein:

```
run DISK-UTILITY
```

Wenn Sie jedoch aus dem CLI starten wollten und die beiden »Copy«-Vorgänge noch nicht getätigt ha-

CLIMATE

»Nomen est omen« – und GFA-Basic macht's möglich. Was andere für teures Geld verkaufen, stellt die AMIGA-Sonderhefte-Redaktion der Öffentlichkeit zur Verfügung: einen DOS-Manager.



Bild 1. Die geniale Benutzeroberfläche von »DISK-UTILITY«

ben, werden Sie von einem freundlichen Requester empfangen, der Sie darauf aufmerksam macht, daß die Datei »DISK__UTILITY.Bef« nicht gefunden werden konnte. So einfach geht's also doch nicht.

Im Gegensatz zu »Disk-Master« oder »CLIMATE« arbeitet »DISK-UTILITY« mit den Amiga-DOS-Kommandos (»Copy«, »Rename« usw.). Dies hätte aber zur Folge, daß »DISK-UTILITY« nicht korrekt mit ARP 1.3 (Amiga-DOS Resource Project) läuft, da ARP den »Resident«-Befehl nicht kennt. Hier hilft die »DISK__UTILITY.Bef«-Datei, in der alle residenten Kommandos und Extensions ausgelagert sind. Sie wird von »DISK-UTILITY« immer im »S«-

Directory gesucht. In Listing 2 sehen Sie, wie diese Datei für die Workbench 1.3 aussehen sollte, wobei das »PURE«-Bit abgefragt wird.

Listing 2:

```
SYS:C/Copy pure
SYS:C/Assign pure
SYS:C/Delete pure
SYS:C/Info pure
SYS:C/Install pure
```

Listing 3 zeigt die Datei in der für ARP 1.3 modifizierten Form.

Listing 3:

```
SYS:C/Copy force
SYS:C/Assign force
SYS:C/Delete force
SYS:C/Info force
SYS:C/Install force
```

Wichtig dabei ist, daß jede dieser Dateien genauso aussehen muß wie im jeweiligen Listing, also mit fünf Befehlen und voller Pfad-Angabe. Leider fiel eine Modifikation unter den Tisch: »Resident« steht noch im Programmtext. Das bedeutet, daß ARP-User den »ARes«-Befehl mit `rename SYS:c/ARes to SYS:c/Resident`

in »Resident« umbenennen müssen, damit »DISK-UTILITY« bei seiner Suche nach dem »Resident«-Befehl im »C«-Directory der Boot-Diskette nicht ins Leere greift. Aus Kompatibilitätsgründen würde ich ARP-Benutzern empfehlen, ein paar Byte auf der Boot-Diskette zu opfern und statt »Rename« lieber mit »Copy« ein Duplikat des »ARes«-Kommandos mit dem Namen »Resident« anzulegen.

Die zweite Extern-Datei »DISK__UTILITY.Def« enthält eine Liste der anzusprechenden Devices und weitere Steuerschalter für »DISK-UTILITY«.

Listing 4: (DISK-UTILITY.Def)

DFO:
DF1:
DHO:
DH1:
RAM:
RAD:
SYS:C/ED
ZOO
NEIN
NEIN

Sollten Sie das Listing abtippen, möchte ich Sie darauf aufmerksam machen, daß Sie diese Datei nicht unbedingt eingeben müssen, um mit »DISK-UTILITY« arbeiten zu können. Es gibt Default-Werte im Programm.

Lassen Sie mich folgende Annahme treffen: Sie haben die Dateien »DISK__UTILITY.Bef« und »DISK__UTILITY.Def« im »S«-Directory Ihrer Boot-Diskette stehen und das Hauptprogramm »DISK-UTILITY« in einem beliebigen Verzeichnis Ihrer Boot-Disk. Auch ob Sie das Programm durch Anklicken des Icons oder durch einen Aufruf aus dem CLI gestartet haben, spielt keine Rolle – Sie sehen auf Ihrem Monitor die Oberfläche von »DISK-UTILITY«, wie in Bild 1 zu sehen.

Der Bildschirm ist in eine linke und eine rechte Hälfte unterteilt. Getrennt werden die beiden durch eine Menü-Leiste. Jeder dieser Menü-Punkte wirkt sich immer nur auf das »Quell-Laufwerk« und seine Dateien aus. Die obersten sechs Gadgets sind für die Anwahl verschiedener Devices reserviert. Wenn Sie jetzt z.B. den Schalter »DFO:« anklicken, so wird das Root-Directory der Disk in Laufwerk 0 eingelesen und alphabetisch geordnet in jener Hälfte des Bildschirms angezeigt, die mit »Quell-Laufwerk« bezeichnet ist. Sub-Directories werden mit einem vorangestellten Asterisk (»*)« gekennzeichnet und am Anfang der Liste ausgegeben. Wollen Sie in ein Unterverzeichnis wechseln, klicken Sie entsprechenden Directory-Eintrag einfach mit der rechten Maustaste an. Wollen Sie aus einem Unterverzeichnis ins nächsthöhere zurückkehren, klicken Sie bitte das Menü-Gadget »PARENT« an. Die Anzahl der auf dem Device noch freien Spei-

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Chefredakteur: Wolfram Höfler –verantwortlich für den redaktionellen Teil

Leitender Redakteur: Andreas Greil (ag)

Redaktion: Albert Petryszyn (pe)

Mitarbeiter dieser Ausgabe: Peter Aurich (pa), Harald Beiler (bl), Ulrich Brieden (ub), Winfried Dietmayer, Dr. Rudolf Egg, Wolfgang Fiebig, Walter Friedhuber, Arnd Goebel, Thomas Hauser, Martin Jobst, Reiner Moll, Norbert Paul, Stephan Quinkertz (sq), Angela Schmidt, Jürgen K. Singer, Gerhard Stock, Marco Vitolini-Naldini

Redaktionsassistentz: Catharina Winter, Petra Kessner (414)

Telefax: 089/4613-433, Hotline: Donnerstag 14 bis 17 Uhr

Alle Artikel sind mit dem Kurzzeichen des Redakteurs und/oder mit dem Namen des Autors/Mitarbeiters gekennzeichnet

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt & Technik Verlag AG verlegten Publikationen. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Art-director: Friedemann Porscha

Titelgestaltung: Wolfgang Berns

Layout: Alexander Kowarzyk (Cheflayouter), Sigrid Groß, Marian Schwarz

Bildredaktion: Janos Feitser (Lt.), Sabine Tennstaedt; Roland Müller (Fotografie); Ewald Standke, Norbert Raab (Spritzgrafik); Werner Nienstedt (Computergrafik)

Anzeigendirektor: Ralph Peter Rauchfuss

Anzeigenleitung: Philipp Schiede (399) – verantwortlich für die Anzeigen

Telefax: 089/4613-775

Anzeigenverwaltung und Disposition: Monika Burse (147)

Auslandsrepräsentation:

Auslandsniederlassungen:

Schweiz: Markt & Technik Vertriebs AG, Kollerstr. 37, CH-6300 Zug, Tel. 042-440550/660, Fax 042-4 15 770, Telex: 862329 mut ch

USA: M&T Publishing Inc., 501 Galveston Drive Redwood City, CA 94063, Telefon: (415) 366-3600, Telex 752-351

Österreich: Markt & Technik Ges. mbH, Große Neugasse 28, A 1040-Wien, Telefon: 0222/5871393, Telex: 047-132532

Anzeigen-Auslandsvertretung:

England: F. A. Smyth & Associates Limited, 23a, Aylmer Parade, London, N2 OPQ. Telefon: 0044/1/3405058, Telefax: 0044/1/3419602

Israel: Baruch Schaefer, Haeskel-Str. 12, 58348 Holon, Israel, Tel. 00972-3-5562256

Taiwan: AIM International Inc., 4F-1, No. 200, Sec. 3, Hsin-I Rd.; Taipei, Taiwan, R.O.C., Tel. 00886-2-7548631, -7548633, Fax 00886-2-7548710

Korea: Young Media Inc., C.P.O. Box: 6113, Seoul/Korea, Tel. 0082-2-7564819, /-7742759, Fax 0082-2-7575789

Vertriebsdirektor: Uwe W. Hagen

Vertriebsmarketing: Petra Schlichthärle

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: ip Internationale Presse, Hauptstätter Straße 96, 7000 Stuttgart 1, Tel. 0711/6483-110

Bezugsmöglichkeiten: Leser-Service: Telefon (089) 4613-366. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen.

Verkaufspreis: Das Einzelheft kostet DM 16,-

Produktion: Technik: Klaus Buck (Lt./180), Wolfgang Meyer (Stellv./887); Herstellung: Otto Albrecht (Lt./917)

Druck: SOV Graphische Betriebe, Laubanger 23, 8600 Bamberg

Urheberrecht: Alle in diesem Heft erschienenen Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen, vorbehalten. Reproduktionen, gleich welcher Art, ob Fotokopie, Mikrofilm oder Erfassung in Datenverarbeitungsanlagen, nur mit schriftlicher Genehmigung des Verlages. Aus der Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen oder verwendeten Bezeichnungen frei von gewerblichen Schutzrechten sind.

Haftung: Für den Fall, daß in diesem Heft unzutreffende Informationen oder in veröffentlichten Programmen oder Schaltungen Fehler enthalten sein sollten, kommt eine Haftung nur bei grober Fahrlässigkeit des Verlages oder seiner Mitarbeiter in Betracht.

Sonderdruck-Dienst: Alle in dieser Ausgabe erschienenen Beiträge sind in Form von Sonderdrucken zu erhalten. Anfragen an Reinhard Jarczok, Tel. 089/4613-185, Fax 4613-774.

© 1990 Markt & Technik Verlag Aktiengesellschaft

Vorstand: Otmar Weber (Vors.), Bernd Balzer

Verlagsleitung: Wolfram Höfler

Direktor Zeitschriften: Michael M. Pauly

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen: Markt & Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon 089/4613-0, Telex 522052, Telefax 089/4613-100

Telefon-Durchwahl im Verlag: Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089/4613 und dann die Nummer, die in den Klammern hinter dem jeweiligen Namen angegeben ist.

cherplätze wird in der obersten Bildschirmzeile angezeigt.

Unter den Zeilen, in denen »Quell-Laufwerk« und »Ziel-Laufwerk« steht, lesen Sie die vollständige Pfadangabe, die zu den Einträgen im darüberliegenden File-Window führt. Und am untersten Bildschirmrand, für jede Bildschirmhälfte separat, wird die Summe der Bytes der angewählten Files ausgegeben (siehe Bild 2).

Beim »**FORMAT**«-Kommando kommt die Ausrichtung von »**DISK-UTILITY**« auf WB 1.3 deutlich heraus. Die anwählbaren Optionen »**FFS**« für das »Fast File System« und »**QUICK**« für das Kurzformatieren durch Löschen der Verwaltungstabellen gibt es nicht unter Workbench 1.2.

Benützen Sie »**COPY**«, indem Sie die gewünschten Einträge durch Anklicken im »Quell-Disk«-Fenster markieren und dann den »**COPY**«-Schalter drücken. »**MAKEDIR**« wirkt immer nur auf die »Quell-Disk«, »**RENAME**« holt nacheinander die markierten File-Namen in ein Eingabefenster und »**DELETE**« gibt Ihnen durch eine Sicherheitsabfrage noch eine letzte Chance, abzubrechen. Nicht ganz so rücksichtsvoll ist »**DISKCOPY**«. Wenn Sie nach der Eingabe von Quell- und Ziellaufwerk <RETURN> drücken, geht's unbarmherzig los - also Vorsicht. »**INSTALL**« wiederum kann mehr, als auf den ersten Blick ersichtlich. Das nach Anklicken des »**INSTALL**«-Gadgets geöffnete Eingabefenster erweckt den Eindruck, nur die Eingabe des gewünschten Laufwerks sei möglich. Verwenden Sie ruhig die komplette CLI-Syntax, also z.B. »df0:check« - das Ergebnis dieser Eingabe sehen Sie dann im hinter dem »**DISK-UTILITY**«-Schirm liegenden Ausgabefenster. Dort wird automatisch umgeschaltet, wenn Sie durch Anklicken des »**INFO**«-Gadgets den gleichnamigen Befehl aktivieren, denn das Ergebnis von »**INFO**« wird dort ausgegeben.

Mit »**EDITOR**« rufen Sie den voreingestellten Textverarbeiter auf den Plan. Ein zu diesem Zeitpunkt angewähltes File, z. B. eine Text-Datei wie »startup-sequence«, wird automatisch an diesen Editor übergeben und nachgeladen. Vorausgesetzt, der Editor unterstützt diese Art der File-Namenübergabe.

Mit »**ALLE**« werden sämtliche Files des aktuellen Quell-Disk-Directory angewählt, mit »**CLEAR**« hingegen alle angewählten Files wieder deaktiviert. »**DISK-UTILITY**« erlaubt auch, aus dem laufenden Programm heraus ein anderes zu starten. Dazu muß nur das gewünschte Programm im »Quell-Disk«-Fenster und anschließend »**PRG START**« angeklickt werden. Eine Parameter-Übergabe ist allerdings nicht vorgesehen.

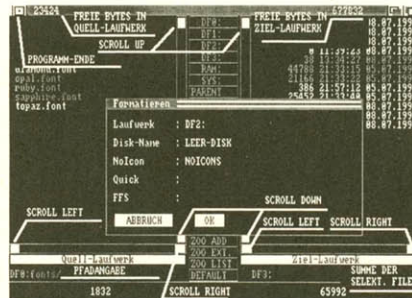


Bild 2. Die »neuralgischen Punkte« von »**DISK-UTILITY**«

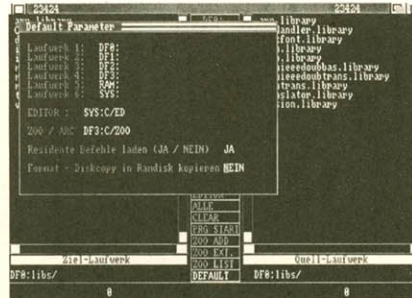


Bild 3. »**DISK-UTILITY**« - die Eintragungen für »**DISK-UTILITY.Def**«

Für alle Archive sind die nächsten drei Gadgets interessant: »**ZOO ADD**«, »**ZOO EXT**«, und »**ZOO LIST**«. Um in »**DISK-UTILITY**« Dateien mit »**ZOO**« zu verdichten, klicken Sie die gewünschten Dateien im Quell-Disk-Fenster an und drücken danach »**ZOO ADD**«. Im darauf geöffneten Eingabefenster tragen Sie Device und gewünschten Namen des Archive-Files ein. Nach <RETURN> startet »**ZOO**« und arbeitet ein File nach dem anderen ab. Die Meldungen, die »**ZOO**« während des Ablaufs ausgibt, sehen Sie im Ausgabe-Fenster hinter der »**DISK-UTILITY**«-Oberfläche. Es ist aber nicht möglich, durch die Anwahl eines Directory-Eintrags das ganze Directory zu

packen. In die andere Richtung arbeitet »**ZOO EXT**« (»**EXT**« wie »**EXTRACT**«). Klicken Sie das Archive-File im Quell-Disk-Fenster an und dann auf »**ZOO Ext**«. Wenn Sie die Vorgaben durch Drücken von <RETURN> akzeptieren, wird als Ziel-Device, in das die entpackten Files geschrieben werden sollen, »**RAM**« vorgeschlagen. Diese Vorgabe können Sie akzeptieren (<RETURN> genügt) oder überschreiben. In jedem Fall wird nach <RETURN> der Entpack-Vorgang aktiviert, über dessen Verlauf Sie wieder im Ausgabefenster informiert werden. Sollten Sie vergessen haben, in welchem Archiv-File sich die Datei befindet, die Sie entpacken wollen, hilft Ihnen »**ZOO LIST**« bei der Suche, denn damit wird das Inhaltsverzeichnis eines angeklickten Archiv-Files angezeigt.

Bleibe noch das »**DEFAULT**«-Gadget am untersten Ende der Menü-Leiste. Nach dem Anklicken dieses Gadgets öffnet sich ein Requester (Bild 3), in dem Sie die gewünschten Vorgaben für Laufwerke etc. eintragen. Sechs Laufwerke können Sie vordefinieren, wobei auch die Angabe logischer Devices wie etwa »**SYS**« oder ein Disketten-Name möglich ist. Einzige Einschränkung: der Eintrag (ohne den notwendigen Doppelpunkt) darf nicht mehr als fünf Zeichen lang sein. Der aufzurufende Editor ist mit voller Pfadangabe einzugeben, ebenso das gewünschte Archiv-Programm, wobei mit jeweils 30 Zeichen wohl auch das tiefste Hard-Disk-Verzeichnis erreichbar sein müßte. »**DISK-UTILITY**« ist für die Verwendung von »**ZOO**« und »**ARC**« vorbereitet (spätestens jetzt werden Sie hoffentlich gemerkt haben, daß weder »**ZOO**« noch »**ARC**« in »**DISK-UTILITY**« integriert sind. Ob die Befehle »**COPY**«, »**DELETE**«, »**INFO**« und »**INSTALL**« resident in den Speicher geladen werden (nicht bei WB 1.2 zulässig) oder bei Bedarf von der Boot-Diskette geholt werden sollen. Bestimmen Sie durch die Eingabe von »**JA**« oder »**NEIN**« im Eingabefeld »Residente Befehle laden (JA / NEIN)«. Gleiches gilt für die Kommandos »**DISKCOPY**« und »**FORMAT**« in der letzten Zeile des »**DEFAULT**«-Eingabefensters. Nachdem Sie diese letzte Zeile mit <RETURN> verlassen haben, wird die

Datei »DISK__UTILITY.Def« auf den letzten Stand gebracht. War sie noch gar nicht vorhanden, wird sie im »S«-Directory der Boot-Diskette angelegt.

Damit wäre so ziemlich alles über »DISK-UTILITY« gesagt - ganz schön, finden Sie nicht? Und in der kompilierten Version ist sie auch

noch sehr schnell. Apropos kompilieren: Dazu wäre noch zu sagen, der GFA-Basic-Compiler-Aufruf zu »DISK-UTILITY« lautet

```
GFA_bcom Disk-UTIL.GFA -
o Disk-Util.o S& S< F< m300
00 GL # Disk-Util.o -
p Disk-Util
```

Um das Compilat von der Workbench aus starten zu können, müssen Sie ein Project-Icon mit Default auf »DISK-UTILITY« kreieren, wenn Ihr Compiler eine Versionsnummer kleiner »3.5« trägt.

»DISK-UTILITY« wird Ihnen sicher in kürzester Zeit zum unentbehrlichen Helfer werden. pe

Programmname: Disk-Utility		
Computer: A500, A1000, A2000 mit Kickstart 1.2 & 1.3		
Sprache: GFA-Basic		
Bemerkung: siehe Text		
Programmautor: Thomas Rau ----- <pre> 1 dL0 @init 2 lm anfang: 3 RO @main 4 3t > PROCEDURE main 5 wF cl=1 6 Xs DO 7 dP2 x1&=MOUSEX 8 mV y1&=MOUSEY 9 rx IF x1&>279 AND x1&<355 AND y1&>0 AND y1&<230 10 zf4 i&=y1&\10 11 9w ii&=(i&*10)+8 12 sV IF rev!=FALSE 13 8C6 rev!=TRUE 14 v1 COLOR 1 15 bR TEXT 283,ii&,feld\$(i&) 16 kv ia&=i& 17 v0 iia&=ii& 18 vv4 ELSE IF rev!=TRUE AND ia&<>i& 19 5L6 rev!=FALSE 20 7z COLOR 3 21 FL TEXT 283,iia&,feld\$(ia&) 22 5D4 ENDIF 23 eS2 ELSE IF rev!=TRUE 24 AQ4 rev!=FALSE 25 Ga COLOR 3,0 26 KQ TEXT 283,iia&,feld\$(ia&) 27 AI2 ENDIF 28 DZ IF MOUSEK=2 29 4n4 @mauspos 30 jC IF tpos!=TRUE 31 sU6 @ordner 32 zC DEFMOUSE 8 33 G04 ENDIF 34 HP2 ENDIF 35 Mz SLEEP 36 BB0 LOOP 37 xZ RETURN 38 af > PROCEDURE init 39 CO inpfad\$="RAM:" 40 FH DIM sortfolgel(255),zeilen1\$(1),index1%(1),zeilen2\$(1),index2%(1),feld\$(25) 41 Y1 DIM for\$(5),ftext\$(5),schlüssel\$(5),bytes%(2),mbytes%(2),an\$(4),resident\$(5) 42 ph DIM unres\$(5) 43 Ee FOR il=0 TO 254 44 O1 sortfolgel(il)=ASC(UPPER\$(CHR\$(il))) 45 cM NEXT il 46 uL FOR il=0 TO 6 47 Tm READ sonderz\$,wert\$ 48 zz sortfolgel(ASC(sonderz\$))=ASC(wert\$) 49 gQ NEXT il 50 nb DATA ä,A,Ä,A,ä,O,Ö,O,ü,U,Û,U,ß,S 51 PN sortfolgel(ASC("*"))=31 52 iN l\$="" 53 y2 pfad1\$="" 54 55 pfad2\$="" </pre>		<pre> 55 q1 source\$="" Quell-Laufwerk "" 56 ZF dest\$="" Ziel-Laufwerk "" 57 2H co\$="Z00" 58 iv editor\$="SYS:C/ED" 59 cn disk\$=SPACE\$(70) 60 8Z FOR il=1 TO 5 61 ts READ ftext\$(il),schlüssel\$(il),for\$(il) 62 td NEXT il 63 yX DATA "Laufwerk : "," DRIVE ","DF0:" 64 KA DATA "Disk-Name : "," NAME ","LEER-DISK" 65 H4 DATA "NoIcon : "," ","NOICONS" 66 Em DATA "Quick : "," ","" 67 ON DATA "FFS : "," ","" 68 16 anfang1&=1 69 pB anfang2&=1 70 Vz start&=1 71 Y8 t! =1 72 Gn max&=23 73 XZ OPEN "o",#1,"CON:10/15/620/200/ Ausgabe-Fenster < Weiter mit rechter Maustaste > " 74 pB OPENS 1,0,0,640,256,2,&HC000 75 eq SETWPEN 1,3 76 Oz OPENW #1,0,0,640,256,&H208,&H1C0C 77 Oh ON BREAK GOSUB ende 78 Ca ON MENU BUTTON GOSUB mausabf. 79 PK ON MESSAGE GOSUB meldung 80 an ON ERROR GOSUB fehler 81 eT TITLEW #1," Programmiert von Rau Thomas 6652 Bexbach (C) Markt & Technik " 82 aa befehl\$="RAM:DiskUtility" 83 ZJ MKDIR befehl\$ 84 SC r\$="JA" 85 xR b\$="JA" 86 vL SETCOLOR 0,0,0,0 87 ph SETCOLOR 1,12,12,12 88 dM SETCOLOR 2,0,6,14 89 Th SETCOLOR 3,13,0,0 90 Jd COLOR 3,0 91 X1 BOX 260,0,275,198 92 Ha BOX 0,198,275,206 93 hs BOX 360,0,375,198 94 nE BOX 360,198,635,206 95 Ia COLOR 1,0 96 ZR PBOX 262,2,273,10 97 Y6 PBOX 262,188,273,196 98 ga PBOX 362,2,373,10 99 hH PBOX 362,188,373,196 100 rp PBOX 2,199,15,205 101 Hs PBOX 258,199,273,205 102 A5 PBOX 618,199,633,205 103 Dp PBOX 362,199,377,205 104 Un COLOR 2,0 105 uE FOR il=0 TO 220 STEP 10 106 V7 BOX 280,il,355,il+10 107 cM NEXT il 108 bv COLOR 3,0 109 LE LINE 0,231,640,231 110 g7 RESTORE feld 111 9o FOR il=0 TO 22 112 4p READ feld\$(il) 113 iS NEXT il 114 b6 IF EXIST("SYS:S/Disk_Utility.def") 115 Is def!=TRUE 116 Sv OPEN "i",#2,"SYS:S/Disk_Utility.Def" 117 zP FOR il=0 TO 5 118 gA2 INPUT #2,feld\$(il) 119 oYO NEXT il 120 Qr INPUT #2,editor\$ </pre>


```

121 Pd INPUT #2,co$
122 GF INPUT #2,r$
123 jk INPUT #2,b$
124 I6 CLOSE #2
125 mh IF r$="JA"
126 VC2 IF EXIST("SYS:S/Disk_Utility.Bef")
127 314 EXEC "Copy C:Resident to "+befehl$, -1,1
128 Ib EXEC "Copy C:Run to "+befehl$, -1,1
129 KI EXEC "Assign C: Ram:DiskUtility", -1,1
130 Yz OPEN "i", #2, "SYS:S/Disk_Utility.Bef"
131 H1 FOR il=1 TO 5
132 F76 INPUT #2,resident$
133 FW resident$(il)=TRIM$(resident$)
134 U6 iposl=@dateinamen(TRIM$(resident$),0)
135 MJ unres$(il)=MID$(resident$(il),iposl)+" Remove"
136 5p4 NEXT il
137 VJ CLOSE #2
138 Op FOR il=1 TO 5
139 dG6 EXEC "Resident "+resident$(il), -1,1
140 9t4 NEXT il
141 ZN CLOSE #2
142 12 res!=TRUE
143 ud2 ELSE
144 704 ALERT 0, "Befehlsdatei <SYS:S/Disk-Utility.Bef>||nich
t gefunden", 1, "OK", okl
145 v8 res!=FALSE
146 5D2 ENDIF
147 6E0 ENDIF
148 rW IF b$="JA"
149 vm2 EXEC "Copy SYS:System/Format to "+befehl$, -1,1
150 eC EXEC "Copy SYS:System/Diskcopy to "+befehl$, -1,1
151 59 befehle$=befehl$
152 3m0 ELSE
153 Ms2 befehle$="SYS:SYSTEM"
154 DLO ENDIF
155 6p ELSE
156 WE EXEC "Copy C:Resident to "+befehl$, -1,1
157 B4 EXEC "Copy C:Run to "+befehl$, -1,1
158 D1 EXEC "Assign C: Ram:DiskUtility", -1,1
159 nQ EXEC "Resident SYS:C/Copy", -1,1
160 ko EXEC "Resident SYS:C/Assign", -1,1
161 28 EXEC "Resident SYS:C/Delete", -1,1
162 IU EXEC "Resident SYS:C/Info", -1,1
163 fF EXEC "Resident SYS:C/Install", -1,1
164 A1 EXEC "Copy SYS:System/Format to "+befehl$, -1,1
165 tR EXEC "Copy SYS:System/Diskcopy to "+befehl$, -1,1
166 KO befehle$=befehl$
167 QR res!=TRUE
168 Om def!=FALSE
169 Sa ENDIF
170 e3 FOR il=8 TO 231 STEP 10
171 rW TEXT 283,il, feld$(il)
172 ef INC il
173 gQ NEXT il
174 k5 COLOR 3,1
175 mr TEXT 2,214,source$
176 30 TEXT 361,214,dest$
177 cu COLOR 1,0
178 NG feld:
179 NX DATA DFO: , DF1: , DHO: , DH1: , RAM: , R
AD:
180 uW DATA PARENT ,FORMAT ,COPY ,MAKEDIR ,RENAME ,DEL
ETE
181 yq DATA DISKCOPY ,INSTALL ,INFO ,EDITOR ,ALLE ,CLE
AR
182 Rf DATA PRG START,ZOO ADD ,ZOO EXT. ,ZOO LIST ,DEFAULT
183 Jv RETURN
184 dW > PROCEDURE dir_lesen(directory$, VAR zanzahl$, zeilen$(), 1
ndex$())
185 hB mbytes$(tl)=0
186 fO RSET l$=STR$(mbytes$(tl))
187 xH TEXT 78*(5.7*(tl-1)),242,l$
188 4F FILES directory$ TO befeh1$+ "_FILES_"
189 Om OPEN "i", #2, befeh1$+ "_FILES_"
190 10 ERASE zeilen$(), index$()
191 ON anzahl$=LOF(#2)/60
192 RL DIM zeilen$(MAX(0,PRED(anzahl$))), index$(MAX(0,PRED(anzahl
$)))
193 Dk IF anzahl$>0
194 14 RECALL #2, zeilen$(), -1, zanzahl$
195 kT ELSE

```

```

196 Ov zanzahl%=0
197 u2 ENDIF
198 UI CLOSE #2
199 Fo KILL befeh1$+ "_files_"
200 aC RETURN
201 zd > PROCEDURE dir_sort(start$, anfang$, zanzahl$, VAR zeilen$
(), index$())
202 27 QSORT zeilen$() WITH sortfolge1(), -1, index$()
203 rE IF zanzahl%>0
204 81 FOR il=0 TO MIN(zanzahl%-1, max$)
205 qD TEXT start$, (il+1)*8, MID$(zeilen$(il), anfang$, 31)
206 Dx NEXT il
207 4C ENDIF
208 1K RETURN
209 5Z > PROCEDURE mausabf.
210 3Q IF MENU(2)=104
211 Oj @mauspos
212 c2 IF tpos!=TRUE AND tl=1 AND text<zeilen1$
213 3g @makieren(1, anfang1$, zeilen1$, zusatz1$, zeilen1$(), index1$
())
214 OU ELSE IF tpos!=TRUE AND tl=2 AND text<zeilen2$
215 G2 @makieren(378, anfang2$, zeilen2$, zusatz2$, zeilen2$(), index
2$())
216 DL ENDIF
217 V2 IF x$>260 AND x$<275 AND y$<(max$+2)*8-2 AND y$>(max$
)*8
218 vV tl=1
219 uO start$=1
220 IG @scroll_hoch(start$, anfang1$, zeilen1$, zusatz1$, zeilen1$()
, index1$())
221 U9 ELSE IF x$>260 AND x$<275 AND y$<10 AND y$>0
222 zZ tl=1
223 yS start$=1
224 rM @scroll_runter(start$, anfang1$, zeilen1$, zusatz1$, zeilen1$
(), index1$())
225 Ld ELSE IF x$>360 AND x$<375 AND y$<(max$+2)*8-2 AND y$>(
max$)*8
226 8j tl=2
227 Lg start$=378
228 1j @scroll_hoch(start$, anfang2$, zeilen2$, zusatz2$, zeilen2$()
, index2$())
229 JQ ELSE IF x$>360 AND x$<375 AND y$<10 AND y$>0
230 Cn tl=2
231 Pk start$=378
232 Fp @scroll_runter(start$, anfang2$, zeilen2$, zusatz2$, zeilen2$
(), index2$())
233 xj ELSE IF x$>2 AND x$<15 AND y$<204 AND y$>196
234 B1 tl=1
235 Ae start$=1
236 BD @scroll_links(start$, anfang1$, zeilen1$, zusatz1$, zeilen1$()
, index1$())
237 xh ELSE IF x$>258 AND x$<273 AND y$<204 AND y$>196
238 Fp tl=1
239 E1 start$=1
240 g6 @scroll_rechts(start$, anfang1$, zeilen1$, zusatz1$, zeilen1$
(), index1$())
241 td ELSE IF x$>618 AND x$<633 AND y$<204 AND y$>196
242 Oz tl=2
243 bw start$=378
244 OV @scroll_rechts(start$, anfang2$, zeilen2$, zusatz2$, zeilen2$
(), index2$())
245 Ab ELSE IF x$>362 AND x$<377 AND y$<204 AND y$>196
246 S3 tl=2
247 fO start$=378
248 if @scroll_links(start$, anfang2$, zeilen2$, zusatz2$, zeilen2$()
, index2$())
249 OU ELSE IF (y$+2)=>0 AND y$<230 AND x$>279 AND x$<355
250 cS ausw$=(y$+2)\10
251 YQ SELECT ausw$
252 P7 CASE 0 TO 5
253 yZ DEFMOUSE 2
254 h8 IF tl=1
255 o1 altpfad$=pfad1$
256 kR pfad1$=TRIM$(feld$(ausw$))
257 9s @ok(2, pfad1$)
258 Pa ELSE IF tl=2
259 xB altpfad$=pfad2$
260 uX pfad2$=TRIM$(feld$(ausw$))

```

**Listing 1. »DISK-UTILITY«-
Hauptprogramm in GFA-Basic**


```

261 vu @ok(60,pfad2$)
262 x5 ENDIF
263 BC IF ok!=TRUE
264 jw @seite_löschen
265 lp text&=-1
266 Gd ordner!=TRUE
267 gi @ordner
268 A9 text&=tpos&
269 4C ENDIF
270 4w CASE 6
271 Gr DEFMOUSE 2
272 8w text&=-1
273 OR IF tl=1
274 ox @parent(pfad1$)
275 gr ELSE IF tl=2
276 u4 @parent(pfad2$)
277 CK ENDIF
278 Sm IF pfadposl AND ordner!=TRUE
279 yB @seite_löschen
280 tv @ordner
281 GO ENDIF
282 ON text&=tpos&
283 JC CASE 7
284 KO @format
285 NH CASE 8
286 De IF tl=1
287 Y1 @copy(anfang1&,zusatz1&,zeilen1%,pfad1$,pfad2$,zeilen1$()
,index1%())
288 Y3 IF cflag!=TRUE
289 NM2 @ok(60,pfad2$)
290 TX zusatz2&=0
291 Kt @dir_lesen(pfad2$,zeilen2%,zeilen2$(),index2%())
292 qq @dir_sort(378,anfang2&,zeilen2%,zeilen2$(),index2%())
293 Sa0 ENDIF
294 zA ELSE IF tl=2
295 xH @copy(anfang2&,zusatz2&,zeilen2%,pfad2$,pfad1$,zeilen2$()
,index2%())
296 gB IF cflag!=TRUE
297 nW2 @ok(2,pfad1$)
298 Yb zusatz1&=0
299 Ak @dir_lesen(pfad1$,zeilen1%,zeilen1$(),index1%())
300 a0 @dir_sort(1,anfang1&,zeilen1%,zeilen1$(),index1%())
301 ai0 ENDIF
302 bj ENDIF
303 hc CASE 9
304 Vw IF tl=1
305 ar @mkdir(anfang1,zeilen1%,pfad1$,zeilen1$(),index1%())
306 3n IF mkflag!=TRUE
307 xg2 @ok(2,pfad1$)
308 hp0 ENDIF
309 EP ELSE IF tl=2
310 OM @mkdir(anfang2,zeilen2%,pfad2$,zeilen2$(),index2%())
311 8s IF mkflag!=TRUE
312 kj2 @ok(60,pfad2$)
313 mu0 ENDIF
314 nv ENDIF
315 PS CASE 10
316 h8 IF tl=1
317 wH @rename(anfang1&,zusatz1&,pfad1$,zeilen1%,zeilen1$(),inde
x1%())
318 sg @dir_sort(1,anfang1&,zeilen1%,zeilen1$(),index1%())
319 OZ ELSE IF tl=2
320 Sj @rename(anfang2&,zusatz2&,pfad2$,zeilen2%,zeilen2$(),inde
x2%())
321 Jt @dir_sort(378,anfang2&,zeilen2%,zeilen2$(),index2%())
322 v3 ENDIF
323 ae CASE 11
324 pG IF tl=1
325 6e @delete(pfad1$,zeilen1%,zeilen1$(),index1%())
326 ar IF delflag!=TRUE
327 H02 @ok(2,pfad1$)
328 190 ENDIF
329 YJ ELSE IF tl=2
330 My @delete(pfad2$,zeilen2%,zeilen2$(),index2%())
331 fw IF delflag!=TRUE
332 432 @ok(60,pfad2$)
333 6EO ENDIF
334 7F ENDIF
335 pu CASE 12
336 j6 @diskcopy
337 u0 CASE 13
338 Yv @install
339 z6 CASE 14
340 1M @ausführen("INFO")
341 4C CASE 15
342 7Y IF tl=1
343 3i @editor(pfad1$,zeilen1%,zeilen1$(),index1%())
344 ny ELSE IF tl=2
345 Gz @editor(pfad2$,zeilen2%,zeilen2$(),index2%())
346 JR ENDIF
347 DM CASE 16
348 De IF tl=1
349 4B @alle(anfang1&,zeilen1%,zusatz1&,zeilen1$(),index1%())
350 t4 ELSE IF tl=2
351 RT @alle(anfang2&,zeilen2%,zusatz2&,zeilen2$(),index2%())
352 PX ENDIF
353 MW CASE 17
354 Jk IF tl=1
355 VG @clear(anfang1&,zeilen1%,zusatz1&,zeilen1$(),index1%())
356 zA ELSE IF tl=2
357 nd @clear(anfang2&,zeilen2%,zusatz2&,zeilen2$(),index2%())
358 Vd ENDIF
359 Vg CASE 18
360 Pq IF tl=1
361 Z7 @prg_start(anfang1&,zusatz1&,pfad1$,zeilen1%,zeilen1$(),i
ndex1%())
362 5G ELSE IF tl=2
363 xb @prg_start(anfang2&,zusatz2&,pfad2$,zeilen2%,zeilen2$(),i
ndex2%())
364 bj ENDIF
365 eq CASE 19
366 Vw IF tl=1
367 aE @zoo_add(anfang1&,zusatz1&,pfad1$,zeilen1%,zeilen1$(),ind
ex1%())
368 BM ELSE IF tl=2
369 1W @zoo_add(anfang2&,zusatz2&,pfad2$,zeilen2%,zeilen2$(),ind
ex2%())
370 hp ENDIF
371 LP CASE 20
372 b2 IF tl=1
373 yq @zoo_extract(anfang1&,zusatz1&,pfad1$,zeilen1%,zeilen1$()
,index1%())
374 HS ELSE IF tl=2
375 TH @zoo_extract(anfang2&,zusatz2&,pfad2$,zeilen2%,zeilen2$()
,index2%())
376 nv ENDIF
377 UZ CASE 21
378 h8 IF tl=1
379 4m @zoo_list(anfang1&,zusatz1&,pfad1$,zeilen1%,zeilen1$(),in
dex1%())
380 NY ELSE IF tl=2
381 MA @zoo_list(anfang2&,zusatz2&,pfad2$,zeilen2%,zeilen2$(),in
dex2%())
382 t1 ENDIF
383 dj CASE 22
384 km @default
385 JN ENDSELECT
386 x5 ENDIF
387 y6 ENDIF
388 Q5 FRONTS 1
389 ac ActivateWindow(WINDOW(1))
390 ly DEFMOUSE 8
391 fH RETURN
392 AI > PROCEDURE ok(sposl,VAR pfad$)
393 xb pposl=INSTR(pfad$,":")
394 aS chpfad$=LEFT$(pfad$,pposl)
395 J9 IF EXIST(chpfad$)
396 ZR ok!=TRUE
397 h9 CHDIR chpfad$
398 pH frei$=STR$(DFREE(0))
399 P3 CHDIR inpfad$
400 iu MID$(disk$,sposl,8)=SPACE$(8)
401 O2 MID$(disk$,sposl,8)=frei$
402 eJ TITLE# #1,disk$
403 6p ELSE
404 FQ ok!=FALSE
405 B1 pfad$=altpfad$
406 iN FRONTS 1
407 IQ ENDIF
408 wY RETURN
409 E8 > PROCEDURE ordner
410 JL IF tl=1 AND text&<zeilen1%

```



```

411 Am @dirauswahl(1,zusatz1&,pfad1$,zeilen1$())
412 mL IF ordner!=TRUE
413 PS zusatz1&=0
414 Lg anfang1&=1
415 2c @dir_lesen(pfad1$,zeilen1$,zeilen1$(),index1$())
416 m3 @dir_sort(start&,anfang1&,zeilen1$,zeilen1$(),index1$())
417 Sa ENDIF
418 Bt ELSE IF tl=2 AND text< zeilen2%
419 48 @dirauswahl(378,zusatz2&,pfad2$,zeilen2$())
420 uT IF ordner!=TRUE
421 ae zusatz2&=0
422 Ws anfang2&=1
423 S1 @dir_lesen(pfad2$,zeilen2$,zeilen2$(),index2$())
424 9U @dir_sort(start&,anfang2&,zeilen2$,zeilen2$(),index2$())
425 a1 ENDIF
426 bj ENDIF
427 Fr RETURN
428 sv > PROCEDURE mauspos
429 5Q MOUSE x&,y&,k&
430 tx SUB y&,2
431 x4 IF x&<=274 AND k&<>0 AND tl=2
432 Nx tl=1
433 Mq start&=1
434 wH COLOR 3,1
435 y3 TEXT 2,214,source$
436 Fa TEXT 361,214,dest$
437 o6 COLOR 1,0
438 MP ELSE IF x&>361 AND k&<>0 AND tl=1
439 ZA tl=2
440 m7 start&=378
441 30 COLOR 3,1
442 7R TEXT 2,214,dest$
443 WK TEXT 361,214,source$
444 vD COLOR 1,0
445 u2 ENDIF
446 Js tpos=y&/8
447 SO text=y&/8
448 Og IF y&>0 AND y&<SUCC(max&)*8 AND (x&<255 OR x&>378)
449 F1 tpos!=TRUE
450 ra ELSE
451 K9 tpos!=FALSE
452 19 ENDIF
453 fH RETURN
454 eH > PROCEDURE seite_löschen
455 3K COLOR 0,0
456 s1 PBOX start&,1,start&+258,194
457 8Q COLOR 1,0
458 kM RETURN
459 cr > PROCEDURE makieren(start&,anfang&,zanzahl%,zusatz&,VAR
zeilen$(),index$())
460 SI y1&=(tpos&+1)*8
461 9e makier$=MID$(zeilen$(text&+zusatz&),anfang&,32)
462 CF IF index$(text&+zusatz&)=0
463 BR index$(text&+zusatz&)=1
464 Lf COLOR 3,0
465 ni TEXT start&,y1&,makier$
466 HZ COLOR 1,0
467 8r ELSE
468 CR index$(text&+zusatz&)=0
469 Kc COLOR 1,0
470 sn TEXT start&,y1&,makier$
471 KS ENDIF
472 jZ @bytes(zusatz&,zeilen$(),index$())
473 zb RETURN
474 tv > PROCEDURE dirauswahl(start&,zusatz&,VAR pfad$,zeilen$())
475 uN IF tpos!=TRUE
476 Z6 IF LEFT$(zeilen$(text&+zusatz&),1)="*"
477 uv npfad$=pfad$+TRIM$(MID$(zeilen$(text&+zusatz&),2,29))+"/"
478 w4 IF EXIST(npfad$)
479 xN pfad$=npfad$
480 DQ @seite_löschen
481 j6 ordner!=TRUE
482 fG DEFMOUSE 2
483 O7 ELSE
484 mh ordner!=FALSE
485 ze FRONTS 1
486 Zh ENDIF
487 SB ELSE
488 q1 ordner!=FALSE
489 ok ENDIF
490 dL ENDIF
491 j2 COLOR 2,0
492 7a TEXT start&,227,"
493 OJ TEXT start&,227,RIGHT$(pfad$,30)
494 j1 COLOR 1,0
495 Lx RETURN
496 LX > PROCEDURE scroll_hoch(start&,anfang&,zanzahl%,VAR zusat
z$,zeilen$(),index$())
497 OQ sty%=SUCC(max&)*8
498 SO start2&=start&+258
499 T4 WHILE MOUSEK=1
500 4n IF (max&+zusatz&)<PRED(zanzahl%)
501 Ez ADD zusatz&,1
502 mM SCROLL 0,-8,start&,1,start2&,sty&+1
503 cZ COLOR 3`index$(max&+zusatz&),0
504 91 TEXT start&,sty&,MID$(zeilen$(max&+zusatz&),anfang&,32)
505 s0 ENDIF
506 bP WEND
507 X9 RETURN
508 j8 > PROCEDURE scroll_runter(start&,anfang&,zanzahl%,VAR zus
atz$,zeilen$(),index$())
509 Cc sty%=SUCC(max&)*8
510 eC start2&=start&+258
511 fG WHILE MOUSEK=1
512 jW IF zusatz&>0
513 hz SUB zusatz&,1
514 LZ SCROLL 0,8,start&,1,start2&,sty&+1
515 xo COLOR 3`index$(zusatz&),0
516 Zx TEXT start&,8,MID$(zeilen$(zusatz&),anfang&,32)
517 4C ENDIF
518 nb WEND
519 jL RETURN
520 Hs > PROCEDURE scroll_links(start&,VAR anfang&,zanzahl%,zusa
tz$,zeilen$(),index$())
521 Oo sty%=SUCC(max&)*8
522 dx stx&=start&+248
523 rP start2&=start&+258
524 sT WHILE MOUSEK=1
525 V3 IF anfang&+32<60
526 2D ADD anfang&,1
527 bG SCROLL -8,0,start&,1,start2&,sty&+1
528 G2 FOR i&=0 TO MIN(PRED(zanzahl%),max&)
529 y0 COLOR 3`index$(i&+zusatz&),0
530 Lt TEXT stx&,SUCC(i&)*8,MID$(zeilen$(i&+zusatz&),
anfang&+31,1)
531 Ie NEXT i&
532 JR ENDIF
533 2q WEND
534 ya RETURN
535 OR > PROCEDURE scroll_rechts(start&,VAR anfang&,zanzahl%,zus
atz$,zeilen$(),index$())
536 d3 sty%=SUCC(max&)*8
537 sC stx&=start&+248
538 6e start2&=start&+258
539 71 WHILE MOUSEK=1
540 CR IF anfang&>1
541 YG SUB anfang&,1
542 Xp SCROLL 8,0,start&,1,start2&,sty&+1
543 VH FOR i&=0 TO MIN(PRED(zanzahl%),max&)
544 DF COLOR 3`index$(i&+zusatz&),0
545 oB TEXT start&,SUCC(i&)*8,MID$(zeilen$(i&+zusatz&),anfang&,1)
546 Xt NEXT i&
547 Yg ENDIF
548 H5 WEND
549 Dp RETURN
550 mf > PROCEDURE parent(VAR pfad$)
551 F5 IF pfad$<>"
552 Mg pfadposl=@dateinamen(pfad$,1)
553 E7 IF pfadposl
554 5k suchpfad$=LEFT$(pfad$,pfadposl)
555 Zf IF EXIST(suchpfad$)
556 q4 pfad$=suchpfad$
557 xK ordner!=TRUE
558 bK ELSE
559 zu ordner!=FALSE
560 lt ENDIF
561 mu ENDIF
562 nv ENDIF
563 R3 RETURN
564 O2 > PROCEDURE copy(anfang&,zusatz&,zanzahl%,pfad$,zpfad$,VA
R zeilen$(),index$())
565 v2 oflag!=FALSE

```

Listing 1: Fortsetzung


```

566 ll IF pfad$ <> "" AND zpfad$ <> ""
567 2d DEFMOUSE 2
568 am FOR i&=0 TO PRED(zanzahl%)
569 Nm IF index%(i&)=1
570 u7 cflag!=TRUE
571 Ry dname$=TRIM$(MID$(zeilen$(i&),2,29))
572 y2 IF LEFT$(zeilen$(i&),1)="*"
573 QP MKDIR zpfad$+dname$
574 J3 comand$="COPY "+CHR$(34)+pfad$+dname$+CHR$(34)+" TO "+CHR$(
(34)+zpfad$+dname$+CHR$(34)+" ALL"
575 sb ELSE
576 kW comand$="COPY "+CHR$(34)+pfad$+dname$+CHR$(34)+" TO "+CHR$(
(34)+zpfad$+CHR$(34)
577 2A ENDIF
578 Dn IF EXIST(pfad$+dname$)
579 GT EXEC comand$,-1,1
580 5D ENDIF
581 6E ENDIF
582 7T NEXT i&
583 8G ENDIF
584 61 @clear(anfang&,zanzahl%,zusatz&,zeilen$(),index%())
585 nP RETURN
586 WH > PROCEDURE format
587 x4 OPENW #3,150,80,400,120,8,&H1402
588 g3 ON MENU BUTTON GOSUB formatabf
589 xe TITLEW #3,"Formatieren"
590 zZ endflag!=FALSE
591 Xa WHILE INKEY$ <> ""
592 zn WEND
593 Z8 format$=""
594 kB FOR il=1 TO 5
595 Xr PRINT AT(2,il*2);ftext$(il)
596 VF NEXT il
597 nE FOR il=1 TO 5
598 QC LOCATE 16,il*2
599 6K FORM INPUT 30 AS for$(il)
600 qX format$=format$+schlüssel$(il)+TRIM$(for$(il))
601 aK NEXT il
602 T1 COLOR 1,0
603 Fw PBOX 10,90,99,100
604 uS PBOX 135,90,180,100
605 ey COLOR 2,1
606 PA TEXT 28,98,"ABBRUCH"
607 Yo TEXT 150,98,"OK"
608 29 '
609 Mg WHILE endflag!=FALSE
610 dG SLEEP
611 I6 WEND
612 M4 CLOSEW #3
613 pD ON MENU BUTTON GOSUB mausabf.
614 Gs RETURN
615 sY > PROCEDURE formatabf
616 C3 x&=MOUSEX
617 KD y&=MOUSEY
618 KH IF y&>90 AND y&<100
619 na IF x&>10 AND x&<100
620 OZ format$=""
621 wD endflag!=TRUE
622 RE ELSE IF x&>135 AND x&<180
623 st IF format$ <> ""
624 zG endflag!=TRUE
625 A1 comando$=befehle$+"/Format"+format$
626 FB @ausführen(comando$)
627 qY ENDIF
628 rz ENDIF
629 sO ENDIF
630 W8 RETURN
631 MU > PROCEDURE mkdir(anfang&,VAR zeilen%,pfad$,zeilen$(),
index%())
632 zq mkflag!=FALSE
633 ZP IF pfad$ <> ""
634 Zr mkpfad$=pfad$
635 Iw OPENW #3,0,100,640,30,0,&H1402
636 tD TITLEW #3,"MakeDir"
637 xc FORM INPUT 78 AS mkpfad$
638 mU CLOSEW #3
639 2s IF TRIM$(mkpfad$) <> ""
640 7Q mkflag!=TRUE
641 O5 MKDIR TRIM$(mkpfad$)
642 gW @dir_lesen(pfad$,zeilen%,zeilen$(),index%())
643 fe @dir_sort(start&,anfang&,zeilen%,zeilen$(),index%())
644 7F ENDIF

```

```

645 8G ENDIF
646 mO RETURN
647 tg > PROCEDURE rename(anfang&,zusatz&,pfad$,zanzahl%,VAR zeil
len$(),index%())
648 k6 rflag!=FALSE
649 tf FOR i&=0 TO PRED(zanzahl%)
650 g5 IF index%(i&)=1
651 dv aname$=pfad$+TRIM$(MID$(zeilen$(i&),2,29))
652 iA rflag!=TRUE
653 zJ IF aname$ <> befehl$
654 wW @Ändern
655 IQ ENDIF
656 JR ENDIF
657 Kg NEXT i&
658 JT IF rflag!=TRUE
659 JE @clear(anfang&,zanzahl%,zusatz&,zeilen$(),index%())
660 NV ENDIF
661 Id RETURN
662 IR > PROCEDURE Ändern
663 9g neu$=SPACE$(28)
664 Zd neunam$=aname$
665 O0 OPENW #3,0,100,640,30,0,&H1402
666 Qk TITLEW #3,"Rename"
667 Ls FORM INPUT 78 AS neunam$
668 rd neunam$=TRIM$(neunam$)
669 OJ neuname$=neunam$+CHR$(0)
670 sL aname$=aname$+CHR$(0)
671 nz aname%=V:aname$
672 XM neuname%=V:neuname$
673 NR IF EXIST(aname$)
674 LY status%=Rename(aname%,neuname%)
675 ok ENDIF
676 6H IF status% <> 0
677 sk posl=@dateinamen(neunam$,0)
678 rW neunam$=MID$(neunam$,posl+1)
679 UV LSET neu$=neunam$
680 2E MID$(zeilen$(i&),2,29)=neu$
681 iQ ENDIF
682 UC CLOSEW #3
683 Nz RETURN
684 gF > PROCEDURE delete(pfad$,zanzahl%,VAR zeilen$(),index%())
685 Z7 delflag!=FALSE
686 QG IF pfad$ <> ""
687 BH ALERT 0,"Wirklich Löschen",1,"ABBRUCH | OK","okl
688 MI IF okl=1
689 dB delflag!=FALSE
690 3J ELSE IF okl=2
691 2d DEFMOUSE 2
692 aM FOR i&=0 TO PRED(zanzahl%)
693 Nm IF index%(i&)=1
694 AP delflag!=TRUE
695 tq delete$=pfad$+TRIM$(MID$(zeilen$(i&),2,29))
696 XQ IF EXIST(delete$)
697 z3 IF LEFT$(zeilen$(i&),1)="*"
698 dG IF delete$ <> befehl$
699 kO delete2$=CHR$(34)+delete$+"/#?"+CHR$(34)+" ALL"
700 EO EXEC "delete "+delete2$,-1,1
701 4G EXEC "delete "+CHR$(34)+delete$+CHR$(34),-1,1
702 3B ENDIF
703 wF ELSE
704 7J EXEC "delete "+CHR$(34)+delete$+CHR$(34),-1,1
705 6E ENDIF
706 7F ENDIF
707 8G ENDIF
708 9V NEXT i&
709 AI ENDIF
710 m3 IF delflag!=TRUE
711 D1 text&=-1
712 xA @seite_löschen
713 sU @ordner
714 ML text&=tpos&
715 GO ENDIF
716 HP ENDIF
717 vX RETURN
718 uw > PROCEDURE diskcopy
719 zB OPENW #3,160,100,300,30,0,&H1402
720 6I TITLEW #3,"DiskCopy"
721 sk eing$="DF0: TO DF1:"
722 7Q FORM INPUT 30 AS eing$
723 UZ IF TRIM$(eing$) <> ""
724 M8 dcopy$=befehle$+"/DISKCOPY "+TRIM$(eing$)
725 Hd @ausführen(dcopy$)

```



```

726 RZ   ENDIF
727 Dv   CLOSEW #3
728 61   RETURN
729 so   > PROCEDURE install
730 AM   OPENW #3,160,100,300,30,0,&H1402
731 aC   TITLEW #3," Install "
732 cA   eing$="DFO:"
733 Ib   FORM INPUT 30 AS eing$
734 K2   CLOSEW #3
735 g1   IF TRIM$(eing$) <> ""
736 s8   dinstall$="INSTALL "+TRIM$(eing$)
737 zK   @ausföhren(dinstall$)
738 d1   ENDIF
739 Ht   RETURN
740 MU   > PROCEDURE editor(pfad$,zanzahl%,VAR zeilen$( ),index$( ))
741 XH   elflag!=FALSE
742 OA   FOR i&=0 TO PRED(zanzahl%)
743 Ba   IF index$(i&)=1
744 du   IF LEFT$(zeilen$(i&),1) <> "*"
745 bn   elflag!=TRUE
746 sx   edit$=pfad$+TRIM$(MID$(zeilen$(i&),2,29))
747 Qn   IF EXIST(edit$)
748 Gm   editier$=editor$+" "+edit$
749 Bp   FRONTS 0
750 BT   EXEC editier$,-1,1
751 Hw   FRONTS 1
752 rz   ENDIF
753 s0   ENDIF
754 t1   ENDIF
755 uG   NEXT i&
756 oP   IF elflag!=FALSE
757 Jx   FRONTS 0
758 Vy   EXEC editor$,-1,1
759 P4   FRONTS 1
760 ra   ELSE
761 Ip   text&=-1
762 ly   @seite_löschen
763 gI   @ordner
764 A9   text&=tpos&
765 4C   ENDIF
766 iK   RETURN
767 AH   > PROCEDURE alle(anfang&,zanzahl%,zusatz&,VAR zeilen$( ),
index$( ))
768 yL   IF zanzahl%>0
769 EL   ARRAYFILL index$( ),1
770 8c   mbytes$(tl)=0
771 Ic   COLOR 3,0
772 Cy   FOR i&=0 TO MIN(PRED(zanzahl%),max&)
773 Ff   TEXT start&,(i&+1)*8,MID$(zeilen$(i&+zusatz&),anfang&,32)
774 DZ   NEXT i&
775 GY   COLOR 1,0
776 w1   FOR i&=0 TO PRED(zanzahl%)
777 OG   bytes$(tl)=VAL(MID$(zeilen$(i&),31,9))
778 HN   mbytes$(tl)=mbytes$(tl)+bytes$(tl)
779 Ie   NEXT i&
780 Fy   RSET l$=STR$(mbytes$(tl))
781 Xr   TEXT 78*(5.7^(tl-1)),242,1$
782 LT   ENDIF
783 zb   RETURN
784 Ma   > PROCEDURE clear(anfang&,zanzahl%,zusatz&,VAR zeilen$( ),
index$( ))
785 Fc   IF zanzahl%>0
786 Pa   ARRAYFILL index$( ),0
787 Pt   mbytes$(tl)=0
788 T1   COLOR 1,0
789 TF   FOR i&=0 TO MIN(PRED(zanzahl%),max&)
790 Ww   TEXT start&,(i&+1)*8,MID$(zeilen$(i&+zusatz&),anfang&,32)
791 Uq   NEXT i&
792 RA   RSET l$=STR$(mbytes$(tl))
793 j3   TEXT 78*(5.7^(tl-1)),242,1$
794 Xf   ENDIF
795 Bn   RETURN
796 yK   > PROCEDURE prg_start(anfang&,zusatz&,pfad$,zanzahl%,VAR
zeilen$( ),index$( ))
797 H3   FOR i&=0 TO PRED(zanzahl%)
798 4T   IF index$(i&)=1
799 Wn   IF LEFT$(zeilen$(i&),1) <> "*"
800 P7   programm$=pfad$+TRIM$(MID$(zeilen$(i&),2,29))
801 AH   IF EXIST(programm$)
802 2g   FRONTS 0
803 9A   EXEC CHR$(34)+programm$+CHR$(34),-1,1
804 8n   FRONTS 1
805 iq   ENDIF
806 jr   ENDIF
807 ks   ENDIF
808 l7   NEXT i&
809 Je   @clear(anfang&,zanzahl%,zusatz&,zeilen$( ),index$( ))
810 Q2   RETURN
811 iw   > PROCEDURE zoo_add(anfang&,zusatz&,pfad$,zanzahl%,VAR ze
ilen$( ),index$( ))
812 SI   IF pfad$ <> ""
813 RU   apfad$=pfad$
814 dT   zooadd$=" ZOO ADD "
815 cQ   OPENW #3,0,100,640,30,0,&H1402
816 Ox   TITLEW #3,zooadd$
817 Tt   FORM INPUT 78 AS pfad$
818 93   pfad$=TRIM$(pfad$)
819 QC   IF RIGHT$(pfad$,1) <> "/" AND RIGHT$(pfad$,1) <> ":"
820 Ky   FRONTS 0
821 9x   PRINT #1,CHR$(27)+"c"
822 W1   RELSEEK #1,0
823 hT   FOR i&=0 TO PRED(zanzahl%)
824 Ut   IF index$(i&)=1
825 wD   IF LEFT$(zeilen$(i&),1) <> "*"
826 7H   zoocom$=co$+" a "+pfad$+" "+apfad$+TRIM$(MID$(zeilen$(i&),
2,29))
827 gP   EXEC zoocom$,-1,1
828 5D   ENDIF
829 6E   ENDIF
830 7T   NEXT i&
831 ap   @warten
832 aF   FRONTS 1
833 UD   addpl=@dateinamen(pfad$,0)
834 M9   pfad$=LEFT$(pfad$,addpl)
835 CK   ENDIF
836 yg   CLOSEW #3
837 C2   IF pfad$=apfad$
838 G4   text&=-1
839 Vs   ordner!=TRUE
840 vX   @ordner
841 PO   text&=tpos&
842 Bu   ELSE
843 CP   pfad$=apfad$
844 ID   @clear(anfang&,zanzahl%,zusatz&,zeilen$( ),index$( ))
845 MU   ENDIF
846 NV   ENDIF
847 Id   RETURN
848 nV   > PROCEDURE zoo_extract(anfang&,zusatz&,pfad$,zanzahl%,VA
R zeilen$( ),index$( ))
849 jt   IF pfad$ <> ""
850 p0   zdir$="RAM:"
851 jQ   zooext$=" ZOO EXTRACT "
852 Aw   FOR i&=0 TO PRED(zanzahl%)
853 xM   IF index$(i&)=1
854 Pg   IF LEFT$(zeilen$(i&),1) <> "*"
855 Ye   test$=TRIM$(MID$(zeilen$(i&),1,30))
856 ec   IF UPPER$(RIGHT$(test$,3))=RIGHT$(co$,3)
857 JX   PRINT #1,CHR$(27)+"c"
858 6L   RELSEEK #1,0
859 hQ   zoocom$=co$+" x "+pfad$+TRIM$(MID$(zeilen$(i&),2,29))
860 XD   OPENW #3,0,100,640,50,0,&H1402
861 VU   TITLEW #3,zooext$
862 SY   CLS
863 p9   FORM INPUT 78 AS zoocom$
864 wC   FORM INPUT 78 AS zdir$
865 3h   FRONTS 0
866 Cv   CHDIR zdir$
867 K3   EXEC zoocom$,-1,1
868 BQ   @warten
869 Bq   FRONTS 1
870 lt   ENDIF
871 mu   ENDIF
872 nv   ENDIF
873 oA   NEXT i&
874 41   CHDIR inpfad$
875 bJ   CLOSEW #3
876 l1   testp$=UPPER$(pfad$)
877 93   testd$=UPPER$(TRIM$(zdir$))
878 ht   IF testp$=testd$
879 vJ   text&=-1
880 AX   ordner!=TRUE
881 aC   @ordner
882 43   text&=tpos&
883 qZ   ELSE

```

Listing 1: Fortsetzung


```

884 wr @clear(anfang&,zanzahl%,zusatz&,zeilen(),index%())
885 08 ENDIF
886 19 ENDIF
887 fh RETURN
888 4U > PROCEDURE zoo_list(anfang&,zusatz&,pfad$,zanzahl%,VAR
zeilen$,index%())
889 nX IF pfad$ <> ""
890 Uc zoolist$=" ZOO LIST "
891 nZ FOR i&=0 TO PRED(zanzahl%)
892 az IF index%(i&)=1
893 AE test$=TRIM$(MID$(zeilen$(i&),1,30))
894 GE IF UPPER$(RIGHT$(test$,3))=RIGHT$(co$,3)
895 XB FRONTS 0
896 vU zoo$=TRIM$(MID$(zeilen$(i&),2,29))
897 ce zoocom$=co$+" 1 "+TRIM$(pfad$)+zoo$
898 OC PRINT #1,CHR$(27)+"c"
899 10 RELSEEK #1,0
900 ra EXEC zoocom$,-1,1
901 ix @warten
902 HP ENDIF
903 IQ ENDIF
904 Jf NEXT i&
905 1Q FRONTS 1
906 6o CLOSEW #3
907 JE @clear(anfang&,zanzahl%,zusatz&,zeilen(),index%())
908 NV ENDIF
909 1d RETURN
910 UB > PROCEDURE default
911 ro OPENW #3,15,15,400,150,0,&H1402
912 EJ TITLEW #3," Default Parameter "
913 au COLOR 3,0
914 yQ FOR il=1 TO 6
915 IZ laufwerk$="Laufwerk "+STR$(il)+": "
916 0J TEXT 10,SUCC(il)*8-2,laufwerk$
917 gQ NEXT il
918 hJ TEXT 10,9*8-2,"EDITOR : "
919 yA TEXT 10,11*8-2,"ZOO / ARC "
920 JG TEXT 10,13*8-2,"Residente Befehle laden (JA / NEIN)"
921 Mo TEXT 10,15*8-2,"Format - Diskcopy in Ramdisk kopieren"
922 6Y FOR il=1 TO 6
923 UE LOCATE 14,il+1
924 dG FORM INPUT 9 AS feld$(PRED(il))
925 oY NEXT il
926 NO LOCATE 13,9
927 CY FORM INPUT 30 AS editor$
928 SW editor$=TRIM$(editor$)
929 DK '
930 OA LOCATE 13,11
931 C8 FORM INPUT 30 AS co$
932 Pg co$=UPPER$(TRIM$(co$))
933 6I LOCATE 40,13
934 D1 FORM INPUT 5 AS r$
935 7W r$=TRIM$(UPPER$(r$))
936 FT LOCATE 40,15
937 Um FORM INPUT 5 AS b$
938 s1 b$=TRIM$(UPPER$(b$))
939 NU '
940 2X OPEN "o",#2,"SYS:S/DISK_UTILITY.Def"
941 Hh FOR il=0 TO 5
942 No PRINT #2,feld$(il)
943 6q NEXT il
944 7V PRINT #2,editor$
945 6H PRINT #2,co$
946 xt PRINT #2,r$
947 QO PRINT #2,b$
948 aO CLOSE #2
949 nV CLOSEW #3
950 BV COLOR 3,0
951 L1 il=0
952 Gf FOR il=8 TO 231 STEP 10
953 T8 TEXT 283,il,feld$(il)
954 GH INC il
955 I2 NEXT il
956 mO RETURN
957 Ah > PROCEDURE ausfuehren(comando$)
958 YC FRONTS 0
959 NB PRINT #1,CHR$(27)+"c"
960 kz RELSEEK #1,0
961 5t EXEC comando$,-1,1
962 hw @warten
963 hM FRONTS 1
964 uW RETURN
965 8m > PROCEDURE bytes(zusatz&,VAR zeilen$,index%())
966 0O bytes$(tl)=VAL(MID$(zeilen$(text&+zusatz&),31,9))
967 NR IF index$(text&+zusatz&)=1
968 LR mbytes$(tl)=mbytes$(tl)+bytes$(tl)
969 I1 RSET l$=STR$(mbytes$(tl))
970 au TEXT 78*(5.7*(tl-1)),242,l$
971 Om ELSE IF index$(text&+zusatz&)=0
972 Zh mbytes$(tl)=mbytes$(tl)-bytes$(tl)
973 M5 RSET l$=STR$(mbytes$(tl))
974 ey TEXT 78*(5.7*(tl-1)),242,l$
975 Sa ENDIF
976 61 RETURN
977 pi > PROCEDURE warten
978 8c REPEAT
979 f1 UNTIL MOUSEK=2
980 Am RETURN
981 yI > PROCEDURE meldung
982 L6 IF MENU(1)=&H200
983 Qx @ende
984 bj ENDIF
985 Fr RETURN
986 ek > PROCEDURE ende
987 ep IF res!=TRUE AND def!=TRUE
988 6X FOR il=1 TO 5
989 he EXEC "Resident "+unres$(il),-1,1
990 rb NEXT il
991 1e EXEC "SYS:C/Assign c: sys:c",-1,1
992 QE ELSE IF res!=TRUE AND def!=FALSE
993 4V EXEC "Resident Copy Remove",-1,1
994 04 EXEC "Resident Assign Remove",-1,1
995 dM EXEC "Resident Delete Remove",-1,1
996 oO EXEC "Resident Info Remove",-1,1
997 vs EXEC "Resident Install Remove",-1,1
998 81 EXEC "SYS:C/Assign c: sys:c",-1,1
999 qy ENDIF
1000 cU EXEC "Delete RAM:DiskUtility/#?",-1,1
1001 4n EXEC "Delete RAM:DiskUtility",-1,1
1002 9c CLOSE
1003 VB CLOSEW #1
1004 vU CLOSES 1
1005 CL QUIT
1006 aC RETURN
1007 Zp > PROCEDURE fehler
1008 jw IF FATAL=TRUE
1009 LD RESUME ende
1010 tc ELSE
1011 T8 FRONTS 1
1012 Jm CLOSE
1013 pX CLOSEW #3
1014 LL tl=1
1015 kE start&=1
1016 Kf COLOR 3,1
1017 MR TEXT 2,214,source$
1018 dy TEXT 361,214,dest$
1019 CU COLOR 1,0
1020 Ez @clear(anfang1&,zeilen1%,zusatz1&,zeilen1$(),index1%())
1021 xY tl=2
1022 AV start&=378
1023 Rm COLOR 3,1
1024 Vp TEXT 2,214,dest$
1025 ui TEXT 361,214,source$
1026 aQ @clear(anfang2&,zeilen2%,zusatz2&,zeilen2$(),index2%())
1027 vx OPEN "o",#1,"CON:10/15/620/200/ Ausgabe-Fenster
< Weiter mit rechter Maustaste > "
1028 tv ActivateWindow(WINDOW(1))
1029 t6 ON ERROR GOSUB fehler
1030 qr RESUME anfang
1031 MU ENDIF
1032 Oc RETURN
1033 lt > FUNCTION dateinamen(dname$,stlenl)
1034 4e stposl=LEN(dname$)+1-stlenl
1035 pn posl=RINSTR(dname$,"/",stposl)
1036 GH IF posl=0
1037 DM posl=RINSTR(dname$,":",stposl)
1038 Tb ENDIF
1039 ex RETURN posl
1040 tJ ENDFUNC
1041 GA ende:
1042 Nu @ende
(C) 1990 M&T

```

Listing 1: Schluß

PassWd — Paßwortschutz für Disketten

Datenschutz für jedermann

Es ist nicht immer einfach, wichtige Disketten und ihren Datenbestand vor unbefugtem Zugriff zu schützen. »PassWd« kann da schnelle Abhilfe schaffen.

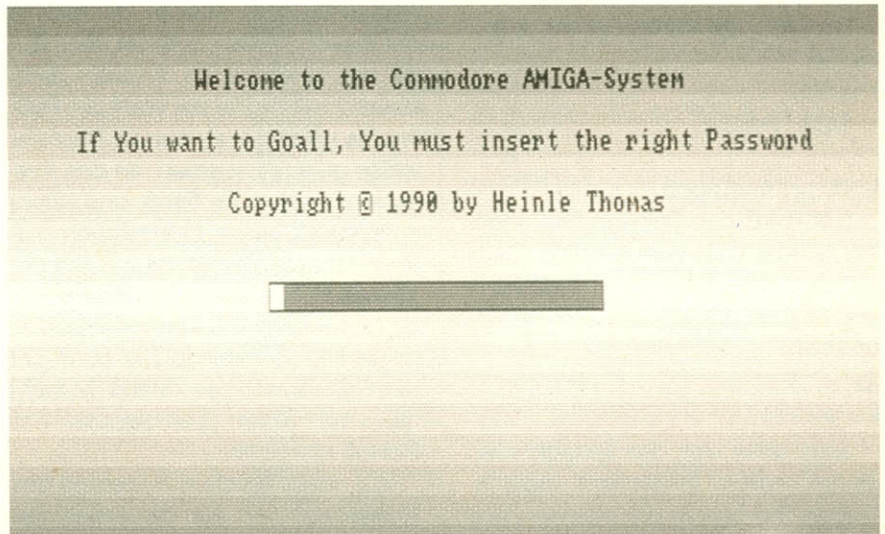


Bild 1. Der gestreifte Bildschirm wartet auf die Paßworteingabe

von Thomas Taubenberger

Das Programm »PassWd« meldet sich beim Booten mit der Aufforderung, ein Paßwort einzugeben, und unterbricht den Bootvorgang so lange, bis das richtige Paßwort eingegeben worden ist (Bild 1).

Das Programm (Listing 1) ist vollständig in »Modula 2« geschrieben und sollte mit dem Checksummer aus AMIGA-Sonderheft Nr. 10 oder AMIGA-Magazin 12/89 abgetippt werden. Anschließend müssen Sie das abgespeicherte ASCII-File mit dem »Modula 2«-Compiler (»M2 Amiga«) übersetzen. Ein lauffähiges Programm erhalten Sie, wenn Sie einen Compiler ab Version 2.0 benutzen. Das Installieren einer lauffähigen Version von »PassWd« auf Diskette oder Festplatte ist denkbar einfach.

Installation auf Diskette

Das compilierte Programm »PassWd« sollte am besten im »S«-Directory der zu schützenden Diskette abgespeichert werden, in dem

sich auch die Startup-Sequence der Diskette befindet. Dies hat zwar für die Funktion des Programms keine Bedeutung, vereinfacht jedoch den Eintrag in der Startup-Sequence und erhöht die Übersichtlichkeit. Als nächstes muß in der Startup-Sequence der Aufruf des Files »PassWd« eingetragen werden. Zunächst sollten Sie von der Workbench-Benutzeroberfläche der zu installierenden Diskette der Command-Line-Interpreter (CLI oder Shell) starten. Nach Erscheinen der Kommandoaufforderung (Promptmeldung) wird in das »S«-Unterverzeichnis gewechselt, in dem sich die Startup-Sequence befindet. Nun wird die Startup-Sequence mit dem Texteditor modifiziert, der sich standardmäßig auf der Workbench-Diskette befindet. Geben Sie dazu folgendes ein:

```
cd s:
ed Startup-Sequence
```

Sie können natürlich auch einen anderen Texteditor verwenden, der das Textfile im ASCII-Format abspei-

chert. Nachdem der Texteditor die Startup-Sequence geladen hat, fügen Sie den Aufruf des Paßwortprogramms »PassWd« in die Startup-Sequence ein:

```
s: PassWd
```

Der Aufruf sollte möglichst nach dem Belegen der Tastatur (»SetMap d«) in die Startup-Sequence eingefügt werden, damit gewährleistet ist, daß eventuell im Paßwort vorhandene Umlaute mit eingegeben werden können. Sie können natürlich das Programm auch in einer beliebigen Schublade »verstauen«. In diesem Fall müssen Sie in der Startup-Sequence den entsprechenden Pfadnamen angeben, z.B.:

```
c:PassWd
```

Zum Verlassen des Texteditors »ED« drücken Sie die ESC-Taste, daraufhin erscheint an der linken unteren Ecke ein »*«-Zeichen. Geben

Sie dann <X> ein, und der Texteditor speichert das geänderte File. So könnte eine geänderte Startup-Sequence aussehen:

```
c:SetPatch >NIL;
Systemroutinen korrigieren
cd c:
echo "A2000 Workbench disk.
Release 1.3 Version 34.20*xn"
SetClock load; Systemzeit aus
Echtzeituhr laden SYS:System/
SetMap d; Deutsche Tastaturbe-
legung
s:PassWd; Paßwortprogramm auf-
rufen
LoadWB
endcli >NIL:
```

Paßwort festlegen

Nun wird das Paßwort-File angelegt. Dazu wird vom CLI die Datei »Secret« mit Hilfe des Texteditors »ED« erstellt. Geben Sie dazu folgendes ein:

```
cd s:
ed Secret
```

Jetzt sollte sich der Texteditor mit einem leerem Bildschirm melden. Geben Sie nun in die oberste Zeile Ihr Paßwort ein. Aber: Bestätigen Sie danach nicht mit >Enter<, da Sie sonst ein »zweizeiliges« Paßwort bekommen, welches bei der Paßwortabfrage nicht mehr eingegeben werden kann, und somit Ihr Computer Ihnen den Zugang zu Ihrer Diskette

bzw. Festplatte verweigert. Speichern Sie anschließend das File ab. (ESC-Taste drücken und nach der Eingabeaufforderung »*« <X> eingeben). In dieser Datei steht nun Ihr Paßwort, das von dem Programm »PassWd« beim Booten von Diskette eingelesen und abgefragt wird.

Beachten Sie bitte, daß diese Datei sich unbedingt im »S«-Verzeichnis befinden und den Namen »Secret« tragen muß, da es sonst zu einem Systemabsturz kommen kann. Darüber hinaus darf das Paßwort, das sich in der Datei »Secret« befindet, die Länge von maximal 26 Zeichen nicht überschreiten. Bei der Wahl des Paßwortes sollte genau auf die Schreibweise geachtet werden, da das Programm neben der Groß- und Kleinschreibung auch Leerzeichen abfragt. Haben Sie z.B. »Markt & Technik« als Paßwort gewählt und am Ende des Wortes ein Leerzeichen eingegeben, so müssen Sie auch bei der Paßwortabfrage »Markt & Technik« mit einem Leerzeichen am Schluß eingeben.

Zur nachträglichen Änderung des Paßwortes muß nur der Inhalt der »Secret«-Datei nach dem oben genannten Schema mit dem Texteditor »ED« behandelt werden.

Installation auf Festplatte

Sie können »PassWd« auch zum Schutz Ihrer Festplatte gegen unbefugte Benutzung verwenden. Dazu

müssen Sie das Programm »PassWd« allerdings etwas modifizieren, da das Programm den Suchweg für die Festplatte nicht selbständig erkennt. In dem fertigen aber noch uncompileden Programm muß in der Programmzeile »DatName:=Open (ADR("df0:s/Secret"),old File);« die Laufwerksbezeichnung »df0:« auf Ihre Festplattenlaufwerksbezeichnung (z.B. »dh0:«) geändert werden. Ein Beispiel:

```
DatName:=Open(ADR("dh0:s/
Secret"),old File);
```

Nach dem Compilieren der Datei verfahren Sie wie bei der Installation auf Diskette. Nun sollte sich Ihr Computer beim Booten mit einem gestreiften Bildschirm melden (die Bildschirmfarbe ist zufällig) und auf die Paßworteingabe warten. Nach der Eingabe des richtigen Codewortes verschwindet der überlagerte Bildschirm und Ihr Amiga fährt mit dem Booten fort. Im Grunde können Sie »PassWd« in jedes beliebige Verzeichnis kopieren, nur muß dann in der Startup-Sequence der entsprechende Pfadname angegeben werden. Das File »Secret« muß aber auf jeden Fall im »S«-Directory enthalten sein.

Wem die Abtipparbeit zuviel ist oder wer keinen »Modula 2«-Compiler besitzt, findet auf der Programm-Service-Diskette beide lauffähigen Versionen. ag

Programmname:	PassWd		
Computer:	A500, A1000, A2000 mit Kickstart 1.2 & 1.3		
Sprache:	Modula 2		
Compiler:	M2 Amiga		
Bemerkung:	siehe Text		

Programmautor: Thomas Heinle <hr/> <pre> 1 4y0 MODULE PassWd; 2 m4 (* 3 aka (P) 1990 by Heinle Thomas / Markt & Technik 4 kd Erstellt auf M2-AMIGA V3.3 von A+L AG./ Schweiz 5 q90 *) 6 pW FROM RandomNumber IMPORT RND,PutSeed; 7 wv FROM SYSTEM IMPORT ADDRESS,ADR; 8 7t FROM Strings IMPORT Delete,Length; 9 aC FROM Str IMPORT Compare,Concat; 10 tC FROM GfxMacros IMPORT OnDisplay,OffDisplay,OffSprite; 11 JH FROM Dos IMPORT Delay,oldFile,FileHandlePtr,Read,Open,Close; 12 GL FROM Graphics IMPORT RastPortPtr,ViewPort,SetAPen,SetBPen,Move,Text, 13 wyP SetRGB4,RectFill,jam1,ViewModes,ViewModeSet;</pre>	<pre> 14 UX0 FROM Intuition IMPORT IDCMPFlags,IDCMPFlagSet,WindowFlags,WindowFlagSet, 15 g1P WindowPtr,CloseWindow,ScreenPtr,CloseScreen,Gadget, 16 2M ActivateGadget,Border,StringInfo,AddGadget,OnGadget, 17 Tn RemoveGadget,strGadget,GadgetFlagSet,GadgetFlags, 18 LO ActivationFlags,ActivationFlagSet,CurrentTime, 19 dd OpenScreen,customScreen,NewScreen,OpenWindow,NewWindow; 20 nW 21 ux0 VAR ns : NewScreen; 22 eS5 nw : NewWindow; 23 aG Info : StringInfo; 24 Ia Setze : BOOLEAN; 25 Sh Rahmen : Border; 26 eS Schirm : ScreenPtr; 27 gX Viewer : ViewPort; 28 cb Raster : RastPortPtr; 29 1i xyFeld : ARRAY [0..9] OF INTEGER; 30 Rd Fenster : WindowPtr; 31 OJ DatName : FileHandlePtr; 32 Q6 Erg,Leser : LONGINT; 33 IQ StringGadget : Gadget; 34 53 Info1,Info2,Info3,Info4 : ARRAY [0..55] OF CHAR; 35 sW Buffer,UndoBuffer,SollWort,Ze : ARRAY [0..26] OF CHAR; 36 SJ rnR,rnG,rnB,R,G,B,z,r,g,b,l,Dummy,Mic : INTEGER;</pre>
---	---


```

37 n00 PROCEDURE Zufall;
38 mF BEGIN
39 EE2 LOOP
40 Lq4 CurrentTime(ADR(Dummy),ADR(Mic));
41 bu PutSeed(Mic);
42 eb Dummy := RND(2);
43 JI Dummy := 0;
44 2G rnR := RND(2);INC(Dummy,rnR);
45 SK rnG := RND(2);INC(Dummy,rnG);
46 kS rnB := RND(2);INC(Dummy,rnB);
47 AN IF Dummy > 0 THEN
48 806 EXIT;
49 i44 END;
50 j52 END;
51 rLO END Zufall;
52 OT BEGIN
53 BM2 WITH ns DO
54 244 leftEdge := 0;
55 Cp topEdge := 0;
56 Xf width := 640;
57 DJ height := 256;
58 8e detailPen := 0;
59 u7 blockPen := 1;
60 lX depth := 4;
61 r1 type := customScreen;
62 yu viewModes := ViewModeSet[hires];
63 qm font := NIL;
64 AV defaultTitle := NIL;
65 wX gadgets := NIL;
66 zi customBitMap := NIL;
67 OM2 END;
68 RN Schirm := OpenScreen(ns);
69 ds WITH nw DO
70 IK4 leftEdge := 0;
71 S5 topEdge := 0;
72 nv width := 640;
73 TZ height := 256;
74 Ou detailPen := 0;
75 AN blockPen := 1;
76 v1 idcmpFlags := IDCMPFlagSet {gadgetUp,gadgetDown};
77 kq flags := WindowFlagSet{activate,borderless};
78 OI firstGadget := NIL;
79 uJ checkMark := NIL;
80 hH title := NIL;
81 xW screen := Schirm;
82 qB bitMap := NIL;
83 DN type := customScreen;
84 Hd2 END;
85 I5 Fenster := OpenWindow(nw);
86 81 Raster := Fenster.rPort;
87 aW Viewer := Schirm.viewPort;
88 A1 z := 3;
89 p6 Buffer := "";
90 pn UndoBuffer := "";
91 u0 Ze := CHR(169);
92 Fo Info1 := "Copyright ";
93 Mj Info2 := " 1990 by Heinle Thomas";
94 pn Info3 := "Welcome to the Commodore AMIGA-System";
95 g2 Info4 := "If You want to Goall, You must insert the
right Password";
96 hr DatName := Open(ADR("df0:s/Secret"),oldFile);
97 dT Lese := Read(DatName,ADR(SollWort),SIZE(SollWort));
98 tn Close(DatName);
99 zG l := Length(SollWort);
100 ay Delete(SollWort,l-1,5);
101 Dt Concat(Info1,Ze);
102 l3 Concat(Info1,Info2);
103 E1 OffDisplay;
104 jd OffSprite;
105 Yn Zufall;
106 Cy FOR r := 2 TO 14 DO
107 eG4 IF rnR = 1 THEN R := r;END;
108 m2 IF rnG = 1 THEN G := r;END;
109 OU IF rnB = 1 THEN B := r;END;
110 4v SetRGB4(ADR(Viewer),r,R,G,B);
111 8E SetAPen(Raster,r);
112 H2 RectFill(Raster,0,z,640,z+10);
113 Zz INC(z,10);
114 l72 END;
115 7U FOR r := 13 TO 2 BY -1 DO
116 nP4 IF rnR = 1 THEN R := r;END;

```

```

117 vB IF rnG = 1 THEN G := r;END;
118 Xd IF rnB = 1 THEN B := r;END;
119 D4 SetRGB4(ADR(Viewer),r,R,G,B);
120 HN SetAPen(Raster,r);
121 QB RectFill(Raster,0,z,640,z+10);
122 l8 INC(z,10);
123 uG2 END;
124 nA SetAPen(Raster,14);
125 Xh SetBPen(Raster,7);
126 Jj Move(Raster,170,60);
127 oJ Text(Raster,ADR(Info3),Length(Info3));
128 eq SetBPen(Raster,9);
129 ob Move(Raster,95,80);
130 zr Text(Raster,ADR(Info4),Length(Info4));
131 qB SetBPen(Raster,11);
132 l6 Move(Raster,190,100);
133 ef Text(Raster,ADR(Info1),Length(Info1));
134 Qr WITH Rahmen DO
135 A14 leftEdge := -1;
136 KT topEdge := -1;
137 tD frontPen := 2;
138 TJ backPen := 0;
139 sE drawMode := jaml;
140 GU count := 5;
141 Fx xy := ADR(xyFeld);
142 4C nextBorder := NIL;
143 Ea2 END;
144 H4 xyFeld[0] := 0; xyFeld[1] := 0;
145 RK xyFeld[2] := 201; xyFeld[3] := 0;
146 5b xyFeld[4] := 201; xyFeld[5] := 9;
147 LT xyFeld[6] := 0; xyFeld[7] := 9;
148 7A xyFeld[8] := 0; xyFeld[9] := 0;
149 Ar WITH Info DO
150 zR4 buffer := ADR(Buffer);
151 SY undoBuffer := ADR(UndoBuffer);
152 q5 bufferPos := 0;
153 sj maxChars := 26;
154 mU dispPos := 0;
155 Qm2 END;
156 ln WITH StringGadget DO
157 Dk4 nextGadget := NIL;
158 Ty leftEdge := 215;
159 Ze topEdge := 124;
160 ll width := 200;
161 Br height := 8;
162 6e flags := GadgetFlagSet{};
163 Pw activation := ActivationFlagSet{gadgImmediate,relVeri
fy,toggleSelect};
164 E9 gadgetType := strGadget;
165 9J gadgetRender := ADR(Rahmen);
166 EC selectRender := NIL;
167 ly gadgetText := NIL;
168 Zl specialInfo := ADR(Info);
169 OD gadgetID := 0;
170 LX userData := NIL;
171 g22 END;
172 ze IF rnR = 1 THEN R := 15;END;
173 70 IF rnG = 1 THEN G := 15;END;
174 jq IF rnB = 1 THEN B := 15;END;
175 2B SetRGB4(ADR(Viewer),15,R,G,B);
176 J7 SetRGB4(ADR(Viewer), 1,R,G,B);
177 OI Dummy := AddGadget(Fenster,ADR(StringGadget),-1);
178 9G OnGadget(ADR(StringGadget),Fenster,NIL);
179 fL OnDisplay;
180 VV LOOP
181 Gz4 Erg := Compare(SollWort,Buffer);
182 nD Setze := ActivateGadget(ADR(StringGadget),Fenster,NIL);
183 HY IF Erg = 0 THEN
184 Ka6 EXIT;
185 uG4 END;
186 vH2 END;
187 aW SetRGB4(ADR(Viewer),15,0,0,0);
188 n6 Delay(20);
189 c6 OffDisplay;
190 qb Dummy := RemoveGadget(Fenster,ADR(StringGadget));
191 8c CloseWindow(Fenster);
192 ze CloseScreen(Schirm);
193 sB Delay(20);
194 SJO END PassWd.

```

**Listing 1. Nach dem Abtippen
muß das Programm mit dem »M2
Amiga« compiliert werden**

(C) 1990 M&T

Mit Volldampf voraus

Bevor Sie zum LötKolben greifen, lesen Sie sich zumindest diesen Artikel genau durch. Eine gewisse praktische Löt-erfahrung sollten Sie nämlich schon mitbringen, da auf der PC-Brückenkarte Lötunkte zu setzen sind und der Quarz »QU1« (siehe Bild 1) ausgelötet werden muß. Um auf einen speziellen Quarz-Oszillator zu verzichten, benutzen wir den vorhande-

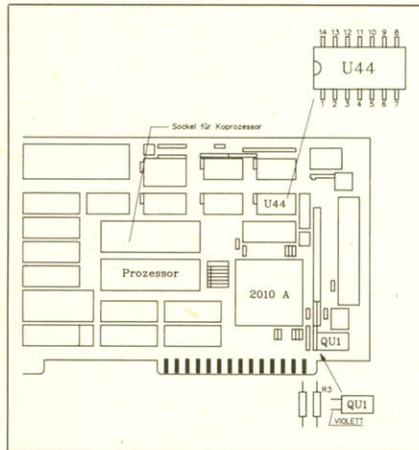


Bild 1. Der Quarz »QU1« ist auf der PC-Karte leicht zu finden

nen Quarz (»QU1« 14,31818 MHz) der XT-Karte. Die Quarzanschlüsse des »QU1« führen auf den Oszillatoreingang des Custom-Chip »FARADAY FE 2010«. In diesem Chip wird die Frequenz durch drei geteilt und ergibt den Prozessortakt von 4,77 MHz. Außerdem enthält der Faraday-Chip mehrere TIMER (Systemuhren). Diese Timer lassen sich nur gemeinsam beschleunigen. Auch der Disketten-Controller erhält hier seinen Takt. So ist es nicht ohne weiteres möglich, nur den Quarz »QU1« gegen einen anderen mit höherer Frequenz (24 MHz / 3 = 8 MHz) auszutauschen: Durch die Takterhöhung wäre kein Disketten-Laufwerkzugriff mehr möglich. Hier greift nun die High-Speed-Platine ein. Beim Diskettenzugriff wird automatisch

Besitzer der A 2088 PC-Brückenkarte von Commodore werden sich nicht selten über deren langsame Taktfrequenz geärgert haben. Mit der High-Speed-Platine haben sie die Möglichkeit, die Taktfrequenz der XT-Karte ohne großen Aufwand auf 8 MHz zu erhöhen.

von Wolfgang Fiebig

Rechengeschwindigkeit benutzen, und diese Uhr nun auch schneller läuft, läßt sich der Tempozuwachs nur von Hand messen.

Soll der Prozessor »NEC V20-8« (8-MHz-Version) auf der XT-Karte betrieben werden, muß der Faraday-Chip »FE 2010-A« vorhanden sein. Mit der älteren Version »FE 2010« gibt es Timing Probleme. Es hat sich aber auch gezeigt, daß der 8088 5 MHz-Prozessor mit der hohen Taktfrequenz zurecht kommt.

Sollten wider Erwarten beim Kaltstart des PCs RAM-Fehler gemeldet werden, hat der Faraday-Chip Timing-Probleme. Sie sollten dann den 24 MHz Quarz-Oszillator auf der

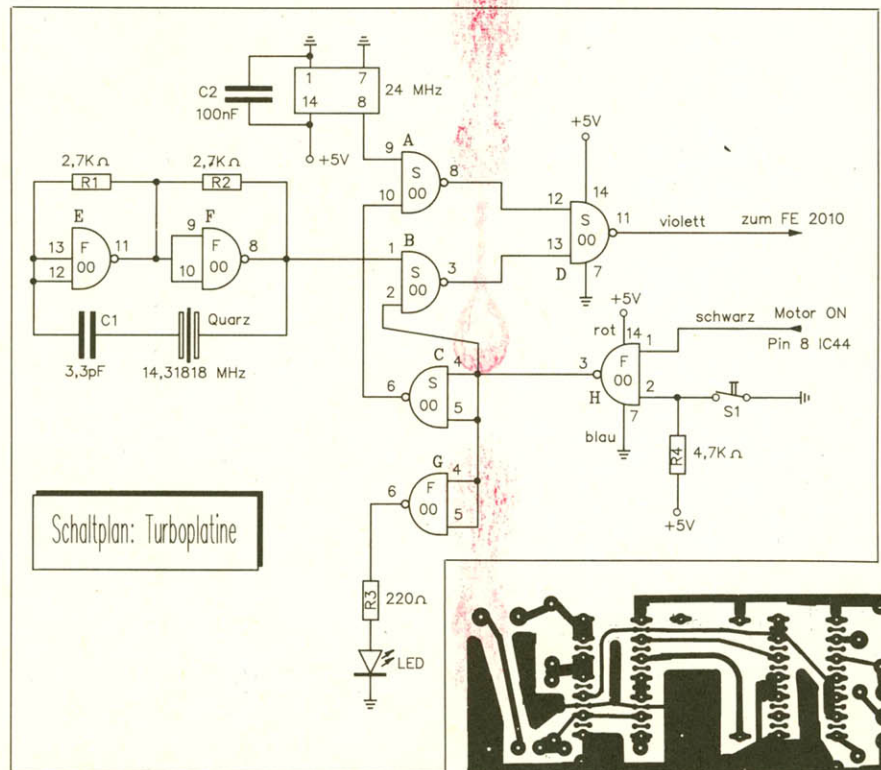


Bild 2. Das Schaltbild der High-Speed-Platine

auf den normalen Takt von 4,77 MHz zurückgeschaltet. Nach Beendigung des Laufwerkzugriffs schaltet die Platine wieder auf 8 MHz um, jedoch wird bei Festplatten im Speed-Modus eine erhöhte Datenübertragungsrate erreicht. Da Testprogramme die Systemuhr zum Ermitteln der

Bild 3. Das Platinenlayout ist 1:1 spiegelverkehrt abgebildet

High-Speed-Platine gegen einen 22 MHz Quarz-Oszillator austauschen. Dann wird der PC zwar nur noch mit 7 MHz getaktet, aber es hilft. Natürlich kann auch der mathematische Coprozessor 8087-8 eingesetzt werden, der ebenfalls mit 8 MHz getaktet ist.

Beschreibung der Schaltung

Der Kondensator »C1« 3,3pF in Reihe mit dem Quarz »QU1« (14,31818 MHz) geschaltet, bildet zusammen mit den Nandgattern »E« und »F« (1/2 74F00) einen freischwingenden Oszillator (siehe Bild 2). Der Ausgang des Oszillators »PIN 8« geht auf den Eingang »PIN 1« vom Nandgatter »B«. Der Ausgang »PIN 8« des Quarz-Oszillators (24 MHz) geht auf den Eingang »PIN 9« vom Nand »A« (1/4 74S00). Nandgatter »C« bewirkt die Umschaltung zwischen den Gattern »A« und »B«. So gelangt nur immer eine der beiden Quarzfrequenzen auf den Ausgang.

»LOW«. So ist eine manuelle Umschaltung zwischen den beiden Taktfrequenzen möglich. Das Nandgatter »G« schaltet die LED (= Leuchtdiode) bei 8 MHz Takt ein und 4,77 MHz aus.

Zusammenbau der High-Speed-Platine:

Die Platine kann nach dem abgedruckten Layout hergestellt werden (siehe Bild 3). Zuerst werden die beiden kleinen Drahtbrücken, die unter den ICs sitzen, eingelötet (siehe Bild 4). Dann folgen die Widerstände, Kondensatoren und die beiden Quarze. Den Quarz 14,31818 MHz zuvor von der XT-Karte »A 2088« vor-

durchgeführt werden. Einen Frequenzzähler oder ein Oszilloskop an die Taktleitung (violett) anschließen, die Schaltung mit 5 Volt versorgen (Blau = GND, Rot = +5 Volt).

Bei offener »MOTOR-ON«-Leitung (schwarzer Draht) und offenem Schalter »S1« müssen 24 MHz anliegen und die LED leuchten. Schalter »S1« geschlossen oder die »MOTOR-ON«-Leitung auf »GND« gelegt, LED aus 14,31818 MHz. Wenn die Funktionsprüfung erfolgreich war, kann mit der Montage der High-Speed-Platine auf der PC-Brückenkarte begonnen werden.

Stückliste:

R1	= 2,7 K
R2	= 2,7 K
R3	= 220 Ω
R4	= 4,7 K
C1	= 3,3 pF
C2	= 100 nF
Q1	= 24 MHz (Quarz-Oszillator)
IC1	= 74S00 (Nandgatter A;B;C;D)
IC2	= 74F00 (Nandgatter E;F;G;H)

LED mit Fassung
1poliger Ausschalter
Platine
Lötnagel
diverse Leitungen

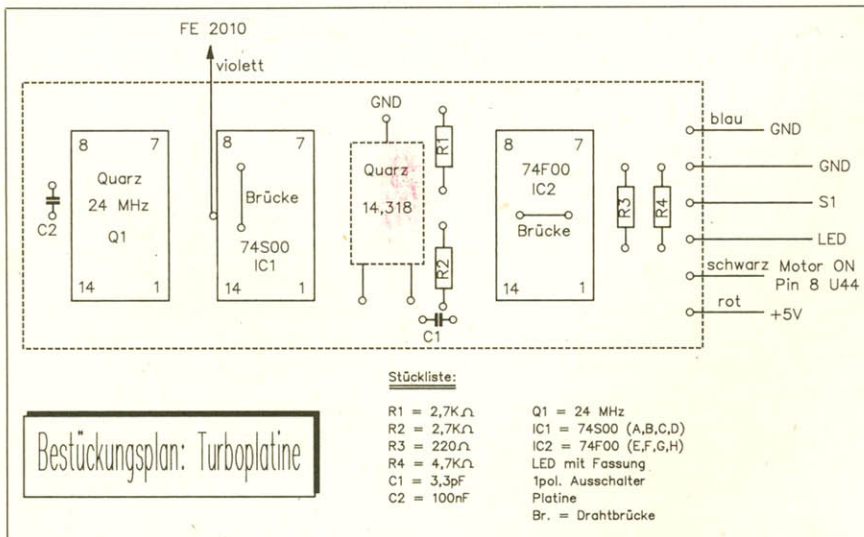


Bild 4. Der Bestückungsplan zeigt auch wo Lötbrücken gesetzt werden

Das Nand »D« führt beide Nandgatter »A« und »B« zusammen. Der Ausgang »PIN 11« wird mit einer möglichst kurzen Leitung (violett) mit dem Oszillatoreingang des »FARADY FE 2010« verbunden.

Für die Umschaltung beim Diskettenzugriff ist die »MOTOR-ON«-Leitung des Floppy-Controllers bestens geeignet. Wird der Pegel »LOW« (= niedrig, Spannung 0 Volt) schaltet das Nandgatter »H« (»PIN 1«) und es erfolgt die Umschaltung auf die niedrige Taktfrequenz.

Nach Beendigung des Diskettenzugriffs wird »PIN 1« vom Nand »H« wieder »High« (= Hoch, Spannung zwischen 4 und 5 Volt) und die Gatter schalten auf den hohen Takt zurück. Das gleiche wird erreicht, wenn der Schalter »S1« auf »GND« (= Masse) schaltet, »PIN 2« von Nand »H« wird

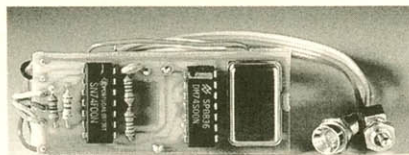


Bild 5. Auf der High-Speed-Platine sieht man die Lötnägel zum Befestigen des PC-Quarzes »QU1«

sichtlich auslöten. Dieser Quarz wird auf der gegenüberliegenden Anschluß-Seite mit der Masse der Speed-Karte verbunden (siehe Bild 5). Zum Schluß werden die beiden ICs eingelötet. Die Taktleitung zum »FARADY FE 2010« wird direkt an IC 74S00 »PIN 11« angelötet.

Vor dem Einbau sollte ein Funktionstest der High-Speed-Platine

Hinweise:

Vor dem Einbau ist zu bedenken, daß Änderungen an der XT-Karte zum Verlust der Garantieansprüche führen. Die meisten Leitungen für unsere Platine können wir am IC U44 (siehe Bild 1) auf der XT-Karte anlöten (Rot = +5 Volt an PIN 14 von IC U44, blau = GND an PIN 7 von IC U44, schwarz = MOTOR-ON an PIN 8 von IC U44).

Die Leitungen werden direkt an die Beinchen vom IC gelötet. Die Taktleitung (violett) löten wir an den unteren der beiden PINs des ausgelöteten Quarzes »QU1«. Zum Befestigen der Platine hat sich doppelseitiges Klebeband gut bewährt. Damit kann die Platine oberhalb vom IC U44 auf die ICs der XT-Karte aufgeklebt werden. Der Schalter S1 und die LED lassen sich leicht ins LED-Panel des Amiga 2000 einbauen.

Nach dem Einstecken der XT-Karte in den A2000 steht einem Funktionstest nichts mehr im Wege. Wir wünschen Ihnen viel Spaß mit Ihrer High-Speed-PC-Karte. pe

Ein paar Bytes mehr

Von Amiga-Anwendern und Fachwelt unbemerkt hat Commodore die Mutterplatine des Amiga 500 neu gestaltet. Sie präsentiert sich dem Betrachter jetzt erstaunlich aufgeräumt: Hochintegrierte Superchips übernehmen die Aufgaben vieler kleinerer Bausteine. Unser Augenmerk gilt aber der neuen Architektur des Hauptspeichers, denn wir wollen mit geringem Aufwand den Hauptspeicher des Amiga 500 von 512 KByte auf 1 MByte erweitern.

In den älteren Amiga 500 bildeten 16 jeweils 16polige DRAM (Dynamic Random Access Memory = dynamische Speicherbausteine) des Typs 41256 die Grundstufe des Hauptspeichers, wobei ein DRAM dieses Typs das Speichervermögen von 1 x 256 KBits, also 32 KByte, hat. Auf der neuen Mutterplatine benötigt man für die 512 KByte nur noch vier dynamische Speicherbausteine des 18poligen Typs 514256. In jedem Speicherchip können 4 x 256 KBit, das sind 128 KByte, geschrieben und wieder ausgelesen werden.

Die neue Mutterplatine des Amiga 500 bietet neben jedem DRAM einen freien Einbauplatz, insgesamt noch vier weitere Plätze. Hier kann der Amiga 500 sehr einfach intern auf das doppelte Speichervermögen, also auf 1 MByte, erweitert werden.

Der Einbau

Für die interne Speichererweiterungen auf 1 MByte benötigen wir nur

- 4 18polige Steckfassungen,
- 4 DRAM 514256,
- einen einpoligen Miniaturschalter und
- etwas zweiadrigen Schaltdraht.

Die Gesamtkosten in Höhe von weniger als 130 Mark - hauptsächlich für die Speicherbausteine - sind also außerordentlich gering.

Bevor wir mit den Arbeiten beginnen, wollen wir berücksichtigen, daß

Alle reden von Abrüstung, wir nicht. 1 MByte Speicher für den A500 on board - eine kühne Forderung, die sich aber erfüllen läßt.

**von Norbert Paul
und Reiner Moll**

die hochintegrierten elektronischen Bausteine des Amiga überwiegend in der CMOS-(Complementary Metal Oxide Semiconductor-)Technologie gefertigt sind. Schon geringe Überspannungen, erst recht die hohen Spannungen durch statische Aufladungen, können diese Bausteine zerstören. Ein CMOS-gerechter Arbeitsplatz sollte aus einer leitenden Unterlage bestehen. Die Handgelenke verbinde man — z.B. am metallenen Uhrenarmband — leitend mit dieser Unterlage. Unerlässlich ist auch ein galvanisch vom Starkstromnetz getrennter und dadurch potentialfreier LötKolben.

Nun, der Einbau gestaltet sich sehr einfach. Zuerst nehmen wir die Diskette aus dem internen Laufwerk. Dann ziehen wir alle am Computer befestigten Stecker ab. Nun legen wir den Amiga mit der Tastatur nach unten auf eine weiche Unterlage. Jetzt werden die drei Schrauben an der Vorderkante des Gehäuses entfernt (Vorsicht! Die mittlere Schraube befindet sich unter einem Siegel = Garantieverlust). Dann die drei Schrauben an der hinteren Kante des Gehäuses entfernen, A500 umdrehen und das Gehäuseoberteil abheben. Im Innern des Amiga müssen wir den Stecker des Tastaturkabels abziehen und können die Tastatur nach hinten wegziehen. (Achtung! Unter der Tastatur befindet sich eine Isolierfolie, die beim Einbau nicht vergessen werden darf.) Die Metall-Abschirmung läßt sich abnehmen, wenn die Laschen aufgebogen und die beiden Befestigungsschrauben

an der Vorderkante und am Expansion-Port entfernt sind. Nach dem Abheben der Abschirmung sieht man die Mutterplatine. Über ihr ist das Disketten-Laufwerk angebracht. Dieses läßt sich nach dem Abziehen der Stromversorgung und des Datenkabels sowie dem Entfernen der drei Befestigungsschrauben leicht herausheben. So, nun trennt uns nur noch das untere Abschirmblech von der Lötseite der Mutterplatine. Um dieses nach unten abziehen zu können, müssen alle Befestigungsbolzen der SubD-Anschlüsse auf der Rückseite entfernt werden.

Nun kommen wir zum eigentlichen Umbau. Die IC-Fassungen werden von oben in die freien Löcher der Steckplätze U20 bis U23 eingepaßt und anschließend auf der Rückseite verlötet. In die freien Fassungen setzen wir nun die DRAM-Chips ein. Jetzt wäre der Amiga 500 in der Lage, mit 1024 KByte zu arbeiten. Da es aber viele ältere Programme gibt, die nur mit dem »kleinen« 512-KByte-Hauptspeicher arbeiten, bauen wir noch einen Schalter ein, der die Erweiterung abschaltet. Für die RAM-Umschaltung ist der Lötjumper »JP7A« neben der Stiftleiste für eine externe RAM-Karte verantwortlich. Die beiden unteren Lötunkte sind leitend verbunden. Über diesen Jumper fragt der Amiga beim Einschalten ab, ob eine Speichererweiterung vorhanden ist. Anstelle der festen Brücke löten wir den Schalter zwischen dem mittleren und dem oberen Lötunkt. Ist der Schalter geschlossen, liegt der mittlere Lötunkt an Masse und der Amiga erkennt das extra RAM. Bei offenem Schalter dagegen werden die zusätzlichen RAM-Bausteine nicht erkannt, und der Amiga meldet sich mit seiner ursprünglichen Hauptspeicher-Ausstattung von 512 KByte.

Leider bestehen gewisse Einschränkungen abhängig von der Bauserie des Amiga 500. An den älteren Amiga 500 der Baureihe Rev. 5 (bis April 1989 gefertigt und ausgeliefert) - zu erkennen an den 16 Speicherbausteinen des Typs 51256 anstelle der vier 514256-DRAMs - ist unsere interne Speichererweiterung nicht möglich. Man ist auf die externe Speichererweiterungskarte angewiesen. Außerdem kann nach diesem Umbau der Speicher nur noch über den Expansion-Port erweitert werden, der interne Steckplatz ist nicht mehr nutzbar.

Ein anderes Problem tritt bei Computern auf, die noch nicht mit den neuen Big Agnus 8372 bestückt sind. Denn wie in allen Personal- und Homecomputern werden im Amiga die herstellungstechnisch weniger aufwendigen und daher preiswerten DRAM-Bausteine verwendet. In deren dynamischen Speicherzellen wird die Information jeweils als winzige Ladung in Kondensatoren gespeichert. Kondensatorladungen bauen sich über sog. Leckströme allmäh-

lich ab - auch bei bester Isolierung. Damit die Informationen (= Kondensatorladungen) nicht verloren gehen, müssen sie in »Refresh« (= Auffrischungs-)Läufen regeneriert werden. Im neueren Amiga 500 der Rev. 7 (etwa seit April 1989) ist für den »Refresh« der Hochintegrationsbaustein Big Agnus 8372 zuständig. Die gleiche Aufgabe erfüllte in den etwas älteren A500 - auch in denen mit den neuen 4-DRAM-514256-Chips (Amiga 500-Rev.6A seit ca. Januar 1989) - der Fat Agnus 8371, jedoch nur bis zu 512 KByte. Der Umbau eines solchen Modells bleibt somit dem Profi vorbehalten, da der »Refresh« für die zusätzlichen 512 KByte mit einer Zusatzplatine durchgeführt werden muß. Oder sie rüsten Ihren Computer mit dem Big Agnus 8372 aus. Wie das geht, lesen Sie im AMIGA-Magazin 11/89, Seite 45.

Der neuere Amiga 500 mit dem Big Agnus 8372 und unserer internen Speichererweiterung verfügt ausschließlich über 1 MByte ChipMEM und kein FastMEM. Dies kann bei einigen Anwendungsprogrammen

und Spielen, die gezielt auf das FastMEM zugreifen, zu Abstürzen führen. Beheben läßt es sich durch eine Speichererweiterung am Expansion-Port, oder Sie fragen beim Händler, ob es von diesem Programm eine spezielle 1-MByte-ChipRAM-Version gibt. Vorteile hat diese Art von Speicher bei Grafik- und Musiksoftware, denn jetzt ist es kein Problem mehr, mit Deluxe Paint Bilder in Hires Overscan 16 Farben zu laden.

Ausblick:

Die erweiterte Anschlußarchitektur der neuen DRAM-Bausteine ermöglicht die Verwendung der 4-MBit-Chips. Zur Speichererweiterung über 1 MByte hinaus auf 2,5 MByte müßte man nur die 1-MByte-Chips gegen die neuen 4-MByte-Chips austauschen. Allerdings müßte der »Refresh« der Speicherbereiche oberhalb von 1 MByte über gesonderte Ansteuerungsschaltungen besorgt werden. Mögliche Weiterentwicklungen der Großintegrationsbausteine Fat Agnus usw. könnten auch dieses Problem lösen. *pe*

Rohlaufwerkanpassung für den Amiga

Selbst ist der Mann

Ein zweites Disketten-Laufwerk erleichtert das Arbeiten mit dem Amiga. Das AMIGA-Sonderheft zeigt, wie Sie mit wenig Aufwand selbst ein Zusatzlaufwerk bauen.

von Gerhard Stock

Jeder Amiga wird mit einem Laufwerk ausgeliefert. Nach kurzer Zeit erwacht der Wunsch nach einem zweiten Laufwerk, zumal das ewige »Disk-Jockey« spielen nicht jedermanns Sache ist.

Warum, wird sich jetzt mancher fragen, muß dieser gewaltige Aufwand getrieben werden, um ein Laufwerk an den Amiga anzuschließen. Der Hauptgrund ist der, daß der

Amiga nicht vollständig kompatibel zum Standard des »Shugart-Busses« (MS-DOS-Industriestandard) ist, der sich mittlerweile auf der ganzen Welt durchgesetzt hat. Wir zeigen Ihnen eine Schaltung, mit der Sie selbst Rohlaufwerke anpassen können.

Elektronik

Die Schaltung selektiert das angeschlossene Laufwerk (Achtung, nur wenn Sie am Rohlaufwerk nichts verändert haben), bearbeitet den Diskettenwechsel, ist abschaltbar und kann auf Laufwerknummer »df1:« bis »df3:« eingestellt werden. In Bild 1 sehen Sie den zugehörigen Schaltplan. Vorab sei gesagt, daß sowohl der Bus auf der Amigaseite als auch der »Shugart«-Bus der Floppy low-aktiv sind, d.h. nur wenn an einem Pin ein Signal anliegt, geht der Pegel an diesem Pin auf 0 Volt. Alle übrigen Pegel liegen auf High, d.h. auf 5 Volt.

Nach einem Systemreset wird »Dres« (Drive Reset) vom Amiga auf 0 Volt gelegt. Nun werden die Select-Leitungen (Sel) und die »Mtrx« (Motor on) auf Low geschaltet. Normalerweise bewirkt »Mtrx«, daß alle angeschlossenen Laufwerke ihre Motoren einschalten. Bei mehreren Laufwerken (maximal vier) kann es mit der Spannungsversorgung besonders beim Amiga 500 Probleme geben. Ein Flipflop übernimmt deshalb die Selektion. Es ist in der Lage, ein Datenbit zu speichern. Das Signal der voreingestellten »Sel«-Leitung wird invertiert und das »Mtrx«-Signal ins Flipflop übernommen. Über ein NAND-Gatter wird der Ausgang an den »Ready«-Pin gemeldet. Mit den DIP-Schaltern kann man die Laufwerknummer einstellen:

Pin 21 = df1:
Pin 9 = df2:
Pin 20 = df3:

Da der Amiga die »Selx«-Leitungen nacheinander abfragt, erkennt er aufgrund der jeweiligen Ready-Rückmeldung, welches Laufwerk angeschlossen ist. Diese Prozedur läuft bei jedem Schreib/Lesezugriff ab, und da auch die »Mto« (Motor on) und die »Inu« (In use) Leitungen über ein NAND-Gatter an den Ausgang des »D-Flipflops« angeschlossen sind, können die Motoren der Laufwerke unabhängig voneinander ein- bzw. ausgeschaltet werden. Die nächste Schwierigkeit liegt im »Chng«-Signal (Diskchange). Man muß es entweder irgendwo im Laufwerk aufspüren und an den Shugart-Bus (Pin 2) anschließen oder mechanisch erzeugen.

Der Trick ist nun, daß man dem Amiga verständlich machen muß, bei welchem Laufwerk ein Diskwechsel vorliegt. Dies geschieht wieder durch die Verschaltung mit dem jeweiligen Select-Signal. Der Computer erkennt jetzt einwandfrei, wo eine Diskette gewechselt wurde und initialisiert das entsprechende Laufwerk neu. Als weiteres Bonbon bietet die Schaltung die Möglichkeit, das Laufwerk für den Amiga abzuschalten. Wenn in einem angeschlossenen Laufwerk keine Disk eingelegt ist, meldet es dies dem Amiga durch ein Low-Signal auf der »Chng«-Leitung. Die Leitung bleibt auf Low, bis der Computer einen »Step«-Impuls auslöst (auf Pin 18). Ist zu diesem Zeitpunkt schon wieder eine Disk im Laufwerk, springt »Chng« auf High zurück und der Amiga initialisiert. Ansonsten bleibt sie auf Low und der Computer muß weiterhin in regelmäßigen Abständen einen »Step«-Impuls auslösen, um zu erkennen, ob sich eine Disk im Laufwerk befindet.

Zusammenbau

Mit dem Schalter »S1« wird die Selectleitung unterbrochen, worauf der Computer das Laufwerk nicht mehr erkennt: Ende der Knackgeräusche. Man beachte aber, daß der für das Disketten-Laufwerk reservierte Speicherplatz (ca. 20 KByte) auf der Workbench erst wieder nach einem Reset freigegeben wird.

Nachdem man die Platine (Bild 2) hergestellt und die Bauteile besorgt hat (siehe Bauteileliste), kann mit

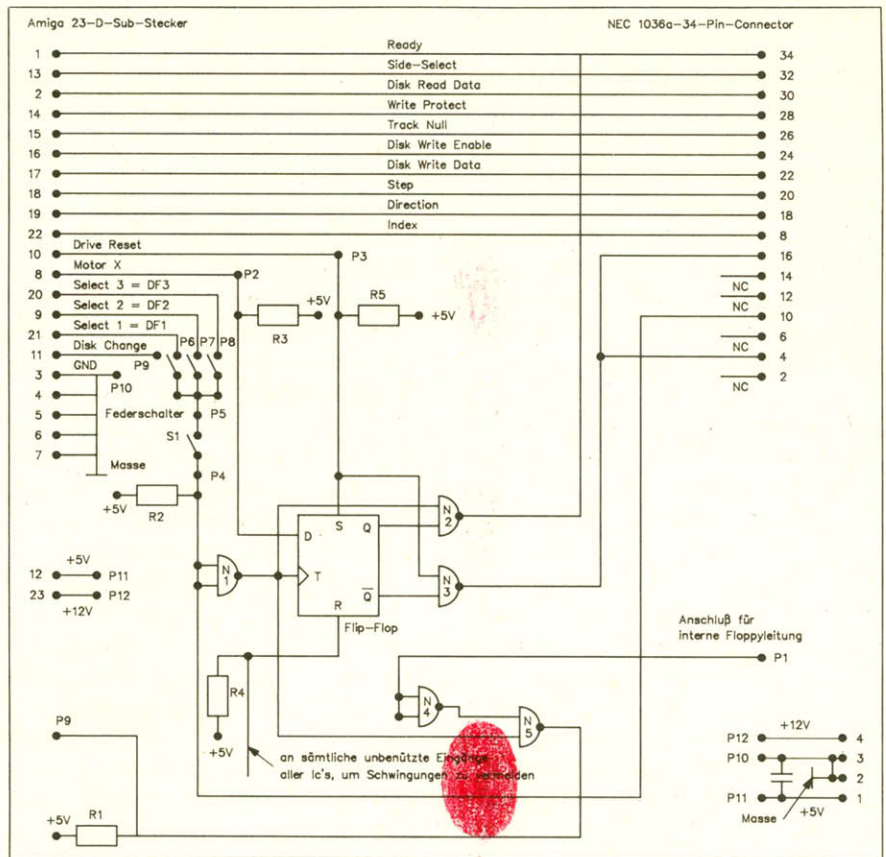


Bild 1. Der vollständige Schaltplan für die zugehörige Interfaceplatine

dem Aufbau begonnen werden. Als erstes wird der blanke Silberdraht mit zwei Zangen an den Enden gepackt und geradegezogen. Dann schneidet man ihn für die Lötbrücken in Stücke und lötet diese auf die Platine. Danach kommen die ICs, die Widerstände, der Kondensator und der DIP-Schalter. Am Schluß das Flachbandkabel abisolieren und in die obere Reihe der 17 nebeneinander liegenden Lötungen einlöten. Dabei liegt Lötauge 2 des »Shugart«-Busses in der Mitte, das äußerste Lötauge entspricht Pin 34 (durchgezählt 2, 4, 6, ..., 32, 34). Ans andere Ende des Kabels kommt der 17polige Flachstecker. Bitte nochmals überprüfen, ob wirklich Lötauge 2 an Pin 2 des »Shugart«-Busses in der Floppy angeschlossen ist. Ebenso wird mit dem vieradrigen Flachbandkabel für die Stromversorgung (Belegung Bild 1 rechts unten) verfahren. Die

zugehörigen Lötungen befinden sich in Bild 2 links neben denen für den »Shugart«-Bus, Pin 1 liegt zur Mitte hin, außen ist Pin 4.

Die übrigen Plazierungen zeigt Bild 3 (Achtung, auch hier alles von unten, von der Lötseite aus gesehen). Eine Änderung ist noch sehr wichtig: Der Punkt P1 auf der Platine wird über einen isolierten Draht mit Lötauge 2 der 17poligen Lötungenreihe des »Shugart«-Busses verbunden. Damit gelangt das Diskchange-Signal an der richtigen Stelle in die Platine.

Nachdem das Rundkabel an einem Ende 5 cm und am anderen 8 cm abisoliert wurde, entfernt man die Isolationslitze bis auf einen kleinen Rest, verdrillt diesen und verzinnt ihn. Die Litze verwendet man als Masseleitung. Dann werden die einzelnen Drähte entwirrt, jeweils auf 3 mm abisoliert und die Enden verzinnt. Am Sub-D-Stecker werden die Pins 3 bis 7 mittels eines blanken Drahtes verbunden und an die Masseleitung des Rundkabels angeschlossen. Danach die übrigen Leitungen des Kabels nach Bild 1 anschließen. Die jeweiligen Pin-

Nummern in der Zeichnung stimmen mit denen des Steckers überein. Nun die Zugentlastung und die Gehäuserückwand lose auf das Kabel stecken. Dies ist notwendig, da das Laufwerk nur von einer Seite ins Gehäuse paßt. Also Vorsicht, zuerst planen, dann bauen. Jetzt wird das Kabel an der Platine festgelötet. Bitte hierbei unbedingt die Farbcodierungen der einzelnen Leitungen beachten, damit gewährleistet ist, daß wirklich die entsprechenden Pins des Steckers mit den zugehörigen Lötäugen auf der Platine verbunden werden. Als nächstes wird auf der Rückblende des Gehäuses – falls nötig – ein Loch für den Ausschalter gebohrt. Darüber hinaus müssen einige Gehäusetyper mit zwei Bohrungen am Boden versehen werden, um dort später das Disketten-Laufwerk zu befestigen.

Nun werden die zwei isolierten Drähte für den Schalter auf der Platine angelötet und dann an den DIP-Schaltern die gewünschte Laufwerknummer eingestellt (Bild 1).

Bis hierher beschäftigt sich die Bauanleitung mit dem Zusammensetzen der Platine. Bevor jetzt das Laufwerk ins Gehäuse eingebaut wird, muß noch die Modifizierung für das Diskchange-Signal durchgeführt werden. Im Anhang sind die Umbauanleitungen der gängigsten Disketten-Laufwerke beschrieben.

Nach der Modifikation des Laufwerks ist die Platine samt Kabel für »Shugart«-Bus und Spannungsversorgung von der einen Seite durch das Gehäuse zu stecken und die Flachstecker am Laufwerk zu befestigen. Dann verlötet man den isolierten Draht am entsprechenden Lötäuge der Platine. Anschließend das Laufwerk in das Gehäuse schieben und verschrauben. Man schraubt den Schalter auf die Rückblende und lötet die beiden Anschlußdrähte an. Dann wird die Rückblende geschlossen. Als letztes noch die Zugentlastung befestigen.

Das Diskchange-Signal

Modifikation NEC 1036a: Zur Erzeugung des »chnng«-Signals am »Shugart«-Bus wird beim NEC 1036a im Laufwerk am Pin »OUT« ein isolierter Draht angelötet. Dieser Pin be-

findet sich rechts neben der Floppy-LED, oberhalb der großen Motorschwungradscheibe. Dort das Kabel seitlich anlöten und sorgfältig so durch das Laufwerk bis in die Nähe des Shugart-Bus-Steckers verlegen, daß der Draht den Motor nicht behindert. Oberhalb von Pin 2 des »Shugart«-Bus-Steckers befindet sich ein Feld mit Lötäugen. Mit einem Ohmmeter stellt man fest, welches Lötäuge mit Pin 2 verbunden ist (auf Durchgang prüfen) und lötet an dieses Auge das andere Ende des isolierten Drahtes. Die eingebauten Steckbrücken im Laufwerk müssen auf Kennung Laufwerk Null (6er

nicht. Gekoppelt ist dieser mechanische Schalter mit einem elektronischen Kontakt, der auf der Platine befestigt ist. Mit einem Voltmeter stellt man fest, auf welcher Leiterbahn sich der Spannungspegel ändert, wenn eine Diskette ins Laufwerk eingelegt wird (von Low auf High). Dieses Signal wird, wie oben beschrieben, an Pin 2 des Shugart-Busses angeschlossen. Noch ein kleiner Hinweis, daß diese Modifikation bei manchen Laufwerken nicht nötig ist, sie besitzen das Diskchange-Signal ab Werk. Dies bitte an Pin 2 mit einem Voltmeter überprüfen.

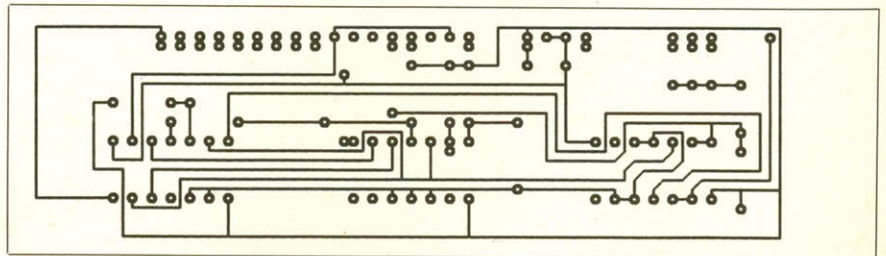


Bild 2. Das Platinenlayout der Interfaceplatine im Maßstab 1:1 (spiegelverkehrt)

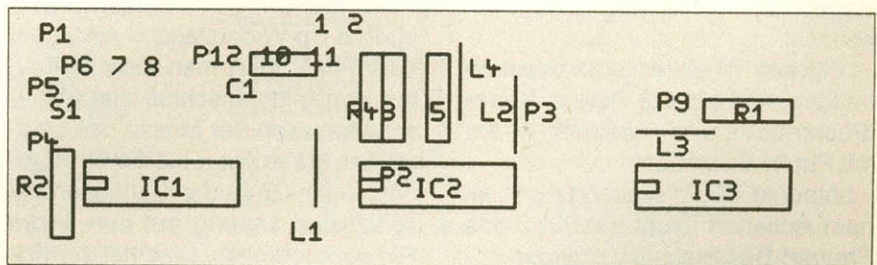


Bild 3. So werden die einzelnen Bauteile auf die Platine gelötet

Jumper) und der kleine Jumper auf Verbindung links geschaltet werden. Dabei liegt das Laufwerk auf dem Kopf und der Shugart-Bus ist zu uns hingewandt.

Bei 3 1/2-Zoll-Laufwerken anderer Hersteller empfiehlt sich folgende Vorgehensweise: Beim Blick durch den Diskettenschlitz ins Laufwerk stellt man fest, daß sich auf der rechten Seite ein mechanischer Schalter befindet, der entweder über einen Plastikzapfen oder über den Auswurfknopf des Laufwerks feststellt, ob eine Diskette im Laufwerk ist oder

NEC 1037a: Dieses Modell ist der Nachfolger der 1036a-Laufwerke. Es ist mit neuer SMD-LOW-Power-Technik gebaut, wodurch z.B. die 12-V-Versorgungsspannung wegfällt, und auf eine höhere Speicherkapazität ausgelegt. Ermöglicht wird dies u. a. durch einen neuentwickelten Steppermotor, der nach dem linearen 2-Phasen-Prinzip arbeitet. Er führt keine rotierende Bewegung mehr aus, sondern fährt linear vor und zurück, ähnlich einer Streichholzsachtel, bei der man das Innenteil heraus- und hineinschiebt. Dadurch werden die Abmessungen des Gerätes erheblich verkleinert. Auch die 1037a besitzen einen genormten Shugart-Bus SA100 und arbeiten einwandfrei mit der Interfaceplatine zusammen.

Bauteileliste:

IC1	74LS00
IC2	74LS74
IC3	74LS38
R1 bis R5	Widerstand, 1 k Ω , Metallfilm, 0,25 W, 5 Prozent
C1	Kondensator, 100 nF, 50V, Keramik
S1	einpoliger Ein/Ausschalter für Rundbohrung
1	23poliger D-Sub-Stecker mit Plastikgehäuse
1	Flachstecker, 17polig
1	einreihiger Flachstecker, 4polig
3	Schalter (Feder - oder DIP-Schalter)
1	Flachbandkabel, ca.10 cm, 40adrig
1	FDD NEC 1036a, NEC 1037a, FD-55-FR oder ähnliche
1	Zugentlastung zur Gehäuserückwand passend
1	kupferkaschierte Hartpapierplatine 100 mm x 30 mm
	etwa 70 cm Rundkabel, 20adrig, geschirmt
	ein zum Laufwerk passendes Metallgehäuse nach Wahl
	etwa 30 cm isolierte Kupferlitze
	etwa 20 cm blanker Silberdraht

Das Diskchange-Signal wird nach einer Empfehlung von NEC bei der »FD 1037a«, Typ 064 (bitte nachprüfen, da es verschiedene Versionen gibt), folgendermaßen erzeugt:

- Das Gehäuse vorsichtig öffnen und das Gehäuseoberteil entfernen.
- Das Gerät liegt mit der Rückseite vor Ihnen, die Platine ist oben.
- Die Leiterbahn, die zu Pin 34 führt, kurz vor Pin 34 mit einem scharfen Gegenstand vorsichtig durchtrennen.
- Lötspunkt TR (Ready/ die Bezeichnungen sind auf der Platine aufgedruckt) über einen isolierten Draht mit Pin 34 verbinden.
- Lötspunkt TD (Diskchange) mit einem isolierten Draht mit Pin 2 des Shugart-Bussteckers verbinden.
- Gehäuse wieder vorsichtig schließen und zusammenbauen.

Bitte daran denken, daß durch diese Änderungen jeglicher Garantieanspruch verlorengeht.

Als letztes kontrolliert man noch, ob der Schiebeschalter auf der Platine an der Gehäuserückwand auf »D0« steht.

TEAC FD 55 FR: Die folgende Beschreibung bezieht sich auf ein »TEAC FD-55-FR«-Laufwerk, kann aber auf fast jedes andere 5 $\frac{1}{4}$ -Zoll-Laufwerk übertragen werden. Im Normalfall wird bei keinem 5 $\frac{1}{4}$ -Zoll-Laufwerk ein Disk-Change-Signal erzeugt, da dies unter Betriebssystemen wie MS-DOS oder CP/M auch nicht erforderlich ist. Man könnte nun versuchen, dieses Signal aus der Elektronik des Laufwerks zu generieren, die Wahl fiel aber auf eine mechanische Lösung,

da diese dann auch bei anderen Laufwerktypen verwendet werden kann. An geeigneter Stelle wurde ein Mikroschalter (hier einer von Conrad Electronic, Bestellnummer: 703702) so mit einer M3 x 10-Schraube befestigt, daß er durch die Mechanik beim Schließen und Öffnen der Laufwerksklappe betätigt wird. Dies muß man beim Einbau genau justieren. Nun zwischen den 5 V des Versorgungssteckers und Pin 2 des Shugart-Busses einen 1 k Ω -Pull-Up-Widerstand einlöten. Vom Pin 2 führt man eine Leitung auf den Mittelanschluß des Mikroschalters, von der Masse des Laufwerkes (es existiert hierfür ein eigener Anschluß an der Gehäuserückseite) eine Leitung auf den linken Pin des Schalters. Dies hat zur Folge, daß bei geschlossenem Laufwerk 5 V an Pin 2 anliegen, bei geöffnetem Laufwerk 0 V. Keine Angst, es entsteht hierbei kein Kurzschluß, da über den Widerstand maximal 5 mA an Strom fließen können. Als weiteres das Widerstands-Array E3317 (existiert in ähnlicher Form auch in allen anderen Laufwerkstypen) aus dem Gerät entfernen, da sonst im Amiga, auf der Interface-Platine und im Laufwerk zu viele Pull-Up-Widerstände parallel liegen und ein zu hoher Verluststrom entsteht. Wer eine 40/80-Trackumschaltung wünscht (unter 40 Tracks kann der MS-DOS-Emulator auch IBM-XT-Disketten verarbeiten), der lötet an

den mit »S7« bezeichneten Lötspunkten einen Schalter an. Wenn er offen ist, arbeitet das Laufwerk mit 80 Spuren, wenn er geschlossen ist mit 40.

Die Jumperbelegung hat beim Amiga wie folgt auszusehen:

Es müssen im einzelnen »F0«, »D0«, »U1«, »HL«, »IU« und »RY« an den beiden Steckbrücken gesetzt sein. Diese befinden sich hinter dem Shugart-Bus auf der Platine. Zum Schluß noch zwei Hinweise: Bitte immer die Anschlußbelegungen des Shugartbusses und der Versorgungsspannung kontrollieren, da sich nicht alle Laufwerk-Hersteller an die Reihenfolge halten und Pins vertauscht sein können (beim TEAC FD-55-FR liegen an Pin 4 des Spannungsversorgungssteckers 5 V und an Pin 1 12 V, also genau vertauscht zur NEC 1036a). Zumindest beim Amiga 500 sollte ein 5 $\frac{1}{4}$ -Zoll-Laufwerk nicht als »df3:« betrieben werden, das Netzteil macht dies nicht mehr mit. Verwenden Sie auch nur »Low-Power«-Versionen, also Laufwerke neueren Datums, da die 12 V durch das Hochfahren des Antriebsmotors stark belastet werden.

All die beschriebenen Änderungen müssen Sie auch durchführen, wenn Sie so ein Laufwerk an den Amiga 2000 anschließen möchten. Lediglich beim ersten Zusatzlaufwerk ist keine Interface-Platine nötig.

Dazu befestigen Sie das Laufwerk im Schacht des Amiga-Gehäuses und schließen die Stromversorgung und das Floppykabel an. Sollte dieses Laufwerk das erste zusätzliche Laufwerk sein, muß der Jumper auf der Amiga-Grundplatine gesetzt werden, der sich unmittelbar neben dem 34poligen Flachbandkabel befindet. Sonst erkennt der Computer dieses Laufwerk nicht.

Zum Abschluß eine Bitte an die Leser: Viele haben sicherlich eigene Erfahrungen beim Modifizieren von Laufwerken verschiedenster Hersteller gemacht und sind bereit, diese Erfahrungen und Anleitungen an die Leser des AMIGA-Sonderhefts weiterzugeben. Die Redaktion wird diese Tips sammeln und in geschlossener Form veröffentlichen. *sq/pe*

AMIGA

SONDERHEFTE

1 Rund um den Amiga

Ausführliche Kurse, Bauanleitung, Tips&Tricks

Das Sonderheft 1 enthält einen ausführlichen Basic-Kurs - alles über den richtigen Umgang der alternativen Benutzeroberfläche "Command Interface Line". Bauanleitung: Mit dem Digitizer und der zugehörigen Software "Digisoft Plus" tauchen Sie ein in die Welt der digitalen Klänge. Außerdem jede Menge Tips&Tricks.

5 Public Domain

Software zum Nulltarif: Das Beste aus aller Welt

Die ganze Faszination der Grafik in einem Projekt. Wichtigster Themenbereich sind Utilities mit zahlreichen Hilfsprogrammen.

9 Die besten Listings des Jahres

20 Spitzenprogramme zum Sparpreis - Tolle Spiele, faszinierende Grafik-Programme, Nützliche Anwendungen, Hilfreiche Tools

10 Basic

Grundlagen: Schritt für Schritt: der leichte Einstieg in die Basic-Programmierung, **Know-how:** Einbindung von System- und Assembler-Routinen, **Professionell:** Hardware-Programmierung in Basic **Tools:** Compiler: Was leisten AC-, GFA- und Hisoft-Basic, Utilities: Der Profi-File-Requester für alle Zwecke, Unvorstellbar: Top-Zeichenprogramm in Basic

2 Listings der Spitzenklasse

Tolle Anwendungen, Tips, Tricks und Tools

"Fontdesigner", "Objekt-Editor", "Haushaltsbuch", "Keyboard-Master", "Fast Load-Copy", "Super-Copy" sind nur einige von vielen Anwendungen im Sonderheft 2. Außerdem: Ratschläge zu Superbase und zwei außergewöhnliche Spiele.

6 Anwendungen

Workshops: Experten verraten alle Tricks im Umgang mit den professionellen Programmen Sonix, Superbase und Beckertext. **Überblick:** Anwendungsprogramme und die Vorstellung der besten Programme. **Top-Programme fast umsonst:** Hochkarätige Anwendungen zur Textverarbeitung, Kontenführung und Dateiverwaltung bestechen durch ausgereifte Benutzerführung.

11 Grafik-Animation

Workshops: Deluxe Paint III, Deluxe Video 3.0 und Create A Shape **Grundlagen:** Alles Wissenswerte über Grafikformate und Grafik-Bildschirme **Tips:** Icons und Images selbst erstellt **Kaufberatung:** die besten Grafik-Programme

3 Basic und Spiele

10 Spitzenspiele zum Abtippen in AMIGA-Basic!

Die absoluten Spiele Top-Hits für den AMIGA in einer Übersicht. Mit Klassikern wie "Shanghai" und vielen brandneuen Hits. Ein ausführlicher Basic-Kurs führt zu Programmen mit rasanten Geschwindigkeiten.

7 Spieleprogrammierung

Knowhow, Projekte, Spiele

Knowhow: Wir lüften die Geheimnisse von Blitter, Copper & Co. **Projekte:** Die erste 3D-Vektor Library für den Amiga ermöglicht völlig neue Erlebnisse in der Grafik-Welt. Lesen Sie über die Entstehung neuer Spiele von der Idee bis zum Programm. **Spiele:** Fünf brandheiße Spiele aus diesem Heft sind - neben vielen Tools und Utilities - auf Diskette erhältlich.

Die AMIGA Sonderhefte bieten umfassende Informationen zu speziellen Themen rund um den AMIGA.

Diese Ausgaben erhalten Sie umgehend von Markt&Technik Leserservice, CSJ Postfach 140 220, 8000 München 5 Am besten füllen Sie gleich den Coupon aus und schicken ihn los.

4 C und Assembler

Zwei Super-Kurse und viele Tips und Tricks

Der große C-Kurs für alle Umsteiger. Programmieren Sie ab sofort mit kräftiger Unterstützung durch Betriebssystem und Libraries. Der Assembler-Kurs enthüllt alle Geheimnisse des 68000er Prozessors.

8 Starthilfe

Know-how für den Einstieg: Kurse - CLI und Workbench, Problemlösung - Amiga optimal konfiguriert **Rat&Tat:** Kaufberatung - Hardware, Software, Public Domain, Pannenhilfe - wenn Sie nicht mehr weiterwissen, Lexikon - So verstehen Sie Computer-Chinesisch.

Ich bestelle: _____ Ausgaben AMIGA SH Nr. _____
 _____ " " " Nr. _____
 _____ " " " Nr. _____
 _____ " " " Nr. _____

Insgesamt: _____ Ausgaben für 16,-DM pro Exemplar zzgl. Versandkosten

Name, Vorname _____

Straße, Hausnummer _____

PLZ, Wohnort _____

Die Bezahlung erfolgt nach Erhalt der Rechnung. Schicken Sie bitte den ausgefüllten Coupon an: Markt&Technik Leserservice, CSJ Postfach 140220, 8000 München 5

AMIGA
BESTELLCOUPON

So macht Lernen Spaß!



Die nächste Prüfung ist gerettet: Die neuen Lernprogramme »Spielend lernen« für Ihren Amiga ersetzen stupide Paukerei durch Abenteuer, Spaß und Quiz. Trockenes Schulwissen wird spannend aufbereitet und einprägsam vermittelt. Jedes Programm hat einen Lernteil und ein Quiz oder Spiel zur Überprüfung des neu erworbenen Wissens.

Jedes Programm nur
DM 49,-*
 (sFr 45,-*/öS 490,-*)
 *Unverbindliche Preisempfehlung

**Erdkunde I
 Bundesrepublik und DDR**
 Bestell-Nr. 38774



Mathematik I – Geometrie
 Bestell-Nr. 38777

Mathematik II – Algebra
 Bestell-Nr. 38778

**Mathematik III
 (Bruchrechnen)**
 Bestell-Nr. 38786

**Erdkunde II – Vereinigte
 Staaten von Amerika**
 Bestell-Nr. 38776

**Physik I ≙ Mechanik,
 Wärmelehre, Optik**
 Bestell-Nr. 38779

Englisch I
 Bestell-Nr. 38775

Deutsch I (Grammatik)
 Bestell-Nr. 38787

Markt&Technik-Bücher und -Software erhalten Sie bei Ihrem Buchhändler, in Computer-Fachgeschäften und in den Fachabteilungen der Warenhäuser.


Markt&Technik
 Zeitschriften · Bücher
 Software · Schulung