DM 14,80/ÖS 124/SFR 14,80 Nr. 7/88

CPC-WELT

CPC SPECIAL



Pascal: Schnelle Dateiverwaltung

Hardware: Sprachausgabe auf dem CPC

Datenfernübertragung: Grundlagen Programme Tips

Programmierung: GEM-Simulation in BASIC

Amstrad im Aufwind

In letzter Zeit schien es, als gäbe es rund um die CPCs nichts Neues. Eher im Gegenteil: Die Schneider-Geräte waren in kaum einem Geschäft mehr zu sehen, und nach Zubehör und Software konnte man lange suchen. Ein CPC mit dem Amstrad-Logo wurde noch nirgends gesichtet.
Grund genug für einen Besuch in Neu-Isenburg. Von Amstrads Pressesprecher Korcan Ataman erfuhren wir, was die britische Firma – die in diesem Jahr übrigens zwanzigjähriges Jubiläum feiert – für die Zukunft plant.
Ab Juli will Amstrad mit preiswerten Audio- und Video-Geräten auch in Deutschland vertreten sein und Schneider das Fürchten lehren. Eine Stereo-Anlage inklusive CD-Player für unter 500 Mark oder eine Vierspur-Bandmaschine für Hobby-Discjockeys und -Bands zu einem fast ebenso unglaublichen Preis sind wahrlich Produkte, die dazu geeignet sind. Den Endverbraucher freut's, die Konkurrenz packt das Grausen.
Doch was bleibt für die Computer-Freaks? Keine Sorge, die „Braune Ware" ist für Amstrad nur ein Nebenverdienst. PCs sind die Geräte, mit denen man das große Geld zu machen hofft.
Auch hier soll akzeptable Qualität zu einem vernünftigen Preis geboten werden.
Dem Ausdruck „Homecomputer-Hersteller" wurde energisch widersprochen.
Im professionellen Bereich möchte man sich einen Namen machen. Drei neue Profi-Computer, die im Oktober zur Kölner Orgatechnik präsentiert werden, sollen das Image der britischen Firma prägen.
Welche Rechner dies sein werden, wollte man uns noch nicht verraten. Spekulationen sind jedoch erlaubt.
Da Schneider mit einem AT aufwarten kann, dürfte ein Rechner auf der Basis des 80286-Prozessors bei Amstrad nicht lange auf sich warten lassen. Mit einem 80386er AT hätten die Briten sogar die Nase vorn.
Da IBMs neues Personal System 2 (PS/2) allgemein als Standard der Zukunft betrachtet wird, will kein Computer-Hersteller den Anschluß an den PS/2-Zug verpassen. Es sollte sich also niemand wundern, wenn da plötzlich ein Rechner mit dem OS/2-Betriebssystem auftaucht, der das Amstrad-Signet auf dem Gehäuse trägt.
Was aber geschieht mit den CPCs? Völlig auf Homecomputer verzichten will Amstrad nicht.
Der CPC 464 soll als Spiele-Maschine vermarktet werden. Wohlgemerkt: nicht als Spiel-Maschine. Denn dafür ist er zu schade.
Für Programmier-Lernwillige und private Anwender ist die 6er-Serie der CPCs gedacht. Amstrad bezeichnet sie als „Computer für den semi-professionellen Anwender".
Der Kauf der Firma Sinclair hat Alan Sugar allerdings einen Rechner beschert, der ein typischer Homecomputer ist: den Spectrum +2. Spectrum-Freaks werden sich freuen, denn nach langem Rätselraten steht es nun fest: Der +2 kommt pünktlich zu Weihnachten, inklusive Joystick und jeder Menge Spiele.
Daß Amstrad nicht in den Ruf eines Spielkonsolen-Herstellers kommen will, zeigt sich daran, daß der Spectrum unter dem Namen Sinclair verkauft werden wird. Im Fachhandel werden Sie ihn nicht erhalten; er ist als Weihnachtsgeschenk geplant, das mal eben im Supermarkt um die Ecke besorgt wird.
Ähnlich dem Sinclair-Computer soll auch für die CPCs zu Weihnachten eine große Verkaufsaktion beginnen. Auch hier sind Überraschungen in Ausstattung und Preis zu erwarten. Von den erzielten Verkaufszahlen wird es letztlich abhängen, ob die derzeit bestehende Software-Krise überwunden wird.
Einige Hersteller tun ihr Desinteresse an CP/M-Programmen deutlich kund und sind dabei, die entsprechende Software vom Markt zu nehmen. Die Verbreitung der CPCs wird einer der Faktoren sein, die eine solche Entscheidung beeinflussen.
Gute Soft- und Hardware für den CPC ist jedoch immer noch erhältlich. Oft sind es die Hersteller, die von Anfang an Zubehör für den Amstrad herstellten, die mit neuen Produkten oder überarbeiteten Versionen aufwarten können.
Einer dieser Hersteller ist die britische Firma dk'tronics. Nachdem wir in der letzten Ausgabe die RAM-Disk für den CPC vorgestellt haben, ist heute der Sprachsynthesizer an der Reihe.
Zum Thema Software haben wir einige DFÜ-Programme für Sie zusammengetragen. Nachdem viele „Eintagsfliegen" auf den Markt kamen und wieder verschwanden, konnten sich doch ein paar gute Programme behaupten. Vielleicht kommen Sie durch unser DFÜ-Special auf den Geschmack. Unsere Verlags-Mailbox steht 24 Stunden am Tag zu Ihrer Verfügung.
Wie in jeder Ausgabe, wollen wir Ihnen auch diesmal den „Blick über den Zaun" nicht vorenthalten. Die Berichte zu allgemeinen Themen der EDV erfreuen sich, wie wir zahlreichen Zuschriften entnehmen können, großer Beliebtheit.
Doch auch für die echten CPC-Anhänger bieten wir wieder Grundlagen-Beiträge und jede Menge Listings. Schauen Sie doch mal rein in Ihre CPC-WELT und schreiben Sie uns Ihre Meinung.

Alwin Ertl □

INHALT

SONDERTEIL-DFÜ

Verbindung mit dem Rest der Welt
DFÜ-Programme für den CPC
Seite 10

Kleines Lexikon der DFÜ
Unentbehrliche Fachausdrücke
Seite 14

Hier spricht Ihr Computer
Die Datennetze der Bundespost
Seite 16

Der größte Computer der Welt
So funktioniert das Telefon-Netz
Seite 19

Wenig Leistung für viel Geld?
Kritische Anmerkungen zur DFÜ
Seite 22

REPORT

Der letzte Kaiser kommt um sechs
Computer in der Videothek
Seite 7

TEST & TECHNIK

Der CPC lernt sprechen
dk'tronics Sprachsynthesizer im Test
Seite 4

Neue BASIC-Dimensionen
RISC-Rechner Archimedes zeigt neue Konzepte auf
Seite 136

GRUNDLAGEN

Unbegrenzte Möglichkeiten
Der vierte und letzte Teil unseres Grafik-Kurses
Seite 31

Bit für Bit
Joystick-Abfrage effektiv programmiert
Seite 39

GEM auf dem CPC
Komfortable Menüs sind auch in BASIC möglich
Seite 100

Matrizen und Vektoren
Mathematik ohne Schrecken – mit unserem Programm
Seite 106

Wie in Babylon
Noch mehr Programmiersprachen zur Auswahl . . .
Seite 118

PASCAL

Blitzschnelle Adreßverwaltung
Dateiverwaltung für Turbo Pascal unter CP/M 3.0
Seite 110

SERVICE

Nippes
Zubehör für Edel-Computer
Seite 36

Dialog
Fragen, Kritik und Wünsche unserer Leser
Seite 28

News
Nachrichten und Neuigkeiten rund um den Computer
Seite 129

TIPS & TRICKS

Die BASIC-Trickkiste
System-Adressen richtig genutzt
Seite 132

BÜCHERKISTE

Wegweiser in den Irrgarten
Der CPC-BASIC-Wegweiser
Seite 24

Einführung in Turbo Pascal unter CP/M 80
Lehrbuch oder Nachschlagwerk?
Seite 26

LISTINGS

RSX Extended BASIC
Befehlserweiterung mit vielen Möglichkeiten
Seite 43

Pinballmania
Der schnellste Flipper für den CPC
Seite 51

Solon
Verteidigen Sie die Galaxis
Seite 58

Topstar
Noch ein Spiel im Weltraum
Seite 83

Disketten-Monitor
Den Disks auf die Tracks geschaut
Seite 88

Zoff in der Galaxis
Bombenleger gegen UFOs
Seite 95

Checker
Unser Prüfsummen-Programm für fehlerfreis Abtippen
Seite 99

LOAD & RUN

Wie immer die neuesten Spiele auf 16 Seiten –
getestet von der Redaktion Load&Run
Seite 65

dk'tronics Sprachsynthesizer

Der CPC lernt sprechen

Die Synthetisierung natürlicher Sprache ist ein Ziel, das seit den Urzeiten der EDV angestrebt wird. Inzwischen gibt es Standard-Chips, mit denen sich verständliche Wörter und Sätze erzeugen lassen. Darauf basiert auch der Sprachsynthesizer für den CPC.

Guten Tag, hier spricht Ihr CPC" – eine solche Begrüßung hatten wir eigentlich erwartet, als wir das Steckmodul des dk'tronics-Sprachsynthesizers angeschlossen hatten. Doch weit gefehlt. Die ROM-Software meldet sich beim Einschalten mit einem bescheidenen

SPEECH ROM 1.1

Ein kurzer Blick ins Handbuch, und wir sehen, wie wir unseren CPC zum Sprechen bringen können. Der RSX-Befehl

|SPEAK

aktiviert das Modul. Leise, aber deutlich zu verstehen, gibt unser CPC bekannt, was da angeschlossen ist:

„dk'tronics speech synthesizer"

tönt es aus den beiden mitgelieferten Lautsprechern. Zugleich erscheint eine Liste der verfügbaren RSX-Befehle auf dem Monitor. Der kleine Knopf, der sich rechts am Gehäuse befindet, offenbart

Der dk'tronics-Sprachsynthesizer sorgt für Stereo-Sound am CPC



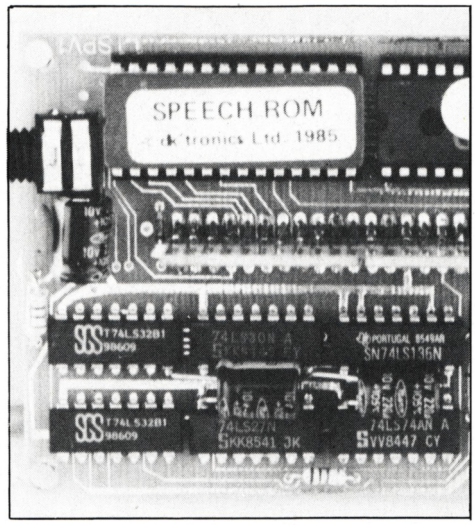
schnell seine Funktion: Er ist der Lautstärke-Regler für den Synthesizer. Da das Gerät mit Hilfe eines kurzen Kabels am Stereo-Ausgang des CPC angeschlossen wird, hat der eingebaute Lautsprecher seine Funktion ebenso verloren wie der Lautstärke-Regler des Rechners.
Nebenbei erhalten Sie eine vollwertige Stereo-Wiedergabe des CPC-Sounds. Bei manchen Musikprogrammen lohnt sich dies durchaus. Der Klang der beiden kleinen Lautsprecher ist allemal besser als das übliche CPC-Gepiepse.
Da der Sprachchip selbst nicht genügend Leistung liefert, um die beiden Lautsprecher betreiben zu können, mußte in das Gerät eine Endstufe eingebaut werden. Es war eine gute Idee von dk'tronics, diese Baugruppe auch als Musik-Verstärker einzusetzen.
Nach einigen Versuchen haben wir die Lautstärke auf ein angenehmes Maß geregelt. Ein Balance-Regler ist ebenfalls vorhanden.
Die Einstellung kann jedoch nur mit Hilfe eines Schraubenziehers verändert werden. Dies ist auch sinnvoll, da der Sprachsynthesizer für die Stereo-Wiedergabe bereits passend justiert ist.
Leider ist der kleine Regler alles andere als griffig, und leicht bewegen läßt er sich schon gar nicht. Da der Platinenstecker unseres 464 eine recht wackelige Verbindung darstellt, haben wir bei unseren Versuchen so manchen ungewollten Reset ausgelöst. Man tut also gut daran, die Finger vom Lautstärke-Regler zu lassen, solange ein Programm läuft.

Sprache mit PRINT

Nachdem diese Vorarbeiten erledigt waren, sollte unser CPC endlich einen vernünftigen Satz aussprechen. Der Synthesizer bietet gleich mehrere Möglichkeiten an, dies zu bewerkstelligen.
Die erste besteht darin, die sogenannte Text-to-Speech-Software zu verwenden. Dies ist ein Programm im ROM des Gerätes, das englischen Klartext in seine einzelnen Laute zerlegen und ausgeben soll.
Für den Anwender ist diese Software die bequemste Lösung. Sie müssen den gewohnten BASIC-Befehl PRINT nach dem Anführungszeichen lediglich um einen Apostroph ergänzen. Dann wird der String an den Sprachsynthesizer ausgegeben.
Die Software kann nicht alle Regeln und Ausnahmen der englischen Sprache erfassen. Deshalb müssen manche Wörter geringfügig falsch geschrieben werden. Nach einigen Versuchen und mit etwas Sprachgefühl ist es jedoch einfach, verständliche Sätze zu produzieren.
Natürlich klingt die Sprachausgabe blechern und monoton. Sie erinnert ein wenig an einen Roboter in einem Science-fiction-Film. Doch wer mehr erwartet hat, muß zugeben, daß seine Hoffnungen unrealistisch waren. Mit dem geringen Bauteile- und Software-Aufwand des dk'tronics-Synthesizers sind die Möglichkeiten eben eingeschränkt.
Immerhin läßt sich die Geschwindigkeit der Ausgabe über ein RSX-Kommando regeln. Damit können Sie Wörtern und Sätzen eine gewisse Betonung verleihen. Eine möglichst langsame Sprechweise ist – im Interesse der Verständlichkeit – jedoch anzuraten.

Deutsche Sprache, schwere Sprache

Die eingebaute Software eignet sich auch für deutsche Texte. Sie müssen die Schreibweise jedes Wortes jedoch ans Englische angleichen. So würde unser bekanntes oo etwa als u gesprochen. Es ist manchmal nicht einfach, eine passende Version herauszufinden.
Für deutsche Texte ist die zweite Möglichkeit, die Ihnen der Sprachsynthesizer bietet, am günstigsten. Sie können dem Chip direkt die numerischen Werte für jeden Laut übergeben.
An dieser Stelle ist ein kurzer Blick



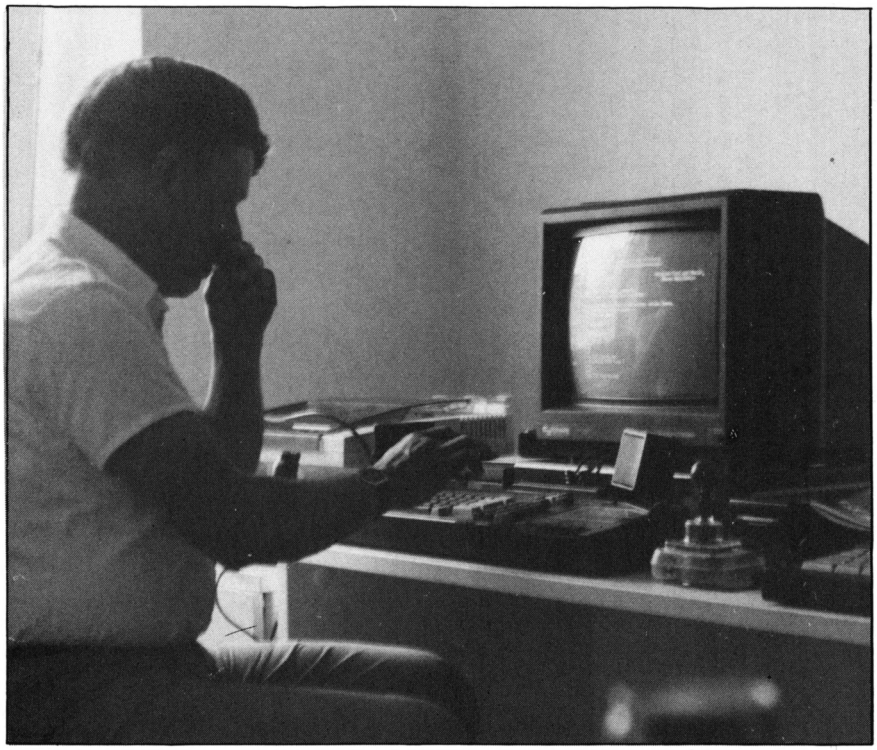
auf die Funktionsweise des Sprach-Chips angebracht. Während ältere Geräte meist einen fest vordefinierten Wortschatz hatten, an dem nichts zu ändern war, arbeitet der dk'tronics-Synthesizer nach dem Prinzip der Allophon-Synthese. Allophone sind die Laute, aus denen sich eine Sprache zusammensetzt. Sie kennen das sicher von der phonetischen Umschrift her, die Ihnen im Wörterbuch die Aussprache eines Wortes angibt.
Der Sprachsynthesizer verfügt über alle englischen Allophone. Eine Liste der zugehörigen nume-

Unser aktuelles Angebot:

Devpac-Assembler	DM 148,–
GFA Einsteiger-Set	DM 188,–
Datamat ST	DM 99,–
Profimat ST	DM 99,–
SPIELE	
Jump Jet	DM 69,–
DMX-Simulator	DM 49,–
Obliterator	DM 89,–
Ikari/Warriors	DM 49,–
The Land Lounge Lizards	DM 69,–
Goldrunner	DM 79,–
PD-Disketten	je DM 5,–
Fuji-Disketten MF 1DD 3,5" (10er Pack)	DM 35,–
ATARI-XE Video Game	DM 359,–

Wir nehmen SOFTWARE-ÜBERSETZUNGEN in 59 Sprachen vor!
Wir führen Finanzierungen durch!
Ständig gebrauchte Bücher und Geräte auf Lager!
Versand per Nachnahme (Versandkostenanteil DM 5,–) ansonsten Vorauskasse.
Fordern Sie unsere ausführliche Hardware- und Computer-Preisliste an!

Sie finden uns in unseren neuen Geschäftsräumen:
SCHÖNAICH Computer
Josef-Schregel-Straße 52, 5160 Düren/
Geschäftspassage, Telefon 02421/10379



Ein Schnappschuß aus der Redaktion: Programm-Redakteur Manfred Eibisch staunte nicht schlecht, als der CPC „Guten Morgen" sagte

rischen Werte finden Sie im Handbuch. Mit Hilfe des RSX-Befehls

```
|FEED
```

können Sie jeden Laut einzeln vorgeben. Obwohl manche Laute fehlen (etwa au, ü oder ch), läßt sich eine verständliche Sprachausgabe auch in Deutsch erzielen.
Ein Problem bei der Ausgabe längerer Sätze stellt der beschränkte Pufferbereich des Synthesizers dar. Er beträgt nur 250 Bytes. Daher kann es vorkommen, daß hin und wieder ein Satzteil „verschluckt" wird.
Sie können mit einem einfachen IF-Befehl abfragen, ob der Chip bereit ist, Daten zu empfangen. Diese Abfrage blockiert jedoch Ihr Programm: Während die Sprachausgabe ansonsten interruptgesteuert abläuft, müssen Sie in diesem Fall auf den langsamen Synthesizer warten, bevor BASIC weitermachen kann.
Dies hätte sich eleganter lösen lassen, etwa durch einen RAM-Puffer, der während des Interrupts abgearbeitet wird. In den meisten Fällen sind die 250 Bytes dennoch ausreichend; denn kaum ein Programmierer wird auf die Idee kommen, den Sprachsynthesizer einen Monolog vortragen zu lassen.
Die RSX-Befehle zur Umleitung des PRINT-Befehls auf den Synthi sind nur als Spielerei zu betrachten. Da auch bei englischen Texten hin und wieder falsche Schreibweisen erforderlich sind, hat es wenig Sinn, sich ein Listing vortragen zu lassen. Die einzige Ausnahme stellen vielleicht DATA-Lader dar, die Sie abgetippt haben: Während der Computer die Zeilen vorliest, können Sie anhand des Listings kontrollieren. Da diese Ausgabeform ohnehin selten oder nie verwendet wird, ist es auch zu verschmerzen, daß der Rechner gnadenlos abstürzt, wenn während des Interrupts auf die Diskette zugegriffen wird.

Verständliches Handbuch

Mit 14 Seiten ist die Anleitung für den Sprachsynthesizer nicht gerade ausführlich. Dennoch ist im Handbuch alles enthalten, was Sie für die Verwendung des Gerätes in BASIC und Assembler wissen müssen. Daß der Text englisch ist, dürfte wenig ausmachen. Ein sinnvoller Einsatz des Synthesizers ist ohnehin nur dann möglich, wenn Sie Englisch lesen und sprechen können.
Der Einsatz des Sprachsynthesizers dürfte sich hauptsächlich bei selbstprogrammierten Spielen lohnen. Er verbessert die Klangqualität des CPC und liefert hervorragende Effekte. Vor einem Zuviel an Sprache muß allerdings gewarnt werden: Die monotone Aussprache kann den Anwender auch entnerven.
Aber auch „ernsthafte" Anwenderprogramme können von etwas Sprache profitieren, wenn sie maßvoll eingesetzt wird. Hier sind die Möglichkeiten weitgehend noch nicht erkannt – wohl aufgrund des geringen Verbreitungsgrades solcher Geräte.
Auch in unserer Redaktion waren die Meinungen geteilt. Die einen hielten den Synthesizer für eine Spielerei, die anderen für eine ernstzunehmende Anwendung.
Wie immer auch Ihre Meinung dazu sei: Vom technischen Standpunkt aus betrachtet kann der dk'tronics-Sprachsynthesizer überzeugen. Der durchgeschleifte Bus ermöglicht den Anschluß zusätzlicher Erweiterungen. Kollisionen mit anderen ROMs haben wir nicht feststellen können; in Verbindung mit weiteren dk'tronics-Produkten (wie etwa der RAM-Disk, die wir in Ausgabe 6/88 vorgestellt haben) arbeitet er einwandfrei.
Die Investition von 129 Mark lohnt sich für CPC-Programmierer allemal. Dem reinen Anwender ist wenig damit gedient, da es keine entsprechende Software gibt.
Außer, wenn Sie ein Fan von Computer-Musik sind: Der Stereo-Sound des Synthesizers kann sich sehen – oder besser: hören – lassen. *AE*□

Der PC in der Videothek

DER LETZTE KAISER KOMMT UM SECHS

Filmverleih leicht gemacht – durch Computereinsatz

Was macht der Besitzer einer Videothek am Sonntag? Er brütet über Statistik und Monatsabrechnung. Ein anderer kann künftig spazierengehen: Er hat sich einen Computer zugelegt.



Ich hätte bereits vor zwei Jahren anfangen sollen, mit dem Computer zu arbeiten." Günther Tauer, Chef der Karlsfelder Videothek „Video-Kiste", hat lange genug an Sonntagen über Statistiken gebrütet. Der Wille zur Umstellung war wohl vorhanden – allein, es hat das geeignete Programm gefehlt.

Dieser Mangel ist seit Februar behoben. Ein befreundeter Programmierer hat ihm Vidiplus entwickelt, das auf die Bedürfnisse seines Geschäfts abgestimmt ist. Warum hat er sich nicht für eines der Fertig-Programme entschieden, die es bereits in allen Variationen und Preisklassen gibt?

Tauer winkt ab: „Ich habe mir das eine oder andere angeschaut. Ist zwar alles gut und schön, aber wenn ich Änderungen brauche, kostet das zusätzlich. Mit hundert Mark und mehr ist man gleich dabei."

Die täglichen Arbeiten lassen sich mit Vidiplus in sechs Masken abwickeln: Verleihgeschäft, Kundendatei, Filmeingabe, Bar-Code-Druck, Datensicherung, Mahnwesen.

Im Moment wird lediglich mit der Kundendatei und der Filmeingabe gearbeitet. Die weiteren Programmteile sollen in Kürze installiert werden. Problemlos zu handhaben sei die Sache auf jeden Fall, meint Tauer.

„... eine Riesenerleichterung"

„Was mir Roland Haas, der Programmierer, beibringt, werde ich vereinfacht ans Personal weitergeben." Etwa eine Stunde hat er dafür veranschlagt, zumal durch den Bar-Code-Lesestift einiges an Unterweisungszeit gespart würde.

Für den Videotheken-Besitzer ist das Programm „eine Riesenerleichterung". Obgleich die Video-Kiste zu den kleineren Geschäften dieser Art zählt, bietet sie auf etwa 60 Quadratmetern Platz für immerhin 3000 Filme. Rund 2500 verschiedene Titel sind derzeit im Angebot. Durchschnittlich 100 werden an den Werktagen verliehen, samstags sind es gar um die 300.

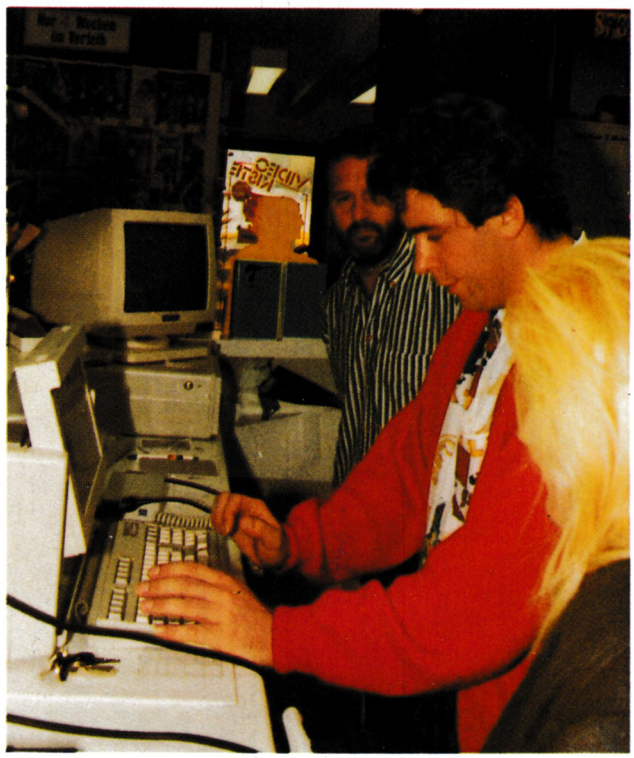
„An Stoßzeiten standen wir manchmal zu dritt hinter der Theke", sagt Tauer, der neben seiner Frau noch zwei Aushilfen für die Vormittage beschäftigt. „Manchmal war es das reine Chaos: Vom Kunden mußte ich außer dem Namen auch wissen, wann er das letzte Mal da war. Konnte er sich nicht genau daran erinnern, begann die ewig lange Sucherei in der alten und neuen Kartei." Was noch dazu kam: Verleih-Formular ausfüllen, auf die unterschiedlichen Preise achten, kassieren.

Das geht jetzt alles schneller: Der Kunde ist mit seinen Daten gespeichert, ebenso die Filme. Der Rest ist

Schluß mit den Strichlisten

ein Kinderspiel: Eingabe der Kundennummer, Namensvergleich, Heraussuchen des gewünschten Films. Preis und Titel werden über Bar-Code eingelesen. Weitere Arbeitsersparnis wird es geben, wenn die neuen lesbaren Kundenkarten verteilt sind. Tauer sieht darin auch ein psychologisches Moment: „Die Dinger werden bestimmt nicht einfach zu Hause vergessen. Die Leute schätzen diese technischen Spielereien. Die mögen es ganz gern, wenn etwas piepst."

Nicht nur fürs tägliche Verleihgeschäft läßt sich das Programm nützen. Schließlich will ein „Videothekar" auch wissen, wie häufig welche Filme über den Ladentisch gehen. Bisher hat Tauer Strichlisten geführt: „Ich muß einen Überblick haben, ob die verschiedenen Titel ihren Einkaufs-



Von links nach rechts; Günther Tauer und Roland Haas führen unserer Reporterin Waltraut Schmidt das System vor.

Unten: Durch den Einsatz der EDV kann sich G. Tauer intensiver den Wünschen seiner Kunden widmen.



preis auch eingespielt haben." Diesen Überblick hat er jetzt auf dem Bildschirm. Ebenso läßt sich daraus ersehen, mit welchen Filmherstellern welcher Umsatz erzielt worden ist.

Faulen Kunden keine Chance

„Von den faulen Kunden wird mir jetzt auch keiner mehr durch die Lappen gehen." Tauer spricht damit ein Problem der meisten Videotheken an: Da leiht sich einer Kassetten aus und verschwindet damit auf Nimmerwiedersehen.

Zwar werden Listen mit den Namen dieser schwarzen Schafe verteilt. In der Kundenkartei werden sie mit einem Sperrvermerk versehen. Nur: „Wenn einer behauptet, er sei zum ersten Mal hier, habe ich das Nachsehen." Diese Zeitgenossen werden jetzt unter einer bestimmten Nummer gespeichert. Da sich jeder neue Kunde sowieso zunächst ausweisen muß, genügt ein Blick auf den Bildschirm: ertappt.

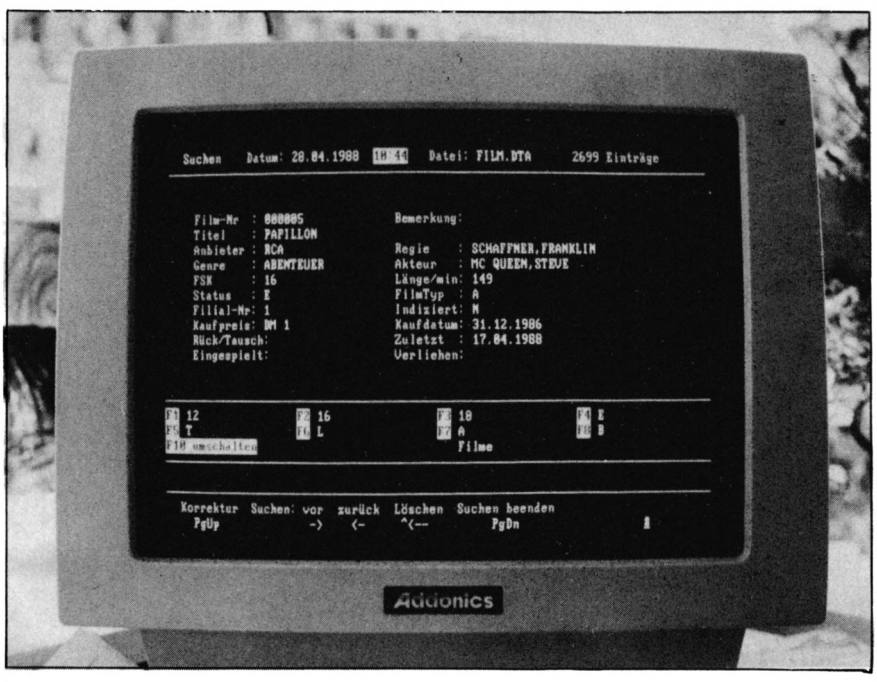
Das Programm zeigt auch an, wenn die Leihfrist überschritten wurde. Etwa 150 Mark, schätzt Tauer, gehen pro Jahr an Nachgebühren verloren – in der Hauptsache durch Unachtsamkeit, wenn viel Betrieb ist. Kein allzu großer Verlust zwar, doch wenn er sich vermeiden läßt ... „Die Leute müssen jetzt warten, bis alles eingelesen und überprüft ist."

Mitunter werden beschädigte Videos zurückgegeben. Wer dafür verantwortlich war, läßt sich nur schwer feststellen. Eine Hilfe bietet Vidiplus: Die letzten drei Kunden, die den Film geliehen hatten, werden aufgelistet.



Links:
Unauffällig und platzsparend paßt sich das System in das Arbeitsfeld ein.

Unten:
Eine der Bildschirmmasken zeigt alle Informationen über die angebotenen Filme. Die Daten der Kunden sind in einer ähnlichen Maske enthalten.



Alles ist jedoch noch nicht machbar: Eine Kundin möchte „Der letzte Kaiser" ausleihen. Tauer schaut auf eine Liste: „Tut mir leid, nicht vor nächster Woche. Höchstens noch heute abend, ab 18 Uhr." Die Reservierung ist im Programm nicht enthalten, es wird noch immer mit der bewährten Listen- und Zettel-Methode gearbeitet. Für die Zukunft ist aber auch sie vorgesehen.

Reservierung fehlt noch

Der Videotheken-Besitzer liebäugelt noch mit einer Textverarbeitung zum Erstellen von Serienbriefen. Ein Drucker ist ja bereits vorhanden.

Steht aber der finanzielle Aufwand in einem akzeptablen Verhältnis zum Ertrag? Bereits die Programme, die derzeit angeboten werden, kosten, je nach Umfang, zwischen 2.000 und 10.000 Mark. Wie teuer war der ganze Spaß? Rund 9.500 Mark insgesamt – für Computer, Drucker und Programm, sagt Haas. Mit Textverarbeitung und Bar-Code würden die Kosten auf etwa 12.000 Mark klettern, rechnet sich Tauer aus. Er ist mit weiteren Händlern in einer Einkaufs-Cooperative zusammengeschlossen, das Programm können alle gemeinsam nutzen.

„Klar, für einen Laden mit etwa fünf- bis sechshundert Titeln lohnt sich ein Computer nicht." Das habe auch mit der Kundenfrequenz zu tun. Aber ab anderthalbtausend Filmen und Kunden lasse sich darüber reden.

„Ohne Computer geht nichts mehr"

Für ihn hat die Sache auf jeden Fall Hand und Fuß. Da sich die täglich anfallenden Arbeiten einfacher und schneller erledigen lassen, kann er in Stoßzeiten auf einen Mitarbeiter verzichten. Überdies: „Ich kann jederzeit nachprüfen, mit welchem Filmgenre und welchen Firmen ich keinen Umsatz mache." Durch gesteuerten Einkauf kann er damit eine Menge Geld einsparen. Für ihn steht fest: „Ohne Computer geht nichts mehr."

was

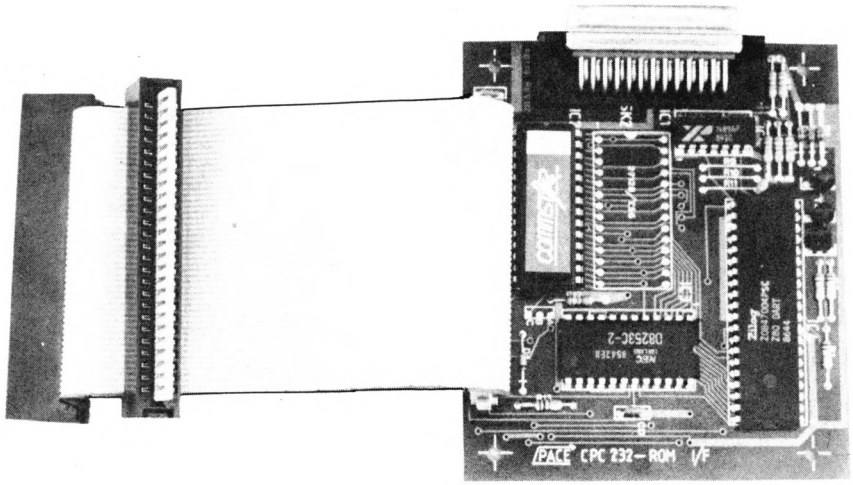
Datenfernübertragung

Verbindung mit dem Rest der Welt

Informationen via Telefon

Die Möglichkeit, Daten über Telefon senden und empfangen zu können, wird bisher nur von wenigen CPC-Besitzern genutzt. Unser Autor Michael Henke zeigt einige Lösungen auf, wie auch Ihr CPC mit der Außenwelt in Verbindung treten kann.

Verkäufer: „Dafür brauchen Sie eine serielle Schnittstelle und ein Modem, sowie natürlich das passende Programm."
Interessent: „Und was kostet das dann zusammen?"
Verkäufer schweigt und wühlt in seinen Unterlagen. „Einen Moment bitte . . ."
Diese Szene ist vielen DFÜ-Interessenten bekannt – ebenso wie der Preis, den sie nach einiger Wartezeit genannt bekamen.
Doch es geht auch billiger; denn gerade auf dem DFÜ-Sektor hat sich einiges getan. Daß der CPC nicht über einen seriellen Port verfügt, stellt allerdings schon das erste Hindernis dar. Es gibt aber eine Möglichkeit, diese Hürde auf preiswerte Art zu überwinden. Damit können auch CPC-Anwender den Einstieg in die Welt der Datenbanken und Mailboxen schaffen.
Betrachten wir aber zunächst, was eine RS232C ist und wie sie funktioniert.



Die Pace-RS232 ist die einzige serielle Schnittstelle für den CPC, die bereits DFÜ-Software im ROM enthält

Daten im Gänsemarsch

Mit RS232 wird der Port eines Rechners bezeichnet, an dem serielle Peripherie-Geräte – wie etwa ein Akustikkoppler – angeschlossen werden können. Am CPC werden Sie diese Schnittstelle jedoch vergeblich suchen. Er

verfügt lediglich über einen parallelen Port.
Parallel bedeutet dabei, daß jeweils acht Bits gleichzeitig über acht verschiedene Leitungen verschickt werden können. Der CPC weicht hier allerdings vom Standard ab, so daß Sie nur sieben Bits senden können. Das genügt jedoch, um jeden gängigen Drukker zu betreiben.
Eine serielle Schnittstelle verschickt jeweils nur ein einziges Bit. Um ein Zeichen zu senden, müssen Sie also acht Bits hintereinander ausgeben.
Ein serieller Port kann jedoch – im Gegensatz zum serienmäßig verfügbaren parallelen Port – nicht nur senden, sondern auch empfangen.
Bei einer parallelen Schnittstelle dienen drei Leitungen dazu, einen reibungslosen Datenfluß sicherzustellen. Man bezeichnet sie als **Status-Leitungen**; sie heißen BUSY, STROBE und ACKNOWLEDGE. Diese Anschlüsse des Ports sind Empfangskanäle; alle anderen sind entweder Datenleitungen (D0 bis D7) oder liegen auf Masse (GND).
Bei der seriellen Schnittstelle steuern nur zwei Leitungen den Datenverkehr. Dieses Verfahren bezeichnet man als **Handshake.** Mit drei Drähten ließe sich also bereits eine serielle Schnittstelle realisieren; im Standard sind jedoch weitere Leitungen enthalten, die zum Beispiel die Betriebsbereitschaft eines Peripherie-Gerätes überprüfen.
In der DFÜ werden für diesen Zweck bestimmte Steuerzeichen (Control-Codes) verwendet. Mit CTRL-S etwa teilen Sie einer Mailbox mit „Bitte warten, ich komme gleich wieder". CTRL-Q bedeutet „Ich bin wieder am Apparat". Weitere Codes sind CTRL-X (Break, Abbrechen) und CTRL-C (Skip, Überspringen).

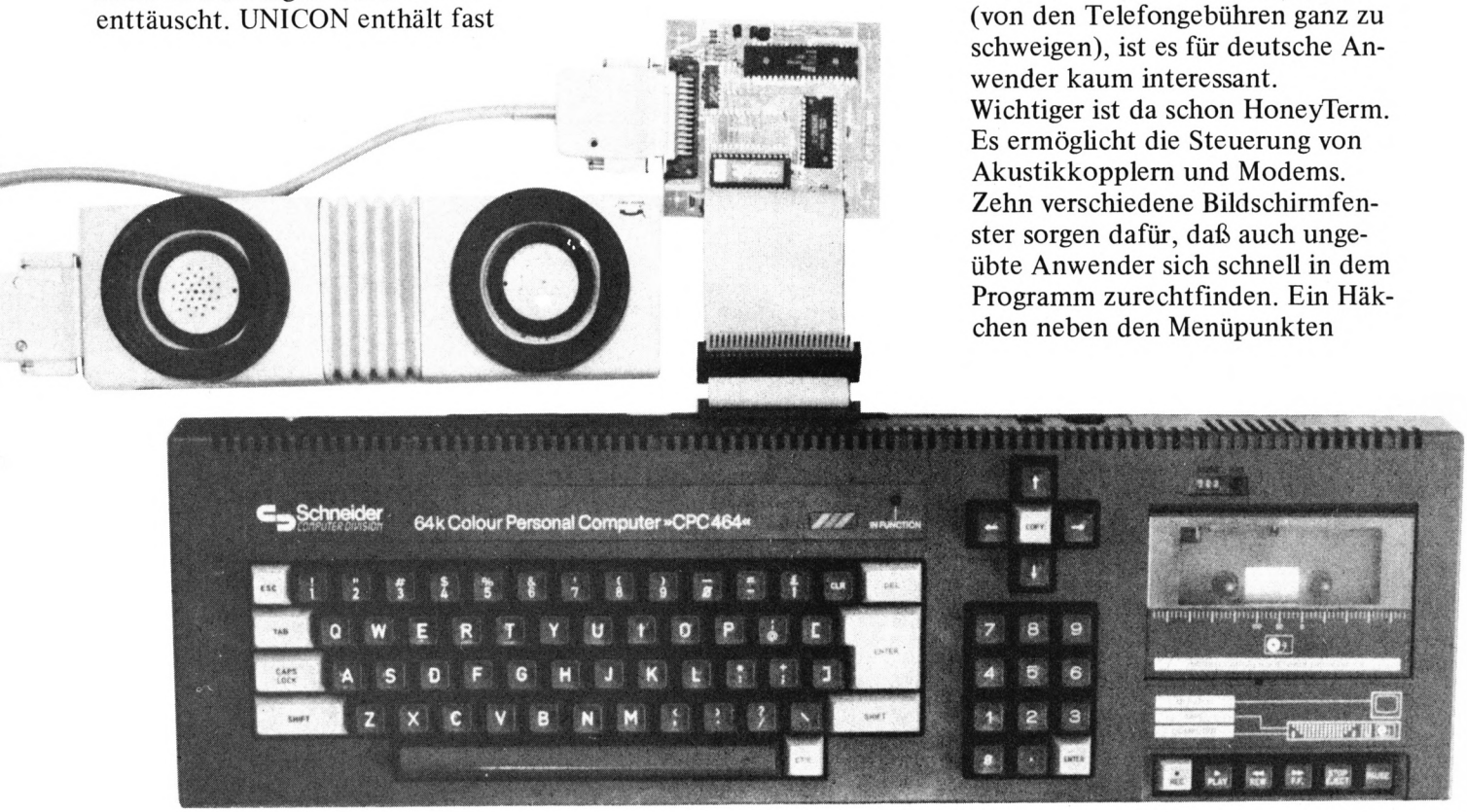
Hard- und Software für den CPC

Seit der CPC den Durchbruch als Heimcomputer schaffte, tat sich auf dem DFÜ-Sektor so gut wie nichts für diesen Rechner. Schneider bot zwar eine Schnittstelle samt Programm an; der Preis dafür war zur Markteinführung allerdings schier unerschwinglich.
Findige Köpfe machten sich jedoch bald daran, eine Lösung zu suchen, die ohne die teure RS-232-Schnittstelle von Schneider auskam. Insgesamt kamen sechs verschiedene Programme auf den Markt, die aus einer seriellen Schnittstelle **und** der passenden Software bestanden. Von diesen sechs Programmen haben zwei bis heute überlebt. Sie heißen TELEPORT und UNICON.
Das Programm TELEPORT war eine angenehme Überraschung, nachdem viele „magere" Programme vorgestellt worden waren. In Verbindung mit einem dataphon s21d Akustikkoppler lief es vorzüglich – kein Wunder, ist die Software doch speziell für dieses Gerät, einen der preiswertesten Akustikkoppler, ausgelegt.
Der Anschluß der Schnittstelle und des Kopplers gestaltete sich problemlos, so daß bald mit den ersten Versuchen an einer Mailbox begonnen werden konnte. Arbeitsweise und Bedienung des Terminal-Programms gaben keinerlei Anlaß zu Beanstandungen. Ein Test der Schnittstelle in Verbindung mit einem seriellen Drukker brachte ebenfalls positive Resultate. Doch dieser Druckertyp ist wenig verbreitet, so daß die DFÜ wohl die Hauptanwendung bleiben wird.
Auf der Kassette findet sich sowohl eine Tape- als auch eine Disk-Version von TELEPORT. Beim Start der Disketten-Version werden alle benötigten Dateien automatisch von der Kassette auf die Disk kopiert. Auf überflüssige Kopierschutz-Maßnahmen hat der Hersteller hier klugerweise verzichtet.
TELEPORT speichert Dateien direkt auf Disk oder Tape, wobei die Disk-Version einige zusätzliche Möglichkeiten bietet. So ist zum Beispiel das Arbeiten mit einem zweiten Laufwerk ohne weiteres möglich, wodurch die Kapazität erheblich erweitert wird. Der integrierte Text-Editor kann Dateien bis maximal 39 KByte verarbeiten; für den Ausdruck von Dateien gibt es kein Limit. Vom Vertreiber wird ein Programm kostenlos angeboten, das TELEPORT an das DATEX-P-Netz anpaßt. Damit ist der Einstieg in professionelle Netze auch mit dem CPC möglich.
Das zweite Programm, UNICON,

Insider-Tip

Die Möglichkeit, eine serielle Schnittstelle mit Hilfe des Druckerports zu realisieren, wird von verschiedenen DFÜ-Paketen für den CPC genutzt. Versierte Bastler und Assembler-Programmierer können sich eine solche Pseudo-RS-232 auch selbst bauen. Dabei wird eine Datenleitung des Druckerports zum Senden, die BUSY-Leitung zum Empfangen von Daten genutzt. Wird die entsprechende Software in die Fast Ticker Chain eingereiht, so sind Übertragungsraten bis zu 300 Baud möglich – genug, um reibungslos mit einer Mailbox zu kommunizieren.
Es genügt jedoch nicht, nur die Leitungen zu verlängern und einen Akustikkoppler anzuschließen. Da Centronics und RS232 unterschiedliche Pegel verwenden, müssen diese erst angepaßt werden. Nur erfahrene Elektronik-Hobbyisten sollten sich an diese Aufgabe heranwagen. *AE*

fällt zunächst durch eine schlechter verarbeitete Schnittstelle auf. Sie wird am Druckerport angeschlossen, wobei ein durchgeschleifter Bus dafür sorgt, daß gleichzeitig ein Printer betrieben werden kann.
Beim ersten Test fiel auf, daß UNICON von denselben Autoren stammt, die auch TELEPORT geschrieben hatten. Das weckte einige Erwartungen.
Diese Erwartungen wurden nicht enttäuscht. UNICON enthält fast



dieselben Optionen wie TELEPORT – und einige mehr. Die Steuerung erfolgt über eine Maus-Simulation, wobei die Cursortasten oder der Joystick als Eingabe-Gerät dienen können.
Ein Pegel-Anpasser auf der Platine sorgt dafür, daß der CPC keinen Schaden nehmen kann; mittels eines Schalters kann zwischen Modem und Drucker gewählt werden. Andere serielle Geräte lassen sich mit dieser Schnittstelle nicht betreiben. Für die DFÜ ist dieses Paket aber beinahe unschlagbar.
Für die RS232 von Schneider beziehungsweise Amstrad existieren ebenfalls noch zwei Programme: TERMINAL STAR und TELECOM 1000.

Die Schneider-Schnittstelle

TERMINAL STAR war das einzige Programm, das die Türkheimer für ihre RS232-Schnittstelle auf den Markt brachten. Es wird zwar professionell auf Diskette geliefert, doch die Benutzerführung läßt einiges zu wünschen übrig.
Der eingebaute Text-Editor brachte den Tester schnell an den Rand der Verzweiflung.
TELECOM 1000 machte einen besseren Eindruck, besaß aber keinen integrierten Editor. Die Handhabung konnte im großen und ganzen überzeugen, reichte aber nicht an die von TELEPORT oder UNICON heran.

Die Pace-Schnittstelle

Neues bot erst wieder die Firma Amstrad mit der Pace-RS232. Die passende Software findet der erstaunte Käufer in ROM-Form auf der Schnittstellen-Platine. Der Aufruf erfolgt über RSX-Befehle; einer Einbindung in CP/M steht ebenfalls nichts im Wege.
Für diese Schnittstelle sind zwei Programme verfügbar, HoneyTerm und HoneyView.
Bei HoneyView handelt es sich um ein Programm, das speziell für das englische Btx-System konzipiert ist. Da hierfür auch ein entsprechendes Modem erforderlich ist (von den Telefongebühren ganz zu schweigen), ist es für deutsche Anwender kaum interessant.
Wichtiger ist da schon HoneyTerm. Es ermöglicht die Steuerung von Akustikkopplern und Modems. Zehn verschiedene Bildschirmfenster sorgen dafür, daß auch ungeübte Anwender sich schnell in dem Programm zurechtfinden. Ein Häkchen neben den Menüpunkten zeigt die gerade aktivierten Optionen; eine Statuszeile am unteren Bildschirmrand liefert zusätzliche Informationen.
Das Handbuch ist leider nur in Englisch erhältlich. Es sorgt damit für die längst fällige Auffrischung der Sprachkenntnisse – gewollt oder ungewollt.
Ein RAM-Puffer sorgt dafür, daß empfangene Daten nicht verlorengehen. Der Inhalt dieses Puffers kann auf jedem Medium (Disk A oder B sowie Kassette) gespeichert oder auf dem Drucker zu Papier gebracht werden.
Zudem ist die Schnittstelle nicht

nur für den Betrieb von Modems gedacht. Auch andere serielle Peripherie-Geräte können angeschlossen werden. Das Interface selbst wird über den Expansions-Port des CPC mit Strom versorgt. Über ein Bus-Kabel findet aber auch der Floppy-Controller noch Platz, also keine Sorge.

Michael Henke/AE □

DFÜ-Programme in Stichpunkten

Die folgenden DFÜ-Programme sind für den CPC noch erhältlich; Sie können sie über den Computer-Fachhandel beziehen. Die Testnoten, die wir den Programmen jeweils gegeben haben, kamen aufgrund subjektiver Erfahrungen zustande. Dennoch muß erwähnt werden, daß einige Programme schlecht abschnitten, weil sie nicht frei von Fehlern waren.

CPC-Termy

Parameter veränderbar; 40- und 80-Zeichen-Darstellung.
Bewertung: Befriedigend

DESC

Serielles Sende- und Empfangs-Programm unter CP/M.
Bewertung: Befriedigend

CPC-Term

Parameter veränderbar; deutscher Zeichensatz; RS232-Schnittstelle über Drucker-Port realisiert.
Bewertung: Ausreichend

Terminal Star

Einstellmöglichkeiten für Übertragungs-Parameter; integrierter Text-Editor: angepaßt an Schneider-RS-232-Schnittstelle.
Bewertung: Mangelhaft

VALC

RS232-Schnittstelle am Floppy-Port realisiert; Menüführung.
Bewertung: Ausreichend

Telecom 1000

Konzipiert für Schneider-RS-232; menügesteuerte SIO-Initialisierung.
Bewertung: Befriedigend

Teleterminal 300 S

Orientierung über Status-Anzeige; RS232 am Drucker-Port realisiert.
Bewertung: Befriedigend

Terminal

Integrierter Texteditor; Paßwortschutz; Diskettenversion erlaubt zwei Benutzer.
Bewertung: Befriedigend

Unicon

Menüführung über Maus-Simulation; Parameter bis 300 Baud; integrierter Texteditor; RS232 am Drucker-Port realisiert.
Bewertung: Gut

High-Term

Integrierter Texteditor: angepaßt für Schneider-RS232-Schnittstelle; integrierter Wandler.
Bewertung: Befriedigend

Teleport

Menüführung; RS232 am Joystick-Port realisiert; integrierter Texteditor; Parameter bis 300 Baud; DATEX-P-fähig.
Bewertung: Sehr gut

CPC-COM 2.1

Ausgefeilte Window-Technik; läuft unter CP/M; geringe Leistung; gutes Handbuch; RS232-Schnittstelle nötig.
Bewertung: Ausreichend

TeleCom

Mit GEM-ähnlicher Fenstertechnik.
Bewertung: Ausreichend

Honeyview/Honeyterm

Menüführung über Zehner-Tastatur; Software auf der Schnittstellen-Platine vorhanden; dürftiges Handbuch; viele Möglichkeiten.
Bewertung: Gut bis sehr gut

Die Auswahl Ihres DFÜ-Programms können Sie nur nach einer Analyse Ihrer Ansprüche treffen. Vor dem Kauf sollten Sie sich das Programm vorführen lassen und die Leistungsfähigkeit einzelner Programme sowie die Preise unterschiedlicher Händler vergleichen. Es lohnt sich aber in jedem Fall, in die Welt der Datenfernübertragung einzusteigen.

Michael Henke/AE □

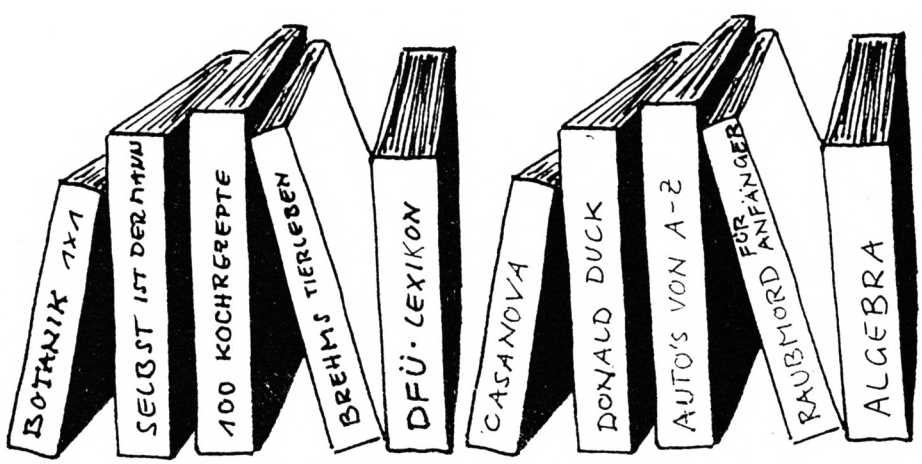
DFÜ-Special

Kleines Lexikon der DFÜ

Wichtige Begriffe

Genauso, wie sich Computer-Freaks bevorzugt in Fachchinesisch unterhalten, umgeben sich auch DFÜ-Anhänger mit einem Hauch des Geheimnisvollen, indem sie möglichst viele Fachbegriffe verwenden. Einige dieser Ausdrücke sind unvermeidbar. Wir wollen Ihnen die wichtigsten kurz vorstellen.

Wir wollen nicht den Anspruch erheben, mit diesem Beitrag eine vollständige Einführung in die Welt der Datenfernübertragung zu bieten. Unser kleines Begriffs-Lexikon soll Ihnen nur als Hilfe dienen, Beiträge und Bücher zu diesem Thema besser zu verstehen.



Baud

ist die Maßeinheit für die Geschwindigkeit, mit der Daten übertragen werden. Benannt wurde sie

nach dem französischen Physiker Baudot.
Ein Baud ist definiert als ein Übertragungsschritt pro Sekunde. Dies wird leider oft falsch als Bit pro Sekunde interpretiert. Für die Übertragung über eine serielle Leitung nach dem FSK-Verfahren trifft dies zwar zu, bestimmte Methoden ermöglichen es aber (zum Beispiel im Funkverkehr), mehrere Bits auf einmal zu codieren. Die Maßeinheiten Bits pro Sekunde (bit/s) und Baud (bd) können daher völlig unterschiedliche Werte angeben.

Akustikkoppler

Gerät zur Umwandlung von Daten in akustische Informationen. Der wichtigste Unterschied zum Modem besteht darin, daß es zwischen Akustikkoppler und Telefon keinerlei elektronische Verbindung gibt. Die einzelnen Töne werden, wie beim normalen Fernsprechen, über den Hörer übertragen und empfangen.

Modem

ist ein Kunstwort für **Mo**dulator-**Dem**odulator. Das Funktionsprinzip entspricht dem des Akustikkopplers; hier besteht jedoch eine elektrische Verbindung zum Telefonnetz. Nach dem derzeitigen Fernmeldeanlagen-Gesetz dürfen nur posteigene Modems ans Netz angeschlossen werden.

DATEX-L

Datenübertragungsnetz der Deutschen Bundespost mit Leitungsvermittlung. Der Anschluß kann nur über spezielle Endgeräte erfolgen.

DATEX-P

Datenübertragungsnetz der Deutschen Bundespost mit Paketvermittlung. Der Anschluß erfolgt über spezielle Endgeräte oder mit Hilfe einer Zubringer-Leitung aus dem öffentlichen Telefonnetz.

Sysop

System **Op**erator, der „Verwalter" und meist Eigentümer einer Mailbox.

FSK

Frequency **S**hift **K**eying, Verschlüsselung durch Frequenz-Änderung. Dieses Verfahren wird von Akustikkopplern und Modems benutzt, um Daten über das Telefonnetz zu übertragen. Dabei werden die Bit-Werte Null und Eins in zwei verschiedene hörbare Frequenzen umgesetzt.

Mailbox

Öffentlich zugänglicher „elektronischer Briefkasten" mit Schwarzen Brettern für alle und Privatfächern für eingetragene Benutzer. Wenn Sie einen Akustikkoppler und ein Terminal-Programm besitzen, rufen Sie einfach die CA-Mailbox, die Mailbox unseres Verlages, an. Als Username verwenden Sie Ihren Vornamen oder ein Pseudonym; als Paßwort genügt GAST.

Download

Das „Herausziehen" einer Datei, meist eines Programms, aus einem anderen Rechner (wie etwa einer Mailbox). Der sendende Rechner ist „downloading", der empfangende ist „uploading".

Paßwort

Ein vom Anbieter einer Mailbox oder Datenbank beziehungsweise vom Benutzer festgelegter Code, der den Anrufenden als berechtigt ausweist. In Mailboxen ist das richtige Paßwort notwendig, um eingetragenen Benutzern den Zugriff auf ihre Privatfächer zu ermöglichen. Bei Datenbanken, die gebührenpflichtig sind, wird damit sichergestellt, daß kein Unbefugter kostenlos Gebrauch von dem angebotenen Service macht.

Protokoll

Methode, nach der eine Übertragung von Daten gesteuert wird. Meist dienen bestimmte Control-Codes dazu, die Übertragung zeitlich abzustimmen, damit beide Rechner nicht gleichzeitig „reden". Vergleichbar ist dies mit bestimmten festgelegten Ausdrücken und Redewendungen beim Funkverkehr, die ebenfalls einer geregelten Kommunikation dienen.

PAD

Packet **A**ssembly/**D**isassembly **F**acility. Spezielle Knotenrechner im DATEX-P-Netz, mit deren Hilfe ein Zugang ins Netz auch ohne spezielle Endgeräte möglich wird.

RS232C

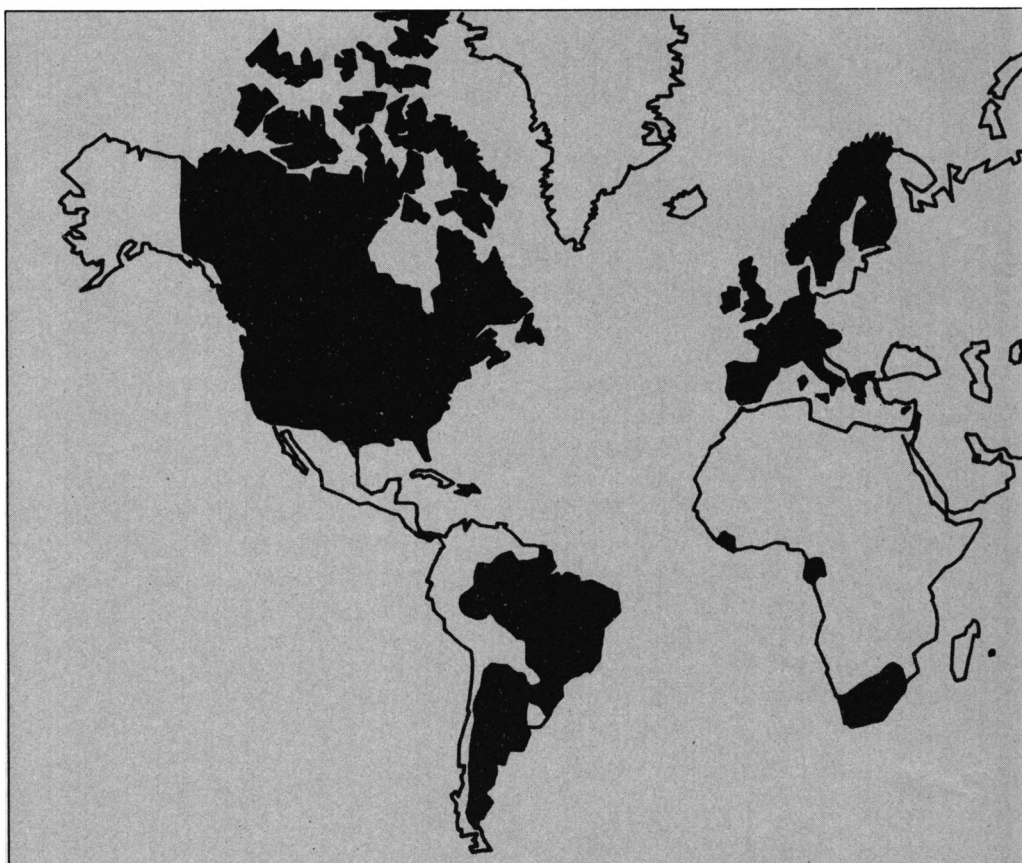
oder V.24 – zwei verschiedene Bezeichnungen für die Norm, nach der die serielle Schnittstelle eines Computers festgelegt wird. Akustikkoppler und Modems können in der Regel nur über eine solche Schnittstelle angeschlossen werden. Bei DFÜ-Freunden ist oft nur kurz von der „RS232" oder der „RS" die Rede. □

Hier spricht I

Daten-Netze

Die Deutsche Bundespost bietet ihren Kunden verschiedene Möglichkeiten an, ihre Computerdaten über öffentliche Netze zu übertragen. Welche Möglichkeiten dies sind und ob es sich lohnt, davon Gebrauch zu machen, wollen wir Ihnen in diesem Beitrag schildern.

Bereits in den sechziger Jahren wurden von der Deutschen Bundespost die ersten Netze zur Datenübertragung geschaffen: das Telex- und das Telefon-Netz.
Das Telex-Netz, der altbekannte Fernschreibdienst, ist heute noch in Betrieb. Aufgrund der niedrigen Übertragungsraten und des eingeschränkten Zeichensatzes verliert es jedoch zunehmend an Bedeutung.
Der Code, der beim Telex zur Zeichenübertragung verwendet wird, besteht aus fünf Bits pro Buchstabe. Um den gesamten Zeichenvorrat eines Computers senden und empfangen zu können, werden hingegen mindestens sieben, besser aber acht Bits benötigt. Da das Telex-Netz zudem nur mit 50 Baud (in diesem Fall identisch mit der Maßeinheit Bits pro Sekunde) arbeitet, ist es für die Übertragung größerer Datenmengen in computerlesbarer Form denkbar ungeeignet.
Im Jahr 1967 wurde daher das erste öffentliche Netz eingeführt, das speziell für die Datenübertragung entwickelt wurde: DATEX-L. DATEX ist ein Kunstwort, das aus der englischen Bezeichnung **Data Exc**hange, also Daten-Austausch, gebildet wurde.
Das L bedeutet Leitungsvermittlung: Zwischen den beiden ver-

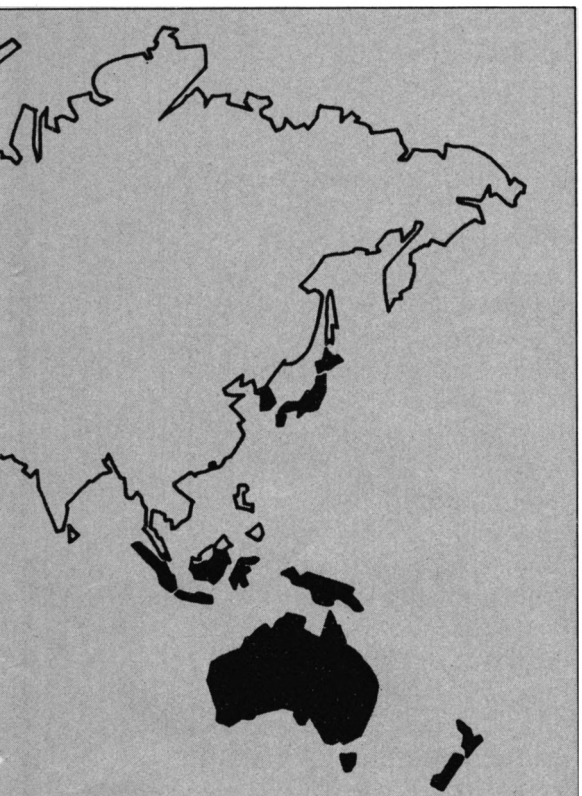


Nordamerika	Südamerika	Europa		
Französische Antillen	Argentinien	Belgien	Irland	Portugal
Kanada	Brasilien	Dänemark	Italien	Schweden
Vereinigte Staaten von Amerika (USA)	Französisch Guayana	Finnland	Luxemburg	Schweiz
	Panama	Frankreich	Niederlande	Spanien
	Peru	Griechenland	Norwegen	
		Großbritannien	Österreich	

Diese Länder können Sie über DATEX- P erreichen (Grafik: DBP)

hr Computer

Datenübertragung über Bundespost-Netze



Afrika
Elfenbeinküste
Gabun
Réunion
Südafrika

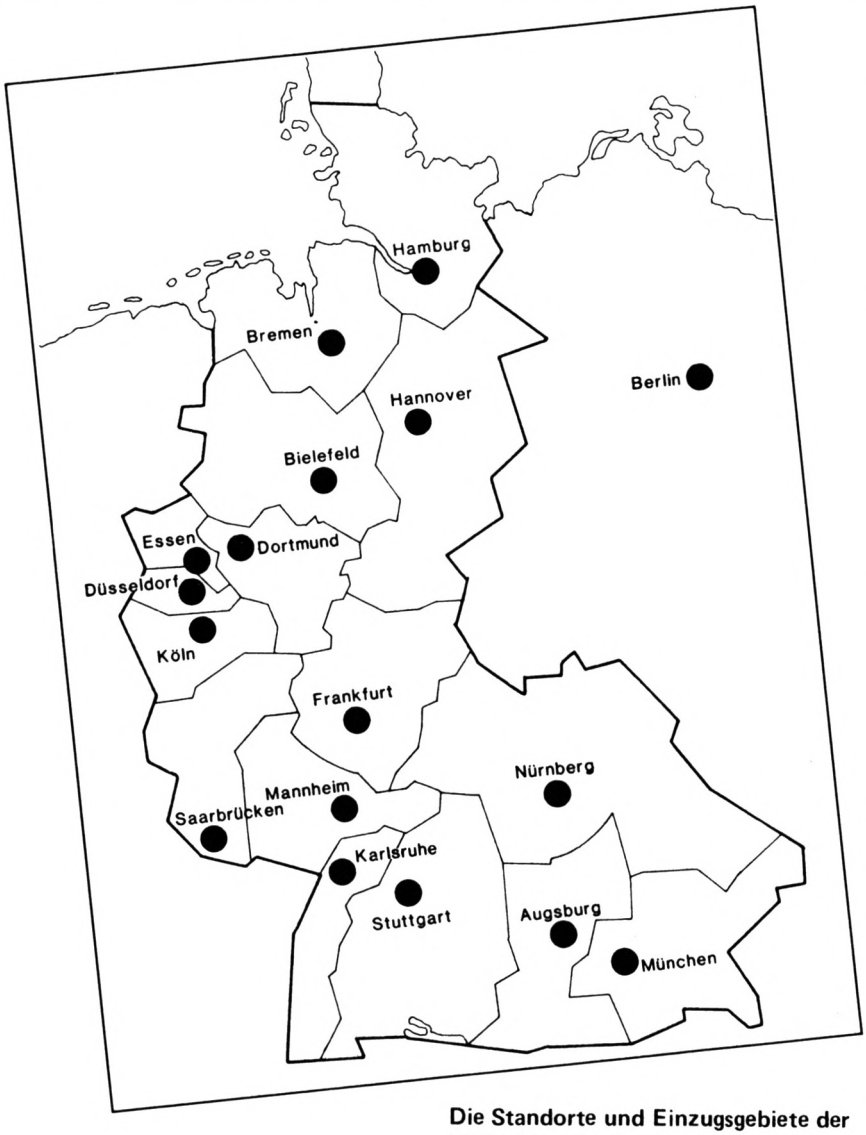
Asien
Bahrain
Hongkong
Israel
Japan
Korea

Ozeanien
Australien
Indonesien
Neuseeland
Singapur

bundenen Stationen wird eine Leitung geschaltet, die für die ganze Dauer der Verbindung bestehen bleibt.
DATEX-L bietet Übertragungsraten von 300 bis 64.000 Bits pro Sekunde. Zur Teilnahme an diesem Netz ist ein besonderes Endgerät notwendig, das von der Post gekauft oder – ähnlich dem Telefon – gemietet werden kann.
Aufgrund der verwendeten Technik ist die Nutzer-Kapazität des DATEX-L-Netzes beschränkt. Entsprechend hoch sind die Grundgebühren, die monatlich für einen Hauptanschluß fällig werden. Sie reichen von 120 Mark bei 300 Baud bis zu 510 Mark bei 9.600 Baud. Höhere Übertragungsraten – bis zu 64 KBit pro Sekunde – können zu höheren Gebühren bereitgestellt werden; jedoch ist dies nicht im gesamten Bundesgebiet möglich.
Das DATEX-L-Netz, das sich durch hohe Übertragungs-Geschwindigkeiten auszeichnet, soll nur noch bis zur Verfügbarkeit des digitalen ISDN-Netzes angeboten werden. ISDN (Integrated Services Digital Network) soll alle bisher existierenden Netze sowie einige zusätzliche Möglichkeiten unter einen Hut bringen. Bis zur Einführung dieses Post-Dienstes kann jedoch noch einige Zeit vergehen.
Der Anschluß an ein leistungsfähiges Datennetz ist damit vorerst die einzige Möglichkeit, große Datenmengen sicher zu übertragen. Während DATEX-L eher für Fälle geeignet ist, in denen regelmäßig große Datenmengen zu übertragen sind, kommt DATEX-P auch für geringere Datenaufkommen in Frage.

Datenpakete: DATEX-P

Das P in der Netzbezeichnung bedeutet Paketvermittlung. Hier wird, im Gegensatz zu DATEX-L, eine virtuelle Verbindung zwischen zwei Stationen aufgebaut. Das heißt, daß durch die Vermittlungsstelle zunächst Leitungskapazitäten reserviert werden. Die Leitung wird aber erst dann geschaltet, wenn tatsächlich Daten übertragen werden.
Für einen Hauptanschluß werden monatliche Gebühren von 140 bis 2.500 Mark berechnet. Sie richten sich wiederum nach der gewünschten Übertragungs-Geschwindigkeit. Diese reicht von 300 bis 48.000 Baud.
Wenn Sie auf die Bereitstellung eines posteigenenen Gerätes verzichten, lassen sich diese Gebühren jedoch unter Umständen reduzieren. Die Struktur des DATEX-P-Netzes erlaubt auch den Anschluß privater Modems oder Akustikkoppler.
Die Umwandlung der Daten in für das P-Netz geeignete „Pakete“



Die Standorte und Einzugsgebiete der PADs (Grafik: DBP)

übernehmen 17 PADs (Packet Assembly and Disassembly Facilities). Diese PADs oder Knotenrechner sind über das Bundesgebiet verteilt und jeweils einem bestimmten Einzugsbereich zugeordnet. Die Aufstellungsorte und Einzugsgebiete können Sie Abbildung 1 entnehmen.
Jeden dieser PADs können Sie über das Telefonnetz anwählen und so eine Verbindung mit Ihrem Computer herstellen. Damit ist jeder Rechner erreichbar, der an das DATEX-P-Netz angeschlossen ist. Allerdings werden zusätzlich die Gesprächsgebühren für die Nutzung des Telefonnetzes als „Zubringer" fällig.
Ein Einschalten der PADs als Vermittler erweist sich jedoch dann als Vorteil, wenn etwa Daten im mobilen Einsatz an eine Zentrale gesendet werden sollen. Die verwendeten Rechner besitzen in der Regel kein Modem und kein DFÜ-Programm, das dem DATEX-P-Standard entspricht. Die PADs übernehmen in diesem Fall die Verpackung der Daten in handliche „Pakete".
Seine wirklichen Stärken zeigt DATEX-P bei der Übertragung von Daten ins Ausland. In Abbildung 2 sehen Sie die Länder, die Sie über dieses Netz erreichen können. Da hier die Gebühren, im Gegensatz zum normalen Fernsprechnetz, nicht zeit-, sondern mengenabhängig sind, könnte dies für manches Unternehmen die günstigste Lösung darstellen.
Die gängigste Methode, Daten zu übertragen, stellt für den privaten Anwender jedoch das herkömmliche Fernsprechnetz dar. Nach Postangaben können dabei Geschwindigkeiten bis zu 4.800 Baud erreicht werden. Rein technisch ist dies zwar richtig; aber wer die Qualität mancher Leitung kennt, wird diese Angaben zu Recht bezweifeln.
Praktikabel sind Übertragungsraten bis 1.200 Baud. Während bei den DATEX-Netzen mit jeder Geschwindigkeit im Duplex-Betrieb (Senden und Empfangen sind gleichzeitig möglich) gearbeitet werden kann, sollten Sie im Direktrufnetz möglichst einfache Verfahren verwenden. Nur dadurch erreichen Sie die geringstmögliche Anzahl an Übertragungsfehlern.
Für gelegentliche Übertragung geringer Datenmengen ist das herkömmliche Telefonnetz jedoch am besten geeignet. Dies gilt nicht nur für Hobbyisten und Freaks, sondern auch für Firmen, die auf diese Weise Kontakt mit nahegelegenen Filialen halten wollen.
Das altbewährte Telefon war ursprünglich nicht zur Übermittlung von EDV-Daten vorgesehen. Dennoch gehört es in der Terminologie der Deutschen Bundespost zu den Datel-Diensten. Datel steht für den englischen Ausdruck **Da**ta **Tel**ecommunications, also Datenfernübertragung (DFÜ).

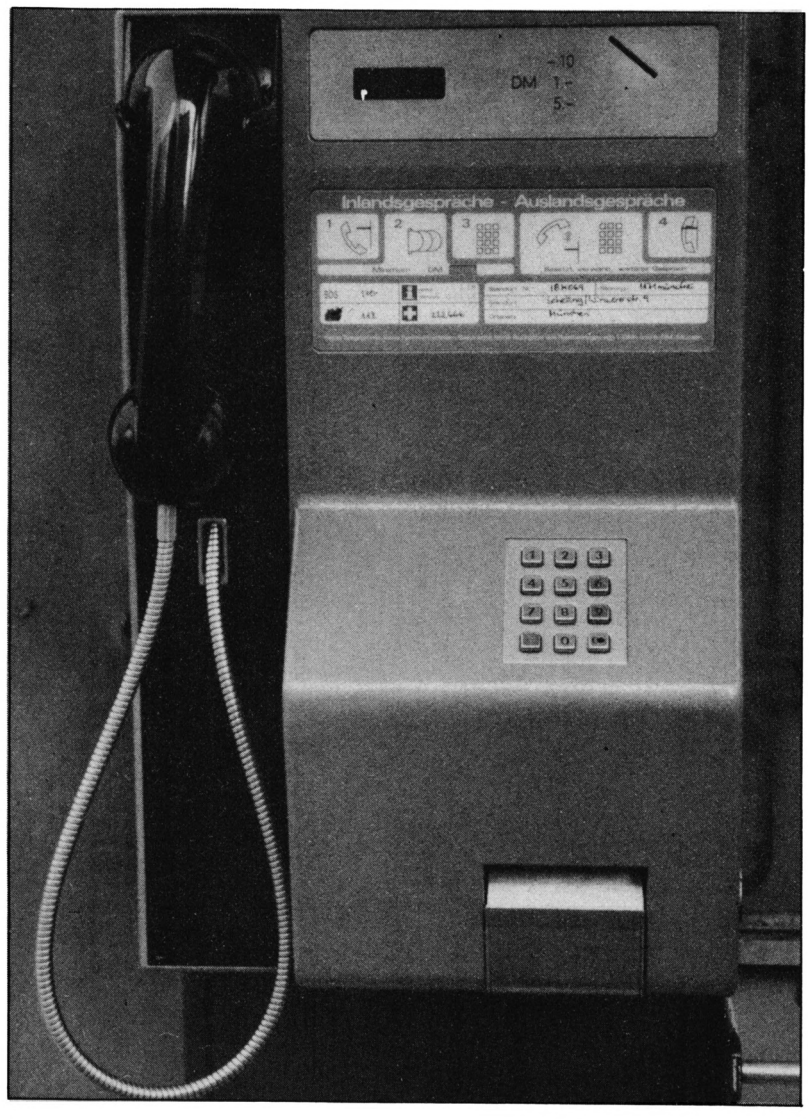
Beratung durch die Post

Jeder der drei Datel-Dienste hat Vor- und Nachteile. Wenn Sie die Möglichkeiten der DFÜ in größerem Umfang einsetzen wollen, so ist zunächst das Kosten-Nutzen-Verhältnis zu bestimmen. Bei jedem Fernmeldeamt gibt es einen sogenannten Datennetz-Koordinator (Postjargon: Danzko), der Ihnen Informationen und Entscheidungshilfen geben kann.
Die Freaks jedenfalls werden beim Bewährten bleiben. Die Investitionskosten für einen 300-Baud-Akustikkoppler belaufen sich auf etwa 200 Mark. Billiger geht's nicht. *AE*□

Der größte Computer der Welt

Jeder hat ihn zuhause

Kennen Sie den größten Computer der Welt? Aber sicher. Sie benutzen ihn täglich, nur ist es Ihnen nicht bewußt. Der größte Rechner, zumindest der mit der größten Anwenderzahl, ist nämlich das Telefonnetz.



Wie selbstverständlich greift heute jedermann zum Telefonhörer und wählt ein paar Ziffern. Ganz gleich, wen Sie wo erreichen wollen: Immer stellen Sie die Verbindung selbst her. Ist Ihnen eigentlich bewußt, was dabei geschieht?

Die Umstellung auf Großrechner läuft bereits

Die meisten Vermittlungsstellen der Post, sowohl in der Bundesrepublik als auch weltweit, haben als Hauptbestandteile noch immer elektromechanische Wähler, die direkt mit der Wählscheibe oder der Tastatur gesteuert werden. Zur Zeit wird zwar auf Rechner umgestellt, die diese Aufgaben schneller und hoffent-

lich auch besser erledigen können, doch das wird noch etwas dauern.
Aber auch die herkömmliche Technik ist schon ein kleines Wunderwerk, denn die Post treibt einen erheblichen Aufwand, um sicherzustellen, daß das Telefonnetz reibungslos funktioniert. Und fast immer erreichen Sie auch den Teilnehmer, den Sie haben wollten. Die meisten Fehlverbindungen kommen durch Falschwahl des Benutzers zustande. Fehler, die in den Einrichtungen der Post liegen, kommen zwar vor, sind aber sehr selten.

Nur acht Prozent können gleichzeitig telefonieren

Bereits wenn Sie Ihren Hörer abnehmen, geschieht einiges, von dem Sie überhaupt nichts bemerken: Es wird der zuständigen Vermittlungsstelle signalisiert, daß ein Gespräch geführt werden soll. Ein sogenannter Anrufsucher stellt fest, wer seinen Hörer abgehoben hat und stellt anschließend die Einrichtungen bereit, die Ihre erste Ziffer aufnehmen sollen. Im Normalfall stehen für je 100 Teilnehmer acht solcher Einrichtungen zur Verfügung. In der Praxis der Post haben sich diese acht Prozent als ausreichend erwiesen.

Der Neunte hat Pech

Allerdings kann es passieren, daß mehr als acht (von je 100) Teilnehmern Gespräche führen wollen. Wenn Sie der neunte sind, hören Sie nicht das Freizeichen, sondern das Besetzt-Zeichen.
Das ist jedoch hauptsächlich zu Weihnachten oder Silvester, wenn alle telefonieren wollen, der Fall. Wie so oft, mußten auch hier Kompromisse geschlossen werden. Sicher hätte die Post mehr als diese acht Gespräche möglich machen können. Technisch ist das kein Problem. Die Grenzen liegen in der Wirtschaftlichkeit. Oder wären Sie bereit, für doppelt so viele Abnehmerleitungen auch das Doppelte an Gebühren zu bezahlen? Wenn Sie Ihre gewünschte Rufnummer Zahl für Zahl eingeben, wird der Teilnehmer, den Sie erreichen wollen, immer mehr eingegrenzt, bis er zuletzt, nach Wahl der letzten Ziffer, eindeutig identifiziert ist. Jetzt können Sie die letzten Neuigkeiten austauschen oder sich zum Kino verabreden. Denken Sie aber daran, daß bei Ortsgesprächen tagsüber alle acht Minuten Ihr Zähler um eine Einheit (= 23 Pfennig) weitergeschaltet wird.

Wie geht das mit den Ferngesprächen?

Was passiert aber, wenn Sie ein Ferngespräch führen wollen? Sie wählen zuerst die Vorwahlnummer, die immer mit einer Null beginnt. Diese Null ist kein Bestandteil der Vorwahlnummer, sondern die, postintern so titulierte, Verkehrsausscheidungsziffer. Mit ihr als erster Zahl verlassen Sie Ihre Ortsvermittlungsstelle und befinden sich in der Fernebene. Und dort wird es interessant.

Immer hoch und runter

Nehmen wir an, Ihre eigene Ortsnetzkennzahl heißt 8441 und Sie wollen jemanden sprechen, der in dem Ortsnetz 4532 wohnt. Dann wird, nachdem Sie die 4532 gewählt haben, eine Leitung für Sie aufgebaut, die über die Knotenvermittlungsstelle 844, dann weiter über die Hauptvermittlungsstelle 84 zur Zentralvermittlungsstelle 8 führt. Von dort geht es weiter zur Zentralvermittlungsstelle 4 und dann stufenweise abwärts, bis Sie in der Zielvermittlungsstelle sind. Hier wird die gewünschte

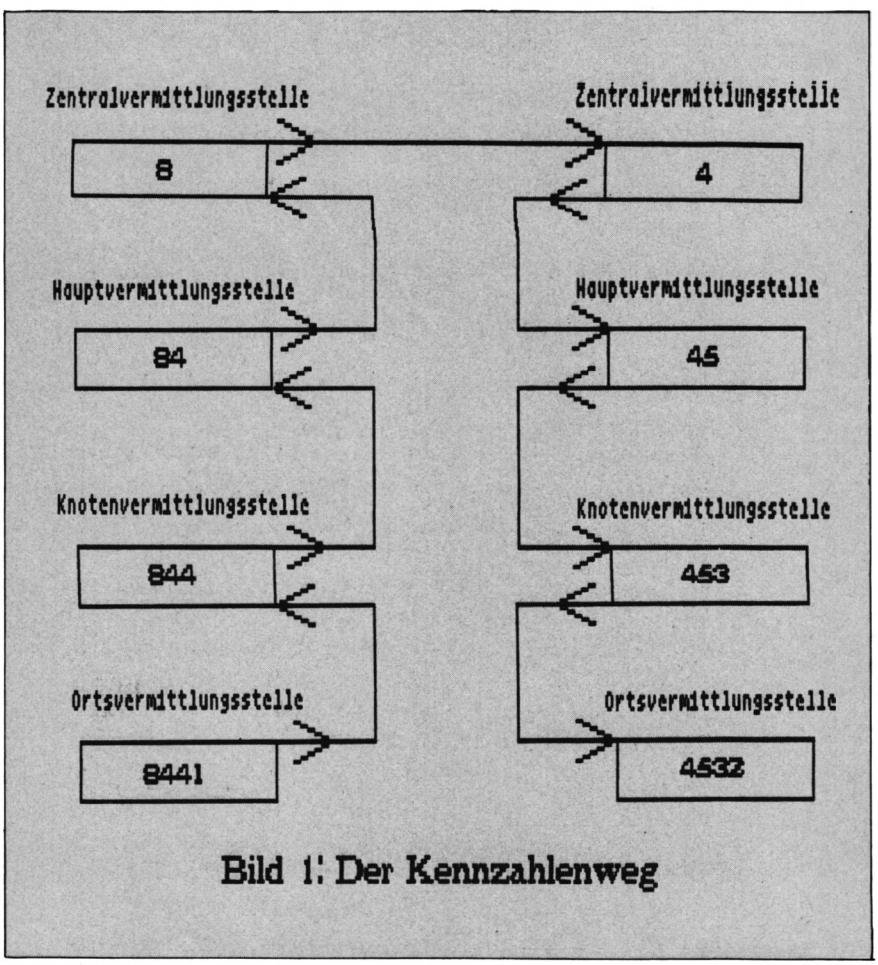


Bild 1: Der Kennzahlenweg

Rufnummer des fremden Ortes ausgewertet.
Dieser Weg ist in Bild 1 dargestellt. Sie können deutlich das stufenweise Weiterreichen Ihres Gesprächs erkennen. Dieser Weg heißt postintern Kennzahlenweg und war das ursprüngliche Grundkonzept beim Aufbau des Fernwahlnetzes in der Bundesrepublik.

Die Querverbindungen machen es möglich

Je mehr das Telefon in den fünfziger, sechziger und besonders in den siebziger Jahren Verbreitung fand und je mehr Ferngespräche geführt wurden, desto mehr Leitungen hätten im Kennzahlenweg bereitgestellt werden müssen.
Doch die Techniker der Post fanden einen anderen Weg.
Um nicht ständig Gespräche hierarchisch auf der einen Seite nach oben und auf der anderen wieder nach ganz unten zu vermitteln, wurden Querwege geschaffen, die auch direkte Leitungen zwischen den Knoten- und Hauptvermittlungsstellen zur Verfügung stellen (Bild 2). Im Laufe der Zeit wurde dieses Querleitungsnetz immer mehr vermascht, so daß heute der Kennzahlenweg die Ausnahme darstellt, da die meisten Gespräche über die Querwege abgewickelt werden. Doch davon bemerken Sie als Teilnehmer nichts.

Wer sucht, der findet

Nach Wahl der Ortsnetzkennzahl wird in den Ursprungs-Knotenvermittlungsstellen geprüft, ob ein Querweg zur Ziel-Knotenvermittlungsstelle vorhanden ist und ob eine Leitung dorthin frei ist. Ist dies nicht der Fall, so wird ein Weg zur Ziel-Hauptvermittlungsstelle überprüft. Ist dies erfolglos, so wird versucht, eine Leitung zur Ziel-Zentralvermittlungsstelle zu finden.
Schlagen all diese Versuche fehl, wird das Gespräch zur Ursprungs-Hauptvermittlungsstelle weitergereicht, und das gleiche Spiel beginnt dort von neuem. Führen auch diese Versuche nicht zum gewünschten Erfolg, so landet Ihr Gespräch in der Ziel-Zentralvermittlungsstelle. Auch dort werden wieder Querwege gesucht, die direkt zur Ziel-Knotenvermittlungsstelle führen.

Gleichzeitig rein und raus

Die Nummer, die Sie wählen, wird nämlich zwischengespeichert und erst dann weitergesandt, wenn die richtige Leitung zum Zielort gefunden wurde. Da dies elektromechanisch geschieht, hören Sie, wie der Rufnummernspeicher (so heißt dieser Zwischenspeicher) Ihre Nummer ausspeichert. Er kann übrigens nicht nur die gewünschte Rufnummer wieder ausgeben, sondern auch zur gleichen Zeit weitere Zahlen von Ihrem Apparat aufnehmen. Also gleichzeitig Daten senden und empfangen.
Die Leitungsbündel zwischen den einzelnen Haupt- und Zentralvermittlungsstellen bestehen nicht aus einem einzelnen Draht, sondern aus Richtfunkstrecken, über die mehr als zehntausend Gespräche gleichzeitig übertragen werden können. Und doch soll am Ende all dieser Leitungen der richtige Partner am Telefon sein. Der Aufwand, den die Post treibt, ist also erheblich. Vielleicht haben Sie jetzt ein bißchen mehr Verständnis, wenn Sie aus Ihrem Telefonhörer das Besetztzeichen oder die Ansage „Kein Anschluß unter dieser Nummer" hören.

ME □

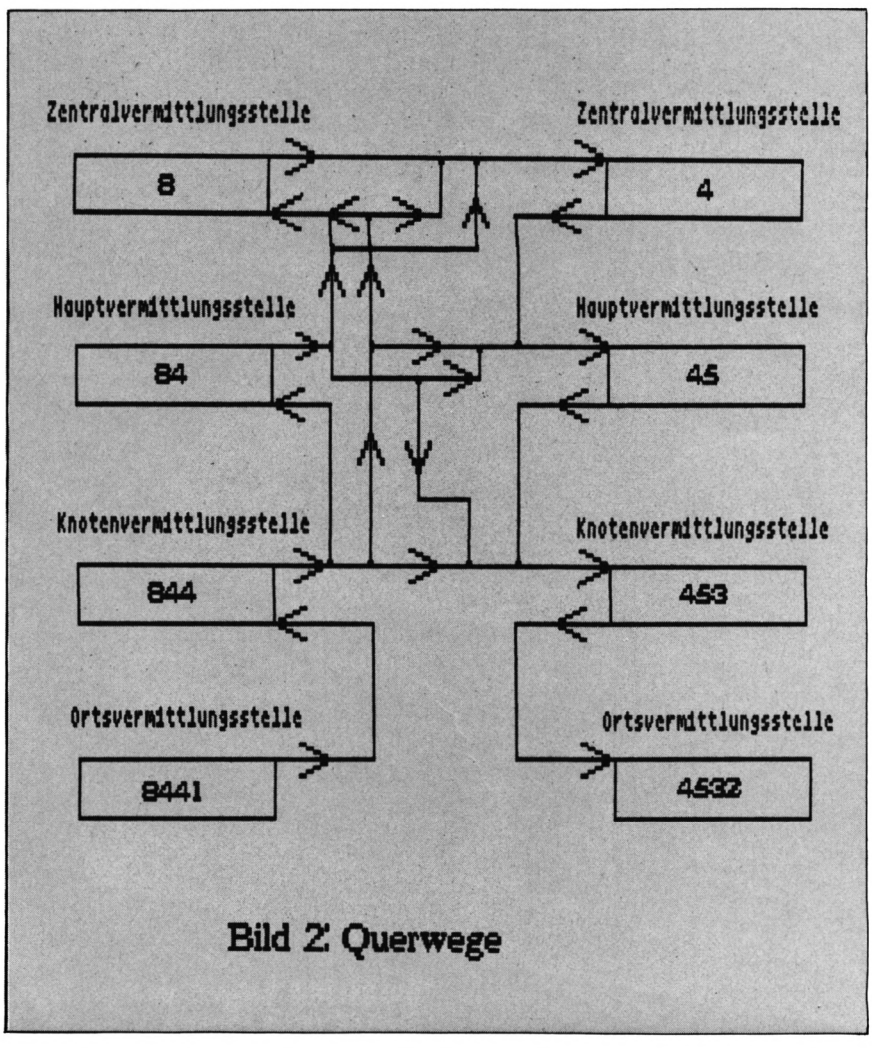


Bild 2: Querwege

Datennetze

Wenig Leistung für viel Geld?

Eine kritische Betrachtung der DFÜ

In dieser Ausgabe stellen wir Ihnen die drei wichtigsten Datennetze der Deutschen Bundespost vor. Bei aller gebotenen Objektivität drängt sich angesichts der Gebühren-Struktur dieser Netze die Frage auf: Lohnt sich DFÜ?



Ein DATEX-Endgerät der Post: Teures Geld für wenig Leistung

Datenfernübertragung ist mit Sicherheit eine praktische Sache. Schnell und in bestmöglicher Qualität erreichen Ihre Informationen den Computer, für den sie bestimmt sind – ob er nun einen oder tausend Kilometer entfernt ist. Ein Telefon und ein Akustikkoppler – oder ein DATEX-Endgerät – ist alles, was Sie und der Empfänger benötigen.
Computerfreaks bereitet es viel Spaß, in Mailboxen nach den neuesten Informationen zu stöbern oder persönliche Mails auszutauschen. Auch geht das „Downloaden" von Public-Domain-Programmen wesentlich schneller als auf dem Umweg über die Gelbe Post, also Brief- und Paket-Transport.
Und es kommt teurer. Veranschlagen wir großzügig eine Mark für das Briefporto, so kostet eine Disk vier Mark und zwei Tage Zeit. Ein Ferngespräch kostet im günstigsten Fall 23 Pfennige für jeweils 38 Sekunden – ebenfalls aufgerundet. Für die erwähnten vier Mark haben Sie also knapp elf Minuten Zeit, in eine Mailbox zu gelangen, das gewünschte Programm zu wählen und auf Ihren Rechner zu übertragen.
Wenn Sie einen Akustikkoppler verwenden, wird Ihnen dies kaum gelingen – selbst wenn Sie ausnahmsweise eine Leitung erwischen, auf der es nicht knackt und rauscht. Wenn Sie ein Modem verwenden, haben Sie zwar eine reelle Chance, das Zeitlimit einzuhalten – machen sich nach dem derzeitigen Fernmeldeanlagen-Gesetz aber strafbar. Die Post duldet nur ihre eigenen Geräte am Netz. Und sie kassiert dafür kräftig ab.
Eine Liberalisierung dieses Gesetzes wurde zwar in Aussicht gestellt, doch wird sie auf Kosten der bisher gewohnten Service-Leistungen gehen. Private Geräte werden nicht mehr von der Post repariert (oder „entstört", wie es im Post-Jargon heißt). Das ist einerseits verständlich, kann aber andererseits teuer werden, wenn der Geräte-Hersteller zufällig in Taiwan residiert. Der Endverbraucher wird über diesen Sachverhalt kaum informiert. Auch für die Bundespost ist es eben einfacher, sich als großen Reformator darzustellen, als auf die negativen Folgen solcher Reformen hinzuweisen.
Es sind ja auch nicht die privaten Telefon-Anschlüsse, abwertend Tante-Emma-Anschlüsse genannt, auf die man Wert legt; zumindest nicht im Bereich der DFÜ. Wo kämen wir auch hin, wenn jeder mit seinem Heimcomputer irgendwelche Daten quer durch die Republik schicken könnte?
Den DFÜ-Freaks wird das Leben so schwer – so teuer – wie möglich gemacht. Große Unternehmen bezahlen für den Anschluß an ein Datennetz zwar nicht weniger, aber das Kosten-Nutzen-Verhältnis gestaltet sich für sie völlig anders.
Ein privater Anwender wird kaum bereit sein, die horrenden Gebühren für einen DATEX-P-Anschluß zu bezahlen. Dabei würde dieser Dienst, wäre er breiter genutzt, große Vorteile bieten. So bleibt man halt bei Telefon und Akustikkoppler – mit all ihren Problemen.
Die angebotenen Informationsdienste, wie etwa Datenbanken, sind ebenfalls nicht für den privaten Nutzer gedacht. Ihr Inhalt ist auch nur in den seltensten Fällen interessant genug; doch fänden sich mit Sicherheit Anwender, wenn dieser Service einer breiteren Öffentlichkeit zugänglich gemacht würde.
Welcher Student etwa hat sich nicht schon gewünscht, innerhalb von Minuten eine Information zu bekommen, nach der er in der Bibliothek stundenlang sucht? Preiswerte DFÜ würde ihm den Zugriff auf eine Datenbank seines Fachgebietes erlauben – egal, ob der Rechner in Frankfurt, London oder Tokio steht.
Die Datenfernübertragung hat eine Zukunft, soviel ist sicher. Wie schnell und in welche Richtung sich die DFÜ entwickeln wird, hängt hauptsächlich von zwei Faktoren ab.
Faktor Nummer eins sind die Anbieter von Mailboxen und Datenbanken. Wenn sie ihr Angebot so gestalten, daß es auch für Privatleute attraktiv wird, ist der Bedarf geweckt.
Faktor Nummer zwei ist die Deutsche Bundespost. Auch sie gehört zu den Anbietern; doch Telebox und Btx bieten dem Privatmann nicht viel. Weitaus wichtiger ist jedoch eine Änderung der Gebühren-Struktur und die Zulassung leistungsfähiger (auch nichtposteigener) Modems. Erst dadurch erhält der vielzitierte Otto Normalverbraucher die Chance, von den angebotenen Dienstleistungen Gebrauch zu machen.

Fazit

Gerade von den Anbietern der Netze und Dienstleistungen ist immer wieder zu hören, wir befänden uns auf dem Weg in die Informationsgesellschaft. Es ist an der Zeit, daß sie auch für Fortschritte auf diesem Weg sorgen, wenn er nicht zu einer Sackgasse werden soll. *AE* □

Anzeige

Kfm. Komplettpaket
für alle Amstrad- u. Schneider-Computer
Hiermit bestellen wir gegen Nachnahme
für Maschine:
( ) Programm komplett (alle Teile) für **DM 198,–** inkl. dtsch. Handbuch) oder
( ) Finanzbuchhaltung mit Kontenblättern, GuV und Rohbilanz, freier Kontenplan,
( ) Kundenbuchhaltung mit Umsatzstatistik, OP's, Mahnungen, Forderungsliste,
( ) Lieferantenbuchhaltung mit Umsatzstatistik, OP's, Verbindlichkeitsliste,
( ) Lagerbestandsführung mit Umsatzstatistik, Lagerliste, Bestellvorschlagsliste,
( ) Auftragsbearbeitung mit Lieferscheinen, Fakturierung, Kundenrabatt, Auftragsrabatt,
( ) Sofortfakturierung mit Kundenrabatt, Auftragsrabatt, Formular frei aufbaubar,
für à **DM 39,90** + 1 x DM 10,– für deutsches Handbuch.

infosystems
SOFTWARE-VERTRIEBS GMBH
Dorrhof 7 · 4419 Laer · Telefon (02554) 1232
DIE HELFENDE HAND IM SOFTWARELAND!

Wegweiser in den Irrgarten

Schneider CPC BASIC-Wegweiser

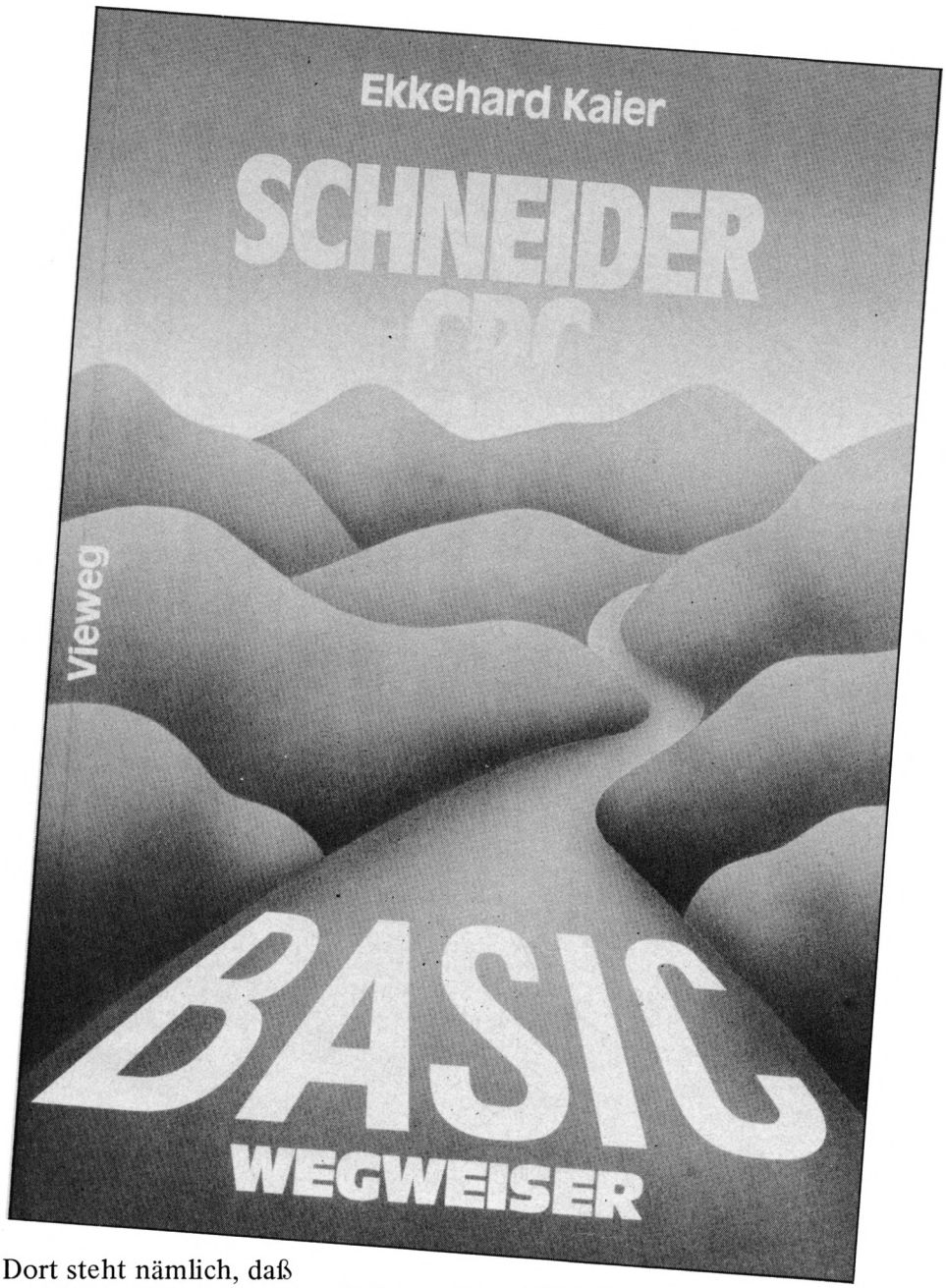
Von einem Buch für BASIC-Einsteiger erwartet man in erster Linie klare Aussagen und genaue Erläuterungen mancher im Handbuch zum CPC nicht eindeutiger BASIC-Befehle. Und wenn dann dem Anfänger noch Tips und Tricks in die Hand gegeben werden, um erfolgreich die ersten Programme zu schreiben, kann eine solche Lektüre nur empfohlen werden.

Lieber nicht lesen

Es gibt aber auch Bücher, die so verwirrend geschrieben sind, daß der Einsteiger sie bald enttäuscht weglegt. Zu dieser zweiten Kategorie gehört „Schneider CPC BASIC-Wegweiser" von Ekkehard Kaiser.
Was da so an Wissen, Halbwissen, Überflüssigem und Unsinn geschrieben wurde, haben wir schon lange nicht mehr gelesen.
Da heißt es zum Beispiel, daß die Drei-Zoll-Disketten unter AMS-DOS mit etwa 180 K-Byte formatiert würden. Warum wird hier das Wort „etwa" benutzt? Wußte es der Autor nicht genauer?
Auch für Sie, Herr Kaier, hier die genauen Angaben: Im CP/M-Format hat die Diskette 169 K-Byte, im DATA-Format 178 KByte Speicherkapazität. Und zwar genau diese Werte und nicht etwa „etwa".
An anderer Stelle wird erläutert, daß 5,25-Zoll-Disketten im IBM-Großrechner-Format 3740 77 Spuren hätten. Was soll dieser Absatz in einem BASIC-Buch für den Schneider CPC, der bekanntlich unter AMS-DOS arbeitet?

Unsinn hoch zwei

Auf den ersten 70 Seiten wird langatmig der Versuch unternommen, die Grundzüge der Datenverwaltung zu erklären. Wer das noch nicht weiß, sollte dieses Kapitel erst gar nicht lesen. Denn es enthält Dinge, die ein CPC-User wirklich nicht zu wissen braucht. Oder interessiert es Sie, wie die Blöcke bei den Commodore-Kassetten 2/3000 verwaltet werden und welchen Schrifttyp die Klarschriftleser bei den Banken benötigen?
Auf Seite 73 findet sich ein genialer Tip, wie man als User preiswert an professionelle Software kommt. Der Autor empfiehlt, sich mit einem EPROM-Brenner in ein Softwaregeschäft zu begeben, dort ein Programm auszusuchen und dies in sein EPROM zu brennen. Abgesehen davon, daß dies bei geschützten Programmen kaum geht und der Software-Verkäufer dies auch nicht erlauben wird, da ihm dadurch ein Geschäft entgehen würde, handelt es sich hier um eine Anleitung zum Raubkopieren. Und so etwas sollte nicht in einem Handbuch für Computer-Neulinge stehen.
Mit Überraschung haben wir auch die Überschrift zum Kapitel 2.5 gelesen.



Dort steht nämlich, daß man beim CPC auch mit dem Betriebssystem MS-DOS arbeiten könne.

MS-DOS beim CPC?

Der Autor schränkt das zwar ein, indem er darauf hinweist, daß man dazu einen 8088-Prozessor benötige, aber ansonsten würde dies schon gehen. Was für ein Unsinn! Wir haben noch nie einen CPC unter MS-DOS erlebt. Meint der Autor vielleicht, man solle einen CPC nehmen, die Innereien entfernen, die entsprechenden Karten einbauen und schon liefe MS-DOS? Das mag vielleicht sein, aber dann ist es eben kein CPC mehr, sondern ein 16-Bit-Rechner in einem CPC-Gehäuse.
Als Krönung des Ganzen werden noch einige BASIC-Befehle erläutert, die sicher den meisten CPC-Usern unbekannt sein dürften. PUT gehört ebenso dazu wie COMMON. Auf unserem CPC erscheint nach Eingabe dieser Schlüsselworte immer wieder die lakonische Bildschirmmeldung
„Syntax error".
Der CPC weigert sich, diese Befehle – die es wohl auf anderen Rechnern gibt – zu akzeptieren. Ob der Autor den CPC überhaupt schon einmal gesehen hat? Vielleicht, aber dann sicher nur im Schaufenster eines Computer-Händlers. Ganz klein gedruckt wird dann dem Leser mitgeteilt, daß diese Befehle von MBASIC stammen und zu Beginn erst von der Diskette geladen werden müssen. Wer aber arbeitet beim CPC mit MBASIC?
Die Liste der falschen Aussagen ließe sich beliebig erweitern. Aber wir haben dieses Buch schnell wieder weggelegt und können Ihnen nur empfehlen, sich statt eines solchen Wegweisers, der in die Sackgasse führt, ein paar Comic-Hefte zu kaufen. Damit haben Sie wenigstens etwas zu lachen; bei dem Buch dagegen kommen Ihnen die Tränen.
Übrigens gibt es vom Autor Ekkehard Kaier die gleichen Wegweiser auch für andere Rechnertypen, zum Beispiel für die MSX-Maschinen, Commodore 116, 16, Plus/4 und den Amiga. Der Umschlag sowie die Einleitung und die ersten Kapitel sind bei allen Büchern identisch, auch der Inhalt ist gleichartig aufgebaut. Unterschiede sind nur stellenweise vorhanden. So kann man aus einem Manuskript gleich mehrere Bücher machen. Ob die Qualität der anderen Werke dadurch größer geworden ist, darf bezweifelt werden. *ME* □

Ekkehard Kaier:
Schneider CPC BASIC-Wegweiser
Braunschweig 1986
Verlag Vieweg
300 Seiten
ohne Preisangabe
ISBN 3-528-04471-3

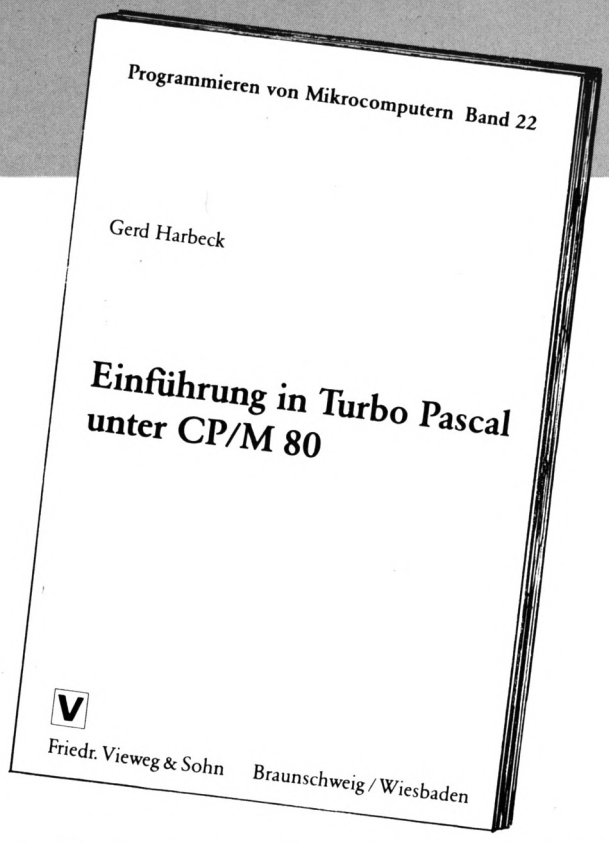
Einführung in Turbo Pascal unter CP/M 80

Lehrbuch oder Nachschlagewerk?

Bereits im Jahr 1986 erschienen ist die Einführung in Turbo Pascal unter CP/M 80 von Gerd Harbeck. Da es seither keine neue Version von Turbo Pascal für CP/M-Systeme wie den CPC gegeben hat, kann es jedoch immer noch als aktuell angesehen werden.
Der Autor beginnt mit einer Beschreibung der CP/M-Befehle und -Dienstprogramme, die Sie für die Anwendung der Sprache Turbo Pascal benötigen. An diese Kurzeinführung schließt sich die Darstellung der einzelnen Compiler-Funktionen an.
Sobald es jedoch an die Bedienung des Turbo-Editors geht, sollten Sie dem Werk kritisch gegenüberstehen. Die dargestellten Tasten-Kombinationen sind nämlich die Standard-Belegungen. CPC-Anwender erhalten jedoch eine vor-installierte Version des Compilers, bei der bereits der Cursorblock und die COPY-Taste berücksichtigt sind. Außerdem können die Belegungen mit Hilfe des TINST-Programms schnell geändert werden.
Die Tabellen, die an dieser Stelle im Buch abgebildet sind, weisen eine geradezu frappierende Ähnlichkeit mit denen des Original-Handbuchs auf. Abgesehen davon, daß die Darstellung der Standard-Tastenbelegung keine wertvolle Hilfe für den Benutzer ist, hätte ein entsprechender Verweis genügt. Hier werden Seiten verschwendet, die an anderer Stelle fehlen.

ASCII für Fortgeschrittene

Wer denkt, mit der Überschrift „Sprachelemente von Turbo Pascal" gehe es auf Seite 28 endlich los, wird enttäuscht. Obwohl das Buch laut Klassifizierung auf dem Umschlag für Fortgeschrittene gedacht ist, fühlte sich der Autor bemüßigt, eine mehr oder weniger ausführliche Abhandlung über den Zeichenvorrat eines Computers zu verfassen.
Das Chaos wird komplett durch eine ASCII-Tabelle, in der alle Codes außer im Dezimal-System auch noch in hexadezimal und dual dargestellt sind. Wenn einem fortgeschrittenen Anwender erklärt werden muß, welche Zeichen er auf seiner Tastatur findet, dann sollten auch diese beiden Zahlensysteme erläutert werden. Deren Kenntnis setzt der Autor aber anscheinend als Grundwissen voraus.
Die Abhandlung des Pascal-Sprachumfangs bewegt sich in dem aus anderen Lehrbüchern sattsam bekannten bescheidenen Rahmen. Zunächst werden auf sechs Seiten die reservierten Wörter dieser Programmiersprache sowie die Konventionen für die



Schreibweise von Bezeichnern vorgestellt.

Informations-Überfluß

Wer sich das alles auf einmal merken kann, muß schon mit einem hervorragenden Gedächtnis gesegnet sein. Ein normaler User kann es sicherlich nicht und atmet erleichtert auf, wenn auf Seite 39 endlich der Aufbau eines Programms beschrieben werden soll. Doch auch in diesem Kapitel muß sich der Leser auf den sprichwörtlichen Sturz ins kalte Wasser gefaßt machen. Die einzelnen Teile eines Pascal-Programms werden zwar vollständig und korrekt beschrieben; wer vorher aber nur BASIC kannte, verliert hier endgültig den Durchblick.
Viele der entstehenden Unklarheiten werden beim Weiterlesen beseitigt; doch in dem Bestreben, alle Punkte vollständig zu beschreiben, hat der Autor wohl zuviel des Guten getan. Eine schrittweise Einführung anhand von Beispielen wäre günstiger gewesen.
Nachdem der Leser sich über fünfzig Seiten hinweggequält hat, findet er endlich das erste Beispiel-Programm. Vorausgesetzt, Sie haben diese fünfzig Seiten auswendig gelernt und haben zudem noch die Fähigkeit, aus dem gesammelten Wissen das Wesentliche zu extrahieren, werden Sie dieses Programm sogar verstehen.

Spaghetti-Code in Reinkultur

An dieser Stelle ist ein weiterer Kritikpunkt anzuführen. Pascal ist eine Programmiersprache, mit der hervorragend strukturierte Programme erstellt werden können. Der Autor schafft es jedoch, in seinen Listings ein Höchstmaß an Unübersichtlichkeit zu produzieren.
Insgesamt erwecken die Programme den Eindruck, als wären sie auf einer Schreibmaschine getippt und im Buch verkleinert wiedergegeben worden. Bereits beim ersten Durchlesen fanden wir einen Fehler (Seite 63, „than" statt „then"). Eine ausführliche Suche haben wir uns erspart.
Zur Unlesbarkeit der Programme trägt auch die durchgehende Verwendung der Pascal-Ersatzzeichen für Kommentare und Indizes bei. Anstelle der bekannten geschweiften und eckigen Klammern werden die Kombinationen „(*" und „(." verwendet. Das kann nicht daran liegen, daß man eckige und geschweifte Klammern nicht hätte setzen können. Auf Seite 29 sind diese Zeichen nämlich zu finden. Nach der Methode „Nürnberger Trichter" geht es auch weiter.
Fast jeder Pascal-Befehl wird kurz beschrieben. Als Beispiele dienen ellenlange Programme, die nur dann verständlich sind, wenn die zuvor theoretisch erworbenen Kenntnisse auch verfügbar sind.
Wichtige Themen wie Rekursion oder dynamische Datenstrukturen werden auch hier nur kurz angerissen. Die Sprache Pascal läßt sich auf 160 Seiten nicht vollständig beschreiben; ein Pascal-Programmierer sollte dies wissen und sich von Anfang an Beschränkungen auferlegen.

Mißglückter Versuch

Der Versuch, in einem Lehrbuch möglichst viele Informationen unterzubringen, ist auch hier gescheitert. Anstelle praktischer Beispiele findet der Leser trockene Beschreibungen, die er ebensogut dem Handbuch entnehmen kann.
Der Autor Gerd Harbeck ist Oberstudiendirektor an einem Gymnasium. Als solcher sollte er gewisse didaktische Techniken beherrschen. In diesem Werk ist davon jedoch nichts zu bemerken.
Die Ausgabe von 32 Mark für dieses Buch lohnt sich nur dann, wenn Sie es als Nachschlagewerk verwenden wollen. Ihrem Compiler ist jedoch ein Turbo-Pascal-Lexikon in Form eines Handbuchs beigefügt. Da die „Einführung in Turbo Pascal" als Lehrwerk jedoch ungeeignet ist, sollten Sie das Geld lieber in die Leerdisketten investieren, die der Turbo-Compiler so zahlreich benötigt. *AE* □

Gerd Harbeck:
Einführung in Turbo Pascal unter CP/M 80
Braunschweig 1986
Verlag Vieweg
160 Seiten
DM 32,–
ISBN 3-528-04440-3

LESERBRIEFE

In eigener Sache

Ausnahmsweise wollen wir diese Dialog-Seiten nicht mit einem Leserbrief beginnen, sondern mit einer Aufforderung an Sie, sich aktiv an der Gestaltung Ihrer CPC WELT zu beteiligen. Schreiben Sie uns, wenn Sie Probleme mit unseren Listings oder Beiträgen haben. Teilen Sie uns mit, was Sie an der CPC WELT gut finden und was noch verbessert werden könnte. Wir sind für jede sachliche Kritik dankbar. Und schließlich würden wir uns auch freuen, Ihre Meinung zu diesem oder jenem speziellen Beitrag zu erfahren. Als kleinen Anreiz, uns zu schreiben, verlosen wir unter den Einsendungen, die uns bis zum Redaktionsschluß der nächsten Ausgabe erreichen, zehn CPC-Spiele auf Diskette. Einzige Bedingung für die Teilnahme an der Verlosung ist, daß Sie uns einen Leserbrief zu einem der oben genannten Themen schreiben. Der Satz „Ich will ein Spiel gewinnen" genügt nicht. Die Teilnahme an der Verlosung ist nicht von einer Veröffentlichung Ihres Briefes abhängig. Der Rechtsweg ist ausgeschlossen.
Doch genug der langen Vorworte. Kommen wir zu Ihren Briefen . . .

Harte Kritik

Zwar schätze ich es sehr, daß auch die CPC-WELT Disketten mit den aufgeführten Programmen verschickt, so wie es verschiedene andere Zeitschriften auch tun. Daß diese Disketten mit den fertigen Programmen verhältnismäßig teuer sind, kann ich noch verstehen: Jeder will das meiste rausholen. Doch daß dabei nicht alle vorgestellten Programme auf dieser Diskette vorhanden sind, das finde ich schon etwas stark. Darf man erwarten, daß in Zukunft jedes Programm einzeln bestellt und bezahlt werden muß?

Klaus Pillath,
4130 Moers 1

Um zu Anfang gleich Ihre Frage zu beantworten: Nein, mit Sicherheit nicht.
Auch trifft Ihr Vorwurf nicht zu, daß nicht alle vorgestellten Programme auf unserer Softbox enthalten seien. Normalerweise befinden sich alle BASIC-Listings auf der Diskette, und zwar – soweit der Platz ausreicht – sowohl in der im Heft abgebildeten Form als auch in einer lauffähigen Version. Gelegentlich liefern wir aus Platzgründen nur die lauffähige Version, also Assembler-Programme ohne die abgedruckten BASIC-Lader. Damit kommen wir dem Wunsch vieler Leser entgegen, die sich die Mühe ersparen wollen, erst eine lauffähige Version erzeugen zu müssen.
In Ausgabe 5/88 haben wir eine Ausnahme von dieser Regel gemacht. Die Turbo-Pascal-Toolbox mußte extra bestellt werden. Das geschah deshalb, weil nicht alle Leser den Turbo-Compiler besitzen. Wer nur in BASIC programmiert, hätte sonst für Programme bezahlen müssen, die er überhaupt nicht verwenden kann. Eine Ausnahme stellen auch umfangreiche Programme dar wie unser Listing des Monats aus Ausgabe 6/88. Es beansprucht allein zwei Diskettenseiten, so daß wir die Softbox aufteilen mußten.
Wenn Sie bedenken, daß für jede Softbox eine Master-Diskette gezogen werden muß und die Disketten, um Fehler möglichst auszuschließen, einzeln kopiert werden, wird Ihnen der Preis vielleicht angemessen erscheinen. Sollten Sie eines der seltenen fehlerhaften Exemplare erhalten haben, so tauschen wir die Diskette anstandslos und schnellstmöglich um – auch dies ist für uns ein wichtiger Bestandteil dessen, was wir unter Leser-Service verstehen.

Hardcopy-Anpassung für CPC DRAW

Das in CPC SPECIAL 5/88 abgedruckte Programm CPC DRAW war für mich das, was ich schon lange suchte. Ich tippte das Listing natürlich voller Erwartung sofort ab. Nachdem ich kleinere Tippfehler beseitigt hatte, lief das Programm auch einwandfrei.
Als ich aber die erste Hardcopy ausdrucken wollte, trat genau das ein, was ich befürchtet hatte: Die eingestellten Druckercodes ignorierte mein DMP 2000 zwar nicht völlig, aber doch so weit, daß das Ergebnis für mich leider unbrauchbar ist. Mit der im Heft abgedruckten Anleitung zur Druckeranpassung komme ich nicht zurecht. Ich wäre Ihnen sehr dankbar, wenn Sie mir helfen könnten.

Ferdinand Horbach,
7240 Horb 1

Wir können helfen. CPC DRAW ist auf einen Speedy 100-80 eingestellt, der leider nicht in allen Punkten Epson-komptaibel ist.
Die Speicher-Adressen, in denen die Steuercodes für den Drucker enthalten sind, können Sie der

Aufstellung auf Seite 83 der Ausgabe 5/88 entnehmen. Die Anpassung an den CMP 2000 und andere Epson-kompatible Drucker können Sie wie folgt vornehmen: Geben Sie nach dem Einschalten oder Zurücksetzen des Rechners ein:
MEMORY 18000:
LOAD"MCDRAW.BIN"
Nun können Sie mittels einzelner POKE-Befehle die Steuersequenzen für den Drucker ändern. Beachten Sie, daß am Ende der Sequenz jeweils eine Null stehen muß. Beginnen Sie also mit dem Abschalten des Sprungs über die Seitenperforation. Auf Ihrem Drucker lautet der entsprechende Code (in BASIC): CHR$(27);"N"; CHR$(0) oder dezimal 27-78-0. Mit der abschließenden Null ergeben sich folgende POKEs:
POKE &9B00,27:
POKE &9B01,78:
POKE &9B02,0:
POKE &9B03,0
Der Zeilenabstand in n/72 Zoll wird auf dem DMP 2000 mit der Sequenz CHR$(27);"A"; CHR$(n) festgesetzt, dezimal also 27-65-n. Da hier 7/72 Zoll vorgesehen sind, steht statt des n der Wert 7:
POKE &9B04,27:
POKE &9B05,65:
POKE &9B06,7:
POKE &9B07,0:
POKE &9B08,0
Eine Änderung des Bit-Grafik-Modus ist für den DMP 2000 nicht nötig. Somit können Sie die geänderte Datei jetzt abspeichern mit
SAVE"MCDRAW.BIN", B,&9000,4424
Setzen Sie den Rechner anschließend zurück und starten Sie CPC DRAW erneut. Die Hardcopy läuft jetzt auf allen Epson-kompatiblen Druckern.

DATA-Lader

Bei unserer Hotline erhalten wir immer wieder Anfragen von BASIC-Einsteigern, die Probleme beim Abtippen mehrteiliger Programme haben. Für uns ist dies Anlaß, an dieser Stelle noch einmal auf diese Listings einzugehen.
Unsere Programme haben einen standardisierten Kopfteil: die Kommentare zu Beginn jedes Listings. Finden Sie in diesem Kopf den Hinweis
"Erzeugt DATEINAME.XXX"
so bedeutet dies, daß das abgedruckte Programm nach dem Start mit RUN automatisch eine zusätzliche Datei auf Ihrer Diskette generiert. Speichern Sie das BASIC-Listing, das Sie abgetippt haben, unter einem beliebigen Dateinamen – nicht aber unter dem Namen, den Sie im Kopf finden. Geben Sie dann den Befehl
RUN
ein. In der Regel sehen Sie nichts auf dem Bildschirm, lediglich das Laufwerk läuft kurz an. Sobald der Rechner sich mit
Ready.
zurückmeldet, ist die benötigte Datei angelegt. Erst wenn Sie alle DATA-Lader auf diese Art und Weise abgearbeitet haben, können Sie das Hauptprogramm starten. Die von den Ladern erzeugten Dateien werden automatisch nachgeladen. Es ist daher wichtig, daß Sie für alle Dateien dieselbe Diskette verwenden.
Wenn Sie mit Kassette arbeiten, müssen Sie zusätzlich auf die Reihenfolge der Dateien achten. In der Regel finden Sie im Text einen Hinweis, in welcher Reihenfolge die Files geladen werden. Es kann vorkommen, daß diese Reihenfolge aus technischen Gründen nicht mit der Abfolge der Listings im Heft übereinstimmt. Lassen Sie hier also besondere Sorgfalt walten.
Beim Abtippen von DATA-Ladern ist aufgrund der großen Menge an Zahlenkolonnen die Verwendung des Checksummers anzuraten. Bevor Sie das Hauptprogramm starten, sollten Sie aber mit CTRL-SHIFT-ESC den Rechner zurücksetzen. Maschinensprache-Programme können mit den Resten des Checksummers kollidieren und unvorhersehbare Ergebnisse erzeugen.

Programmaufrufe mit Turbo Pascal

Gelegentlich benutze ich Turbo Pascal zum Erstellen von Programmen. Es hat sich gezeigt, daß nach der Benutzung von Turbo Pascal und dem anschließenden Aufruf von WordStar oder dBASE mein CPC 464 abstürzt. Ebenso verhält es sich, wenn ich Turbo-COM-Files starte und danach eines der genannten Programme aufrufe. Ein erneutes Booten (^C) nützt nichts.
Ich würde aber gerne eine Benutzer-Oberfläche unter Turbo Pascal schreiben, mit der ich alle anderen CP/M-Programme aufrufen kann. Ist Ihnen zu diesem Thema etwas bekannt?

Günther Behrendsen, 2086 Ellerau

Beim Aufruf von COM-Files aus Turbo Pascal ergeben sich unter CP/M 2.2 einige Probleme. Da diese Optionen unter CP/M Plus einwandfrei funktionieren, dürfte dies an einigen Unzulänglichkeiten des Betriebssystems liegen.
Mit dem Execute-Befehl des Turbo-Compilers können Sie die meisten Anwender-Programme problemlos aufrufen.
Das einzige Problem ist die Rückkehr in die von Ihnen gewünschte Benutzer-Oberfläche. Die Programme terminieren mit einer Rückkehr ins CP/M oder mit einem Absturz des Rechners.
Allgemein läßt sich sagen, daß die Realisierung einer Benutzeroberfläche unter Turbo Pascal nur mit tiefgreifenden Änderungen des Betriebssystems realisiert werden kann. Diese Änderungen sind im Prinzip möglich; Sie werden jedoch Verständnis dafür haben, wenn wir Ihnen keinen konkreten Lösungsweg vorschlagen können. Dies bedürfte auch von unserer Seite umfangreicher

Recherchen und Experimente.

Checksummer fehlte

Ich habe aus der CPC-WELT 5/88 das Listing CPCDRAW eingegeben und festgestellt, daß dieses Programm auf meinem Computer nicht läuft. Der DATA-Lader erzeugt keine Binärdatei, und die Symbol-Datei ELO.SYM wird nicht geladen. Es müssen also noch Fehler im Listing vorhanden sein. Ich bitte Sie um Zusendung einer Korrektur.
Ihre Zeitschrift habe ich jetzt erstmals gekauft und bin mit den Beiträgen und Listings sehr zufrieden. Leider fehlt mir das Listing Ihres Checksummers.

Friedhelm Ziese,
5990 Altena 8

Beim Listing CPCDRAW können wir leider nicht helfen. Es ist in der Form, wie es in Ausgabe 5/88 abgedruckt ist, fehlerfrei. Davon haben wir uns in umfangreichen Tests überzeugt. Wir empfehlen Ihnen daher, die in Frage kommenden Listings und Programmzeilen nochmals zu überprüfen. Alle Listings erhalten Sie selbstverständlich auch auf unserer Softbox.
Der Checksummer befindet sich normalerweise in jeder Ausgabe – außer in Ausgabe 5, was auf einen Irrtum des verantwortlichen Redakteurs zurückzuführen ist. Wir bitten an dieser Stelle um Entschuldigung und werden uns bemühen, Ihnen in Zukunft den gewohnten Checksummer zu liefern.

CLUBECKE

Die PDUG stellt sich vor

Das Angebot der Public Domain User Group (PDUG) ist breit gefächert. Neben zahlreichen Public-Domain-Programmen für CPC, Joyce und PC wird auch ein Konvertierungs-Service von Drei-Zoll-CP/M-Disketten auf 5,25-Zoll-MS-DOS-Disks und andere 5,25-Zoll-Formate geboten.
Wer sich mit eigenen Programmen an der deutschen Public-Domain-Szene beteiligen möchte, wird gebeten, sich an den Club zu wenden. Informationen sind erhältlich gegen 1,60 Mark in Briefmarken; eine Katalog-Diskette für PCs ist gegen Marken im Wert von zwei Mark erhältlich. Die Anschrift lautet:

PDUG,
Postfach 1118,
6464 Linsengericht



Computer-Stammtisch

Der Computer-Stammtisch Mannheim besteht schon seit mehreren Jahren. Zu den Club-Aktivitäten gehören monatliche Treffen sowie ein Club-Blatt. Der Stammtisch trifft sich jeden ersten Samstag im Monat in Mannheim. Wer mehr Informationen haben will, wendet sich an

Michael Mertel,
Postfach 1252
6805 Heddesheim

Bezugsquellen – Was gibt es wo? Von CPC-WELT getestet

Wir wollen hier keine Gratiswerbung betreiben – es finden sich auch die Vertreiber jener Soft- und Hardware, die bei uns nicht auf Gegenliebe stieß. Aber ein Testbericht ist völlig sinnlos, wenn wir nicht auch mitteilen, wo das Besprochene zu beziehen ist. Leider können nicht alle Händler berücksichtigt werden, die die Ware im Sortiment führen. Dies soll nur ein letzter Hinweis für denjenigen Interessenten sein, der bei seinem Fachhändler auf leere Regale stieß.

Balloon (1/88)
Asphalt (1/88)
Fußball Manager (1/88)
Reisende im Wind 2 (1/88)
Sidewalk (1/88)
Renegade (1/88)
Classix 1 (1/88)
Renegade (2/88)
Guild of Thieves (1/88)
Trantor (1/88)

ariolasoft GmbH
Postfach 1350
4830 Gütersloh

Firetrap (2/88)
Trantor (2/88)

Activision Deutschland GmbH
Postfach 760680
2000 Hamburg 76

C-Compiler (2/88)

Arnor (Deutschland)
Hans-Henny-Jahn-Weg 21
2000 Hamburg 76

Henschke Datentechnik
Aidlinger Weg 6
7034 Gaertingen

James Bond – Der Hauch des Todes (1/88)

Rushware GmbH
An der Gümpgesbrücke 24
4044 Kaarst 2

Multiface II

Weeske Computer Elektronik
Potsdamer Ring 10
7150 Backnang
und
Romantic Robot (Goldscheider)
Ben Gurion Ring 80
6000 Frankfurt 56

BTX Modul zum CPC 464 (2/88)
Drucker LQ 3500 (1/88)

Im Fachhandel

dk'tronics Silicon Disc (6/88)
dk'tronics Sprach synthesizer (7/88)

Weeske Computer & Elektronik
Potsdamer Ring 10
7150 Backnang

Plonker-Box (6/88)

Lindy-Elektronik GmbH
Karl-Kuntz-Weg 9
6800 Mannheim 25

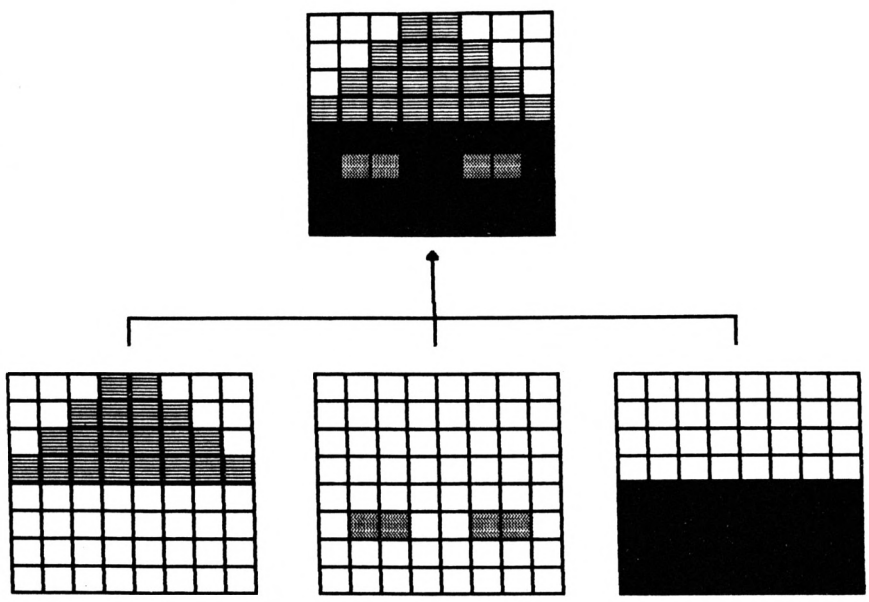
Grafik-Kurs Teil 4

Unbegrenzte Möglichkeiten

Grafikmodi und Koordinatensysteme

In den letzten drei Folgen dieses Kurses haben Sie die wichtigsten Grafikbefehle und ihre Funktionsweisen kennengelernt. Mit der Vorstellung der verschiedenen Grafik-Modi des CPC und einer Anwendung für mathematisch Interessierte soll diese Serie abgeschlossen werden.

Mit Hilfe der SYMBOL-Anweisung können Sie neue Zeichen definieren. Innerhalb seiner Matrix aus acht mal acht Punkten ist ein Character nur in einer Farbe darstellbar. Die Darstellung mehrfarbiger Zeichen ist aber möglich, wenn Sie einen kleinen Trick anwenden.
Im Textmodus unterscheidet der Schneider CPC zwei Darstellungsarten: undurchsichtige und durchsichtige. Diese beiden sind auch als Opaque- und Transparent-Modus bekannt. Der Opaque-Modus ist die Standard-Einstellung. Er bewirkt, daß jedes Zeichen, das an einer Textposition abgebildet wird, den Hintergrund vollständig überschreibt.



Durch Überlagerung können Sie im Transparent-Modus mehrfarbige Zeichen erzeugen

PRINT beeinflußt die Grafik

Diese Voreinstellung können Sie ändern, indem Sie das entsprechende Steuerzeichen einsetzen. Es hat den ASCII-Wert 22 und erfordert einen Parameter: Null für die Aktivierung des Opaque-Modus, Eins für Umschaltung auf Transparent-Modus.
Der Befehl

PRINT CHR$(22);CHR$(1);

bewirkt, daß nur die gesetzten Punkte eines Characters den Hin-

tergrund mit der aktuellen PEN-Farbe überschreiben. Alle Bildpunkte, die in der Matrix-Definition mit Null belegt wurden, bewirken keine Veränderung auf dem Bildschirm.
Durch diese Eigenschaft des Transparent-Modus können Sie mehrfarbige Zeichen abbilden. Für jede Farbe, die dabei verwendet werden soll, müssen Sie einen eigenen Character bestimmen. Besitzt die gewünschte Figur drei Farben, so sind drei SYMBOL-Definitionen notwendig. Anschließend müssen Sie die Zeichen übereinander abbilden, wobei Sie vor jeder Bildschirm-Ausgabe einen neuen PEN setzen.
Listing 1 zeigt dieses Verfahren anhand eines Beispiels. Auf dem Monitor wird ein kleines Haus ausgegeben.
Wie das Zeichen zustande kommt, sehen Sie in Abbildung 1: Drei Zeichen werden in verschiedenen Farben an derselben Position dargestellt. Dafür sorgt der Steuercode 8 in den Zeilen 210 und 220. Er bewegt den Cursor um eine Schreib-Position nach links.
Wenn Sie in einem Programm den Transparent-Modus verwenden, dürfen Sie nicht vergessen, wieder auf Opaque umzuschalten. Fehlt die entsprechende Anweisung, so bleibt die Einstellung über das Programmende hinaus erhalten. Wenn Sie dann etwa eine Zeile editieren und dabei ein Zeichen löschen, schiebt sich der rechte Teil um ein Zeichen nach links. Dabei wird der Hintergrund nicht gelöscht, so daß die Zeile nicht mehr lesbar ist.
Der Transparent-Modus ist nach dem Befehl TAG wieder abgeschaltet. Wenn Sie einen Steuercode an einer Grafikposition ausgeben, wird sein druckbares Äquivalent dargestellt, ohne daß das Zeichen die gewünschte Wirkung hätte. Daher ist es nicht möglich, den Transparent-Modus innerhalb einer Grafik zu verwenden.
Für die Grafik-Darstellung existieren statt dessen vier weitere Modi,

abgebildeter Punkt	zu setzender Punkt	Ergebnis
gesetzt	gesetzt	nicht gesetzt
gesetzt	nicht gesetzt	gesetzt
nicht gesetzt	gesetzt	gesetzt
nicht gesetzt	nicht gesetzt	nicht gesetzt

XOR ist eine logische Funktion, mit der Sie vor allem in der Grafik interessante Effekte erzielen

die mit dem Code 23 und einem Parameter von null bis vier angesprochen werden. Sie bestimmen, auf welche Art ein neu zu setzender Bildpunkt mit dem Hintergrund verknüpft werden soll. Die Voreinstellung entspricht dem Befehl

```
PRINT CHR$(23);CHR$(0);
```

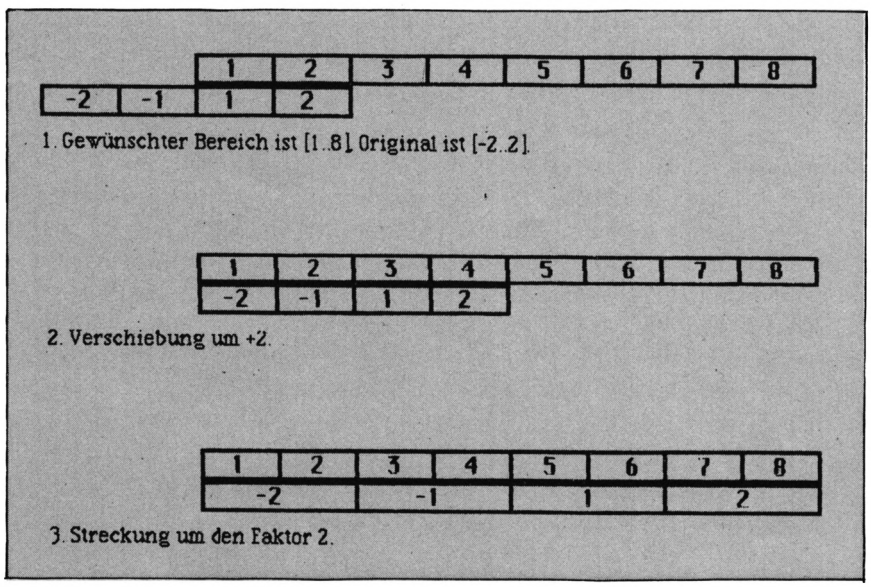
Damit aktivieren Sie den sogenannten Force-Modus. Jeder Punkt wird in der aktuellen Grafik-Farbe gesetzt; der Hintergrund wird nicht berücksichtigt.
Die übrigen drei Modi sind nach den logischen Funktionen benannt, die sie ausführen. OR- und AND-Modus (mit den Parametern drei beziehungsweise vier) sind nur dann interessant, wenn Sie bestimmte Farbänderungen durchführen oder zwei Bilder überlagern wollen. Für die Anwendung in einem Programm ist nur die XOR-Funktion von Bedeutung.
Die Aussagentabelle für diese Funktion finden Sie in Abbildung 2. Die dort angegebenen Werte gelten für MODE 2; in den anderen Modi sind außer Setzen und Rücksetzen eines Punktes auch Farbänderungen möglich.
Die wichtigste Eigenschaft der XOR-Funktion ist jedoch, daß zweimaliges Setzen eines Punktes den Hintergrund wiederherstellt. Damit können Sie zum Beispiel in einem Anwenderprogramm einen Zeiger über den Bildschirm wandern lassen, ohne die Darstellung zu beeinträchtigen. Eine Maussteuerung wird damit denkbar.
Dieser Modus bietet den Vorteil, daß er durch TAG nicht deaktiviert wird. Daher können Sie auch Texte auf dem Grafik-Schirm wieder löschen. Dafür hat CHR$(23) im Textmodus keine Wirkung.
Diese Trennung der beiden wichtigsten Steuercodes auf dem CPC ist zwar bedauerlich, aber nicht zu ändern.
In Listing 2 wird der XOR-Modus nicht abgeschaltet. Wenn Sie das Programm durch Doppel-ESC unterbrechen und erneut starten, erkennen Sie die Auswirkungen, die eine solche Unterlassungssünde nach sich zieht. Da beim Zeichnen der Sinus-Kurve manche Bildpunkte mehrmals gesetzt werden, entstehen Lücken in der Abbildung.

Darstellung von Funktionen

Wenn Sie die Zeile 170 ansehen, fällt Ihnen sicher die umständliche Berechnung für eine solche Grafik auf. Es wäre daher wünschenswert, fertige Unterprogramme zu besitzen, die die Anpassung einer Funktion an den Bildschirm-Bereich übernehmen.
Die dafür nötige Subroutine finden Sie in Listing 3 ab Zeile 330. Die Umrechnungen wirken auf den ersten Blick sehr kompliziert; wenn Sie die Bedeutung der zahlreichen Variablen verstanden haben, erkennen Sie die einfachen mathematischen Grundlagen, die für diese Routine nötig sind.
Mit den Variablen xunten, xoben, yunten und yoben bestimmen Sie den Bereich, in dem die Werte Ihrer Funktion liegen. In Listing 3 soll die Sinus-Funktion von −2*Pi bis +2*Pi gezeichnet werden. Die Werte des Sinus liegen

zwischen −1 und +1; damit die Grafik nicht ganz an die Bildschirm-Ränder reicht, wurde der y-Bereich in jede Richtung um 0,1 erweitert. In den Variablen x und y erwartet das Unterprogramm die Koordinaten des Punktes, der gezeichnet werden soll – und zwar in dem Koordinatensystem, das Sie gerade definiert haben.
Diese sechs Variablen müssen mit sinnvollen Werten belegt sein, damit unsere Subroutine ihre Arbeit tun kann. Bei der Angleichung muß darauf geachtet werden, daß keiner dieser Werte verändert wird. Der Punkt, der tatsächlich auf dem Bildschirm dargestellt wird, wird daher in den Rückgabe-Variablen xs und ys gespeichert.

Im Unterprogramm fallen zunächst die Variablen xadd und yadd auf. Sie bestimmen, wie weit ein Punkt verschoben werden muß, um in den gültigen Wertebereich zu gelangen.
In unserem Fall beginnt das x-Intervall bei −2*Pi, also etwa −6,28. Der darstellbare Bereich beginnt jedoch bei null. Zu jedem Wert, der auf der x-Achse abgebildet werden soll, muß also 6,28 addiert werden. Da die Addition von 6,28 identisch ist mit einer Subtraktion von −6,28, erhält xadd den negativen Wert der untersten Intervallgrenze. Sie können das Verfahren mit einem positiven Startwert leicht nachprüfen.
Würden Sie nur xadd und yadd zu den errechneten Werten addieren, so würde die Sinuskurve im Bereich (0..12,56/0..2,2) dargestellt. Da Sie davon nicht viel erkennen würden, muß zusätzlich eine Streckung in x- und y-Richtung erfolgen.



Mathematik läßt sich auch grafisch darstellen: Hier die Abbildung eines Wertebereichs auf ein festes Koordinaten- System

Parameter machen Programme durchschaubar

Werte, die für die Funktion der Subroutine erforderlich sind, bezeichnet man als Parameter. Ihre Belegung mit Werten im Hauptprogramm und das anschließende GOSUB heißt Übergabe von Parametern an eine Subroutine.
Wenn Sie in BASIC einen sauberen Programmierstil einhalten wollen, sollten Sie zwischen Parametern, internen Variablen und Rückgabewerten streng trennen.

Anpassung des Wertebereichs

Jeder Wert wird dazu mit einem Faktor multipliziert, der sich aus dem Bildschirm- und dem Wertebereich durch Division ergibt. Um dies zu verdeutlichen, sei angenommen, daß sich das x-Intervall von null bis 320 erstreckt. Damit Sie eine Abbildung erhalten, die den gesamten Bildschirm einnimmt, müssen Sie jeden Wert aus diesem Intervall mit zwei multiplizieren. Umgekehrt müßte bei einem Intervall von null bis 1.280 eine Division durch zwei beziehungsweise eine Multiplikation mit 0,5 erfolgen, um die gewünschte Darstellung zu erhalten.
Der Streckungsfaktor ist also nichts anderes als das Verhältnis von Bildschirm- zu Funktions-Intervall. Die Länge des Funktionsbereichs ergibt sich, indem Sie die untere Grenze von der oberen subtrahieren. Zur Sicherheit wird in den Zeilen 360 und 370 noch der Betrag ermittelt, so daß ein eventuell vorhandenes negatives Vorzeichen wegfällt. Zeile 380 und 390 nehmen die endgültige Anpassung mit Hilfe der errechneten Faktoren vor. Danach stehen die benötigten Bildschirm-Koordinaten in den Variablen xs und ys.
Diese Methode ist in Abbildung 3 grafisch dargestellt. Um das Verfahren klar ersichtlich zu machen, wurde ein Bildschirm-Bereich von acht Bildpunkten angenommen. Die Funktion soll Werte von −2 bis 2 annehmen können.
Die Grafik zeigt gleichzeitig, daß eine Streckung um einen positiven Faktor eine Vergröberung der Darstellung zur Folge hat. In der Praxis läßt sich dieses Problem jedoch umgehen, da fast jede Funktion beliebig viele reelle Werte liefert. Sie müssen lediglich die Schrittweite für deren Berechnung hinreichend klein wählen. Schwierigkeiten treten nur bei der Darstellung eines sehr engen Funktionsausschnitts auf, bei dem die Rechengenauigkeit des CPC nicht mehr ausreicht.

Grafikfenster: Es geht auch ohne ORIGIN

Die Routine in Listing 3 stellt eine Funktion nur im vollständigen

Bildbereich dar. Es wäre von Vorteil, wenn Sie zusätzlich den Bildausschnitt frei wählen könnten. Diese Möglichkeit bietet Ihnen Listing 4.
Zusätzlich zu den in Abbildung 4 beschriebenen Variablen existieren hier bildxunten, bildxoben, bildyunten und bildyoben, mit denen Sie den gewünschten Ausschnitt festlegen. Als interne Variablen wurden xbreite und xhoehe eingeführt. Sie geben die Größe des Bildschirm-Intervalls an, die im Gegensatz zu Listing 3 nicht mehr fest mit 640 beziehungsweise 400 angegeben werden kann.
Mit diesen Erläuterungen fällt es Ihnen bestimmt nicht schwer, das Programm zu verstehen. Es ist empfehlenswert, das Unterprogramm ab Zeile 380 getrennt abzutippen, mit brauchbaren Zeilennummern – etwa ab 50000 – zu versehen und abzuspeichern. Damit können Sie es mit dem MERGE-Befehl in Ihre eigenen Programme einbinden, wenn Sie Funktionsdarstellungen benötigen.

Parameter	
xunten	untere Grenze (x-Werte) des Anwenderbereichs
xoben	obere Grenze (x-Werte) des Anwenderbereichs
yunten	untere Grenze (y-Werte) des Anwenderbereichs
yoben	obere Grenze (y-Werte) des Anwenderbereichs
x	darzustellender Punkt; Anwender-Wert x
y	darzustellender Punkt; Anwender-Wert y
Interne Variable	
xadd	Verschiebungs-Konstante x-Achse
yadd	Verschiebungs-Konstante y-Achse
xfaktor	Streckungs-Faktor x-Achse
yfaktor	Streckungs-Faktor y-Achse
Rückgabe-Werte	
xs	darzustellender Punkt, x-Wert
ys	darzustellender Punkt, y-Wert

Die Ein-/Ausgabe-Schnittstellen unseres Unterprogramms in tabellarischer Form

Der CPC – ein Grafik-Computer

Der CPC besitzt eine Auflösung, wie sie etwa IBM-kompatible Geräte bieten. Wenn Sie die vielfältigen Möglichkeiten des Schneider-Computers richtig einzusetzen wissen, können Sie Grafik-Programme schreiben, die in ihrer Leistung mit Software für professionelle Rechner durchaus konkurrieren können.
Wenn Sie im Laufe dieser Serie nicht nur mit den Grafikbefehlen des CPC, sondern auch mit allgemeinen programmtechnischen Hinweisen konfrontiert wurden, so liegt das in der Natur der Sache. Ein derart komplexes Thema kann nicht mit einer Beschreibung der einzelnen Befehle abgehandelt werden. Dazu würde ein Blick ins BASIC-Handbuch genügen. Vielmehr sollten hier einige – beileibe nicht alle – mögliche Methoden effektiver Programmierung aufgezeigt werden. Wie und wo Sie diese Verfahren einsetzen, bleibt Ihnen überlassen.

AE □

```
100 'Haus                                  (0B04)
110 '                                      (07E7)
120 'Farben festlegen                      (17FE)
130 INK 0,0:INK 1,26:INK 2,6:INK 3
,11:PAPER 0:BORDER 0:MODE 0                (2358)
140 'Zeichen definieren                    (19BD)
150 SYMBOL 252,&18,&3C,&7E,&FF,0,0
,0,0                                       (20F8)
160 SYMBOL 253,0,0,0,0,0,&66,0,0           (1AD6)
170 SYMBOL 254,0,0,0,0,&FF,&FF,&FF
,&FF                                       (2032)
180 'Zeichen ueberlagern                   (1AE8)
190 PRINT CHR$(22);CHR$(1);'Transp
arent ein                                  (243E)
200 LOCATE 10,10                           (0B9D)
210 PEN 1:PRINT CHR$(254);CHR$(8);
                                           (164F)
220 PEN 2:PRINT CHR$(252);CHR$(8);
                                           (1665)
230 PEN 3:PRINT CHR$(253)                  (0F13)
240 PRINT CHR$(22);CHR$(0);'Transp
arent aus                                  (2476)
250 PRINT:PRINT                            (08F6)
```

```
100 'XOR-Demo                          <0F27>
110 '                                  <07E7>
120 'Farben festlegen                  <17FE>
130 INK 0,0:INK 1,6:INK 2,21:PAPER
 0:BORDER 0:MODE 0                     <1DDA>
140 'Bild zeichnen                     <14E6>
150 MOVE 0,200                         <0A35>
160 FOR i=0 TO 2*PI STEP 0.02          <18A2>
170 DRAW i*(640/(2*PI)),200+SIN(i)
*190,1                                 <2804>
180 NEXT i                             <0AD8>
190 'Linie darueber                    <15C7>
200 PRINT CHR$(23);CHR$(1);'XOR an
                                       <1BAE>
210 FOR i=400 TO 0 STEP-2              <1380>
220 MOVE 0,i:DRAWR 640,0,2             <15D2>
230 FOR j=1 TO 10:NEXT j               <155C>
240 DRAWR-640,0,2                      <0E62>
250 NEXT i                             <0A65>
260 GOTO 210                           <098D>
```

```
100 'Sinus                             <0C6A>
110 '                                  <07E7>
120 'Farben bestimmem                  <17AB>
130 INK 0,0:PAPER 0:INK 1,15:BORDE
R 0:MODE 1                             <18C0>
140 'Wertebereich setzen               <1AD1>
150 xunten=-2*PI                       <14A5>
160 xoben=2*PI                         <1245>
170 yunten=-1.1                        <1679>
180 yoben=1.1                          <146B>
190 'Startpunkt setzen                 <1896>
200 x=-2*PI                            <0F0F>
210 y=SIN(x)                           <1239>
220 GOSUB 330                          <09DD>
230 PLOT xs,ys,1                       <133E>
240 'Kurve zeichnen                    <1536>
250 FOR x=-2*PI TO 2*PI STEP 0.02      <1CC5>
260 y=SIN(x)                           <129E>
270 GOSUB 330                          <0941>
280 DRAW xs,ys                         <115F>
290 NEXT x                             <0AF2>
300 'Warten auf Taste                  <17D2>
310 WHILE INKEY$="":WEND               <0D2B>
320 END                                <06E3>
330 'Koordinaten angleichen            <1D43>
340 xadd=-xunten                       <17E3>
350 yadd=-yunten                       <176F>
360 xfaktor=ABS(640/(xoben-xunten)
)                                      <2C84>
370 yfaktor=ABS(400/(yoben-yunten)
)                                      <2CBF>
380 xs=(x+xadd)*xfaktor                <2402>
390 ys=(y+yadd)*yfaktor                <24F5>
400 RETURN                             <0647>
```

```
100 'Sinus                             <0C6A>
110 '                                  <07E7>
120 'Farben bestimmem                  <17AB>
130 INK 0,0:PAPER 0:INK 1,15:BORDE
R 0:MODE 1                             <18C0>
140 'Wertebereich setzen               <1AD1>
150 xunten=-2*PI                       <14A5>
160 xoben=2*PI                         <1245>
170 yunten=-1.1                        <1679>
180 yoben=1.1                          <146B>
190 'Bildbereich setzen                <199D>
200 bildxunten=0                       <141D>
210 bildxoben=320                      <15BB>
220 bildyunten=200                     <152E>
230 bildyoben=399                      <1548>
240 'Startpunkt setzen                 <18FA>
250 x=-2*PI                            <0F73>
260 y=SIN(x)                           <129E>
270 GOSUB 380                          <09D2>
280 PLOT xs,ys,1                       <13A3>
290 'Kurve zeichnen                    <159B>
300 FOR x=-2*PI TO 2*PI STEP 0.02      <1C29>
310 y=SIN(x)                           <1201>
320 GOSUB 380                          <0935>
330 DRAW xs,ys                         <11C3>
340 NEXT x                             <0A55>
350 'Warten auf Taste                  <1735>
360 WHILE INKEY$="":WEND               <0D8F>
370 END                                <0648>
380 'Koordinaten angleichen            <1DA7>
390 xadd=-xunten                       <1748>
400 yadd=-yunten                       <17D3>
410 xbreite=bildxoben-bildxunten       <2AE5>
420 yhoehe=bildyoben-bildyunten        <2969>
430 xfaktor=ABS(xbreite/(xoben-xun
ten))                                  <3370>
440 yfaktor=ABS(yhoehe/(yoben-yunt
en))                                   <32BD>
450 xs=(x+xadd)*xfaktor+bildxunten
                                       <327F>
460 ys=(y+yadd)*yfaktor+bildyunten
                                       <321C>
470 RETURN                             <06D3>
```

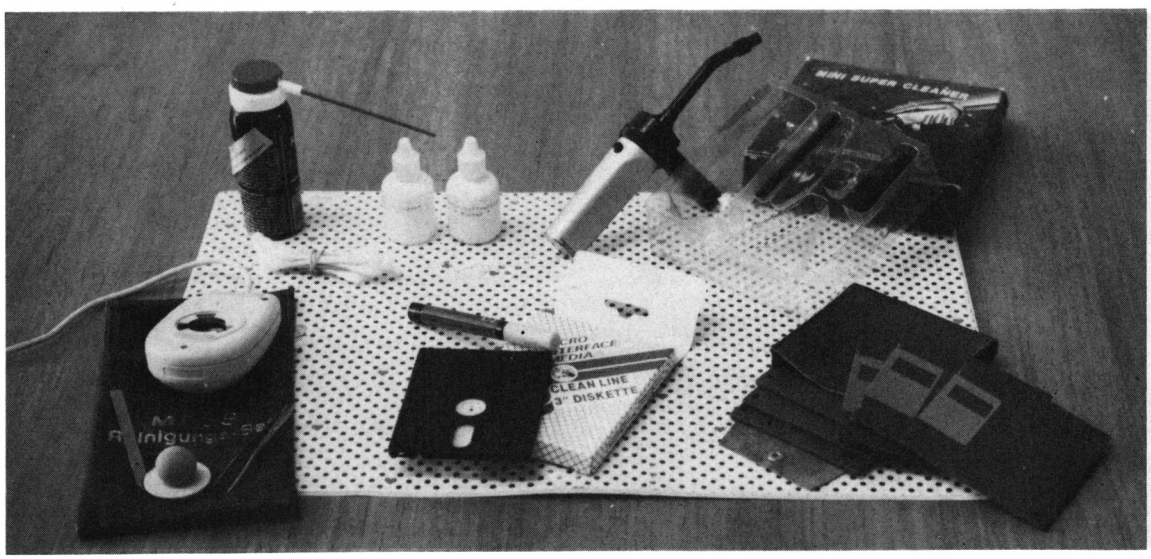


Zubehör für den Computer

Nippes

Kleinkram rund um den Rechner

Kleine Zubehör-Teile machen die Arbeit oft erst effizient. Doch wenn der Schreibtisch irgendwann überquillt vor lauter „nützlichem" Kleinkram – was dann?



Fabrikanten von Auto-Zubehör haben den Trend der Zeit längst erkannt. Kleinigkeiten machen ein Fahrzeug erst liebenswert. Das fängt an beim selbstgehäkelten Hütchen für die Klopapier-Rolle und hört auf bei sportlichen Accessoires wie Alu-Felgen und Spoilern.

Nun käme zwar niemand auf die Idee, seinem Computer ein Rallye-Outfit zu verleihen, indem er ihm eine Flosse zur Verringerung des CW-Wertes ans Gehäuse montiert. Doch die Nippes-Hersteller fanden auch in diesem Bereich Marktlücken.

Der Rechner ist nämlich kein reines Arbeitstier mehr. Insofern ist der Vergleich mit dem Auto berechtigt. Ebenso, wie sich das eigene Fahrzeug vom Beförderungsmittel zum Hobby oder sogar Lebensinhalt gewandelt hat, stehen heute in vielen Wohnzimmern Computer, die ihren Besitzern nicht nur das Leben leichter machen sollen. Vielmehr wird ein großer Teil der Freizeit dem eigenen Rechner gewidmet.

Was also liegt näher, als seinem geliebten Homecomputer oder PC einmal etwas Gutes zu tun? Nehmen wir an, der stolze Computer-Besitzer fährt in Urlaub und läßt seinen Rechner für zwei Wochen unbeaufsichtigt in der Heimat.

Nach der Rückkehr wird er entsetzt feststellen, daß ein Staubflöckchen die Unverschämtheit besessen hat, sich auf

seinem Computer niederzulassen. Der Hobbyist schwört sich, seinen Rechner entweder nie wieder alleinzulassen oder ihn vor den Gewalten der Natur zu schützen.

Doch ein so sensibles Wesen wie sein geliebter Computer macht ihm dieses Unterfangen nicht allzu leicht. Während das nicht minder geliebte Auto in einer Garage relativ sicher ist, stellt selbst ein solch massives Gebäude für den Compi keinen Schutz dar. Auch dort gibt es schließlich Staub.

Ein Aufschrei tönte also aus den Kehlen der PC-Besitzer: „Wer schützt unsere armen, hilflosen Computer?" Dieser Schreckensruf drohte schon ungehört zu verhallen, als sich einige mutige Unternehmer ein Herz faßten. Sie produzierten — endlich! — Staubschutzhaben für des Deutschen zweitliebstes Kind.

Vorbei waren die Zeiten der selbstgehäkelten oder -gestrickten Notbehelfe. Endlich gab es ein Mittel, das dem heimischen PC seinen alten Glanz bewahrte.

Zum Gehäuse des Rechners mußte die Staubschutzhabe freilich passen. Schließlich gibt keiner seinen Liebling der Lächerlichkeit preis. Wie sähen denn auch 135er Reifen an einem Ferrari aus? Genauso lächerlich wie eine Stoffhaube auf einem AT.

Nein, für solche Luxus-Modelle muß der Staubschutz mindestens aus gut verarbeitetem Kunststoff sein. Durchsichtig, versteht sich, damit jedermann die erstklassige Tastatur erkennen und bewundern kann. Ganz nobel wirkt Rauchglas, besonders auf einem 386er. Für den C64 hingegen genügt eine Plastikfolie.

Damit schienen die Computer-Besitzer glänzend dazustehen. Aber ach — der Mensch ist ja so vergeßlich! Wer denkt schon immer daran, die seinem Rechner gebührenden Schutzmaßnahmen zu treffen? Besonders Raucher haben ein schweres Leben. Zu der allgemeinen Diskriminierung durch ihre nichtrauchenden Mitmenschen kommt auch noch die Verschmutzung der Tastatur ihres Lieblings durch verlorene Asche.

Wieder schien es so, als wäre ein Punkt erreicht, an dem eine tragische Entscheidung ansteht. Sollte man auf den geliebten Computer oder auf die nicht minder geliebte Zigarette verzichten? Doch wieder nahte im letzten Moment Hilfe.

Der Tastatur-Staubsauger war die absolute Lösung aller Probleme. Endlich war es möglich, Rauchen und Rechnen miteinander zu verbinden. Was machte es schon, wenn ab und zu etwas Asche zwischen die Tasten geriet? Staubsauger an, und ruckzuck erstrahlte der Computer in neuem Glanz.

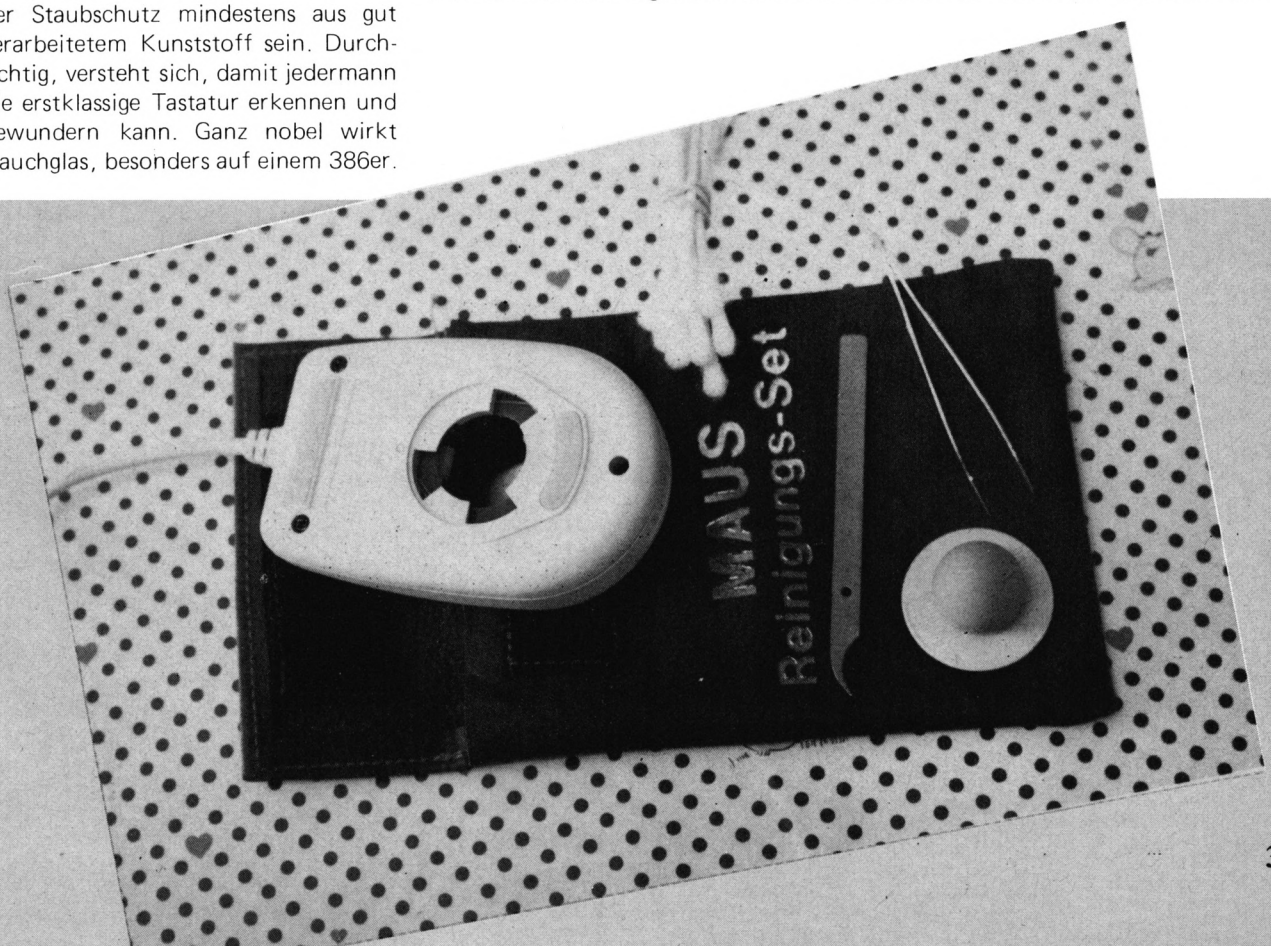
Sie mögen sich jetzt fragen, wo denn des Deutschen liebstes Hobby — sein Auto — geblieben sei. In der Tat gab es hier gewisse Probleme. Computer waren für den mobilen Einsatz lange Zeit zu unhandlich. Mit einem Portable, den Sie an den Zigaretten-Anzünder anschließen können (Kabel wird mitgeliefert), hat die Firma Amstrad aber auch dieses Problem gelöst.

Dabei bleiben lediglich zwei Fragen offen. Womit sollen Sie sich eine Zigarette anzünden, wenn der Anzünder-Stecker belegt ist? Ferner ist noch ungeklärt, ob die Multitasking-Fähigkeiten eines biologischen Rechners des Typs Homo Sapiens nicht überstrapaziert werden, wenn er bei Tempo 200 auch noch einen Computer bedienen soll.

Doch auch diese Probleme, soviel ist sicher, werden in naher oder ferner Zukunft gelöst werden. Fürs erste werden wir uns wohl mit dem heimischen Schreibtisch begnügen müssen. Das genügt auch vollkommen, denn hier stellen sich dem Anwender noch genügend Hindernisse in den Weg, die ihm die Freude an seinem Hobby vergällen können.

Der Schmutz ist des Deutschen größter Feind. Läßt sich eine Tastatur noch mühelos mit dem passenden Staubsauger sauberhalten, so schienen die Anwender mit der Einführung eines neuen Peripherie-Geräts im Kampf gegen Dreck endgültig zu unterliegen. Die Rede ist — natürlich — von dem Ding mit der Rollkugel und den zwei Knöpfen, kurz: von der Maus.

Irgendwann weigert sich jede Maus, weiter zu rollen. Das Tierchen ist ein-



fach zu verschmutzt; und schlecht gepflegte Mäuse – das weiß jeder Hobby-Züchter – arbeiten nicht gerne.
Dieser Arbeitsunlust konnte nur mit zwei Methoden abgeholfen werden. Die erste war, den Schreibtisch peinlichst sauber zu halten. Aus den oben angeführten Gründen (Asche etcetera) war dies kaum möglich. Methode Nummer zwei bestand darin, die Maus gelegentlich zu säubern.
Während dies bei lebenden Mäusen durch ein lauwarmes Bad problemlos möglich ist, stößt man bei Computer-Mäusen mit diesem Verfahren auf wenig Gegenliebe. Elektronik und Wasser sind nun einmal natürliche Feinde.
Wieder – nun schon zum dritten Mal – war ein Punkt erreicht, an dem die Anwender weder aus noch ein wußten. Und wiederum, kurz bevor sie vor Ärger den ganzen Krempel aus dem Fenster warfen, kam die Rettung.
Das Maus-Reinigungs-Set erwies sich als praktisches Hilfsmittel zur Pflege dieser niedlichen Tierchen. Mit Werkzeugen, die einem Zahnarzt oder Chirurgen zur Ehre gereichen würden, können Sie nun Ihre Maus operieren, auch wenn Sie keinen Professoren-Titel haben oder nicht Brinkmann heißen.
Doch leider ist noch kein Meister vom Himmel gefallen. Wundern Sie sich also nicht, wenn Sie hinterher mit einem Schulterzucken nur äußern können: „Operation gelungen – Patient tot".
Derartige Arbeiten überlassen Anwender, die ihren Rechner lieben, ohnehin dem Fachmann. Es käme ja wohl auch niemand auf die Idee, um unseren Vergleich wiederaufzugreifen, selbst dilettantisch an seinem Auto herumzupfuschen. Und ganz besonders nicht, wenn ein Stern auf der Kühlerhaube prangt.
Das Interesse des stolzen Besitzers gilt vielmehr dem ausgefallenen Zubehör. Mit den richtigen Accessoires macht man ja erst deutlich, daß man zum Kreis der Eingeweihten, zu den „richtigen" Anwendern gehört.
Eine Staubschutzhaube macht sich da recht gut. Sie ist zwar schon ein alter Hut, doch in der richtigen Verarbeitung kann sie das Prunkstück des heimischen Büros darstellen.
Ein solches Gerät hat nur einen kleinen Nachteil. Man kann es nicht stets mit sich herumtragen, um überall zu beweisen, wie „in" man doch ist. Ach, wie schön war es doch, als wir noch unsere Autos als Hobbies hatten! Einen Zündschlüssel mit einem möglichst auffälligen Marken-Symbol konnte man immer „unauffällig" herumliegen lassen.
Das Statussymbol des Computer-Besitzers ist die Diskette 5,25 Zoll sind out, 3,5 Zoll sind in. Diese Größe bietet auf dem Schreibtisch nicht etwa besondere Vorteile, wie man annehmen könnte. Eine solche Disk läßt sich lediglich bequem in jeder Jackentasche unterbringen – und zu gegebener Zeit, auffällig unauffällig, vorzeigen.
Nur läßt sich – um wieder einen Vergleich zu wählen – mit einem Geldschein allein kein Eindruck schinden. Eine noble Brieftasche aus echtem Leder, mit Monogramm und natürlich Markenware, lässig gezückt und auf den Tisch gelegt, kann dagegen schon beeindrucken.
Daraus folgern wir, daß auch Disketten richtig verpackt sein müssen, wenn sie ihre Wirkung auf den Gesprächspartner nicht verfehlen sollen. Ein Mäppchen aus Kunstleder (merkt doch keiner), farblich abgestimmt auf die blauen Disketten, muß schon sein. Einzig das Monogramm fehlt; aber das können Sie auch nachträglich einprägen lassen.
Zur stilechten Ausstattung eines echten Comuter-Anwenders gehört natürlich auch die Maus-Unterlage auf dem Schreibtisch: Echtes Leder und hervorragend gepolstert. Was, Sie haben noch keine? Also, mal ehrlich – so ganz up to date sind Sie wohl nicht mehr, was?
Gut zur Schau gestellt auf dem Wohnzimmer-Schreibtisch gehört natürlcih auch die Reinigungs-Diskette. Man sieht es ihr an, daß sie unbenutzt ist. Welcher Computer-Anwender käme denn auch auf die Idee, sich von so einem Ding sein Laufwerk ruinieren zu lassen? Doch ein echter User muß einfach demonstrieren, daß er dem Schmutz auch in der hintersten Ecke seines Computers keine Chance läßt.
Der Computer, zu dem all dieses Zubehör gehört, muß natürlich auch von entsprechendem Format sein. Mit einem ZX-81 läßt sich kein Staat machen; ein IBM AT muß es schon sein. Original, versteht sich, kein billiger Taiwan-Nachbau. Das IBM-Logo auf dem Gehäuse gilt mindestens ebensoviel wie der angesprochene Stern auf der Kühlerhaube.
Und wieder sind Parallelen zu entdecken. Edle Autos sind genauso wenig zum Fahren da wie edle Computer zum Arbeiten. Nur gereinigt werden beide öfter als ihnen gut tut.
Die stolzen Eigentümer solcher Geräte werden aber niemals sagen, „Ich **habe** einen AT". Es genügt nicht, einen Computer nur zu besitzen. Man muß vielmehr zeigen, daß man ihn auch beherrscht: „Ich **arbeite** mit einem AT" klingt doch schon viel besser. Kommt Ihnen dieses Verhalten bekannt vor? Richtig – siehe oben!
Der Nippes-Industrie sind Besitzerstolz und Geltungsbedürfnis nur allzu recht. Welcher Lötkolben-Freak etwa würde den ganzen Kram schon kaufen? Nein, bei einem Arbeits-Computer liegt der ganze Kram doch nur im Weg.
Die Prestige-Rechner, die Anschau-Computer und deren Eigentümer sind es, auf die man ein Auge geworfen hat. Und wir wären keine Deutschen, wenn wir diesen Firmen keinen Absatzmarkt bieten würden. ● *AE*



Joystick-Abfrage

BIT FÜR BIT

Binäre Operationen verkürzen Ihre Programme

Bei der Abfrage des Joysticks entwickeln manche BASIC-Programmierer viel Fantasie. In manchen Listings werden bis zu zehn Programmzeilen für diese einfache Aufgabe verwendet. Mit ein wenig Mühe geht es aber auch einfacher.

Die Joystick-Abfrage, die wir Ihnen in diesem Beitrag vorstellen, benötigt nur eine einzige Zeile. An die Subroutine werden die aktuellen Grafik-Koordinaten in den Variablen **x** und **y** übergeben. Die Werte in **xschritt** und **yschritt** geben zusätzlich an, wie weit in die jeweilige Richtung gesprungen werden soll.
Das Unterprogramm liefert die anhand der Joystick-Auslenkung berichtigten x- und y-Werte. Darüber hinaus wird das Flag **aktion** gesetzt. Es hat den Wert −1, wenn die Feuertaste betätigt wurde.

Einfache Hardware: der Joystick

Um eine kurze Joystick-Abfrage zu programmieren, müssen Sie sich zunächst mit dem Funktionsprinzip eines solchen Steuerknüppels vertraut machen. Der neunpolige Anschluß des CPC besitzt fünf Eingänge, die für Joystick null vorgesehen sind.
Ein sechster Anschluß liefert eine Spannung von +5 Volt. Im Joystick befinden sich in der Regel fünf Mikro-Schalter: vier für die Richtung oben, unten, links und rechts sowie ein weiterer für die Feuertaste.
Wird der Joystick ausgelenkt oder die Feuertaste betätigt, so schließen die entsprechenden Schalter. An den zugehörigen Eingängen liegt dadurch Spannung an.



Der CPC überprüft den Jostick-Eingang in regelmäßigen Abständen. Dies geschieht zusammen mit der Tastatur, die ja auch immer wieder abgefragt werden muß, um auf eine Eingabe reagieren zu können. Die Spannungen, die am Eingang anliegen, werden im Speicher als ein Byte dargestellt.
Dieses Byte können Sie in BASIC aus dem Speicher lesen, wenn Sie die Funktion

```
JOY(0)
```

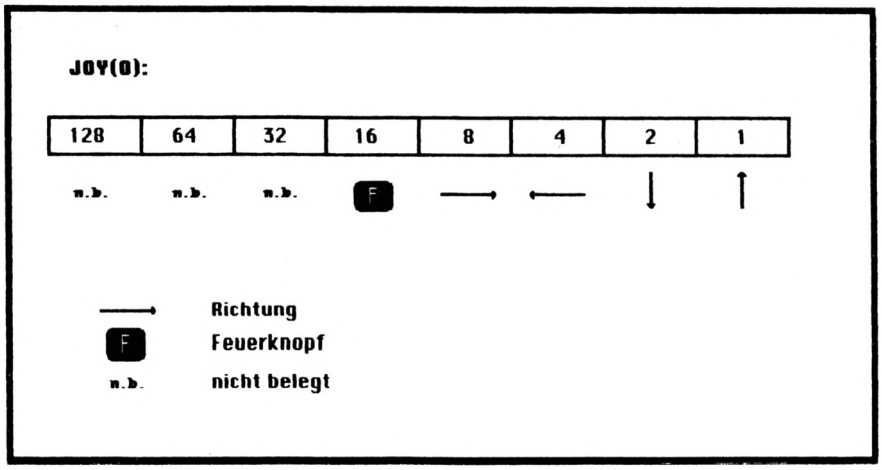
anwenden. Jedes gesetzte Bit stellt eine Richtung dar, wie Sie Abbildung 1 entnehmen können. Da zwar acht Bits zur Verfügung stehen, aber nur fünf Leitungen benötigt werden, sind die oberen drei nicht belegt. Das sechste Bit wird für einen eventuell vorhandenen zweiten Feuerknopf benötigt; diese Möglichkeit soll hier jedoch nicht berücksichtigt werden.

Unzählige Möglichkeiten

Mit dem Joystick können Sie in insgesamt acht Richtungen lenken. In Kombination mit der Feuertaste ergeben sich sechzehn verschiedene Möglichkeiten. Außerdem könnte es sein, daß nur die Feuertaste betätigt oder der Steuerknüppel überhaupt nicht ausgelenkt wird.
Um jede dieser Optionen einzeln abzufragen, müßten Sie 18 verschiedene Werte überprüfen. Dies ist selbst dann der Fall, wenn Sie nur lenken und schießen wollen: Gleichzeitig mit dem Druck auf die Feuertaste können Sie ja in eine beliebige Richtung steuern. Es muß also eine Möglichkeit gefunden werden, zu überprüfen, ob ein einzelnes Bit gesetzt ist. Die Funktion

```
AND
```

ist des Rätsels Lösung.
Abbildung zwei zeigt die Wahrheitstabelle der AND-Funktion.



Die Belegung der einzelnen Bit bei der JOY-Funktion

Logische Funktionen haben die Eigenschaft, zwei Zahlen bitweise zu verknüpfen. AND läßt dabei im Ergebnis nur an den Stellen ein Bit stehen, an denen in beiden Operanden eines gesetzt war.
Um herauszufinden, ob der Joystick in eine bestimmte Richtung gelenkt wird, müssen Sie also den Dezimalwert für diese Richtung mit dem durch JOY(0) gelieferten Wert via AND verknüpfen.
Abbildung 3 zeigt, wie eine Überprüfung für die Auslenkung nach rechts ausfällt. Das vierte Bit muß gesetzt sein. Im Dezimal-System entspricht dies der Zahl Acht. Die Operation

```
8 AND 8
```

liefert die Acht als Ergebnis.
Was aber geschieht, wenn Sie nach rechts unten steuern? In diesem Fall schließen zwei Kontakte gleichzeitig, so daß zwei Bits gesetzt sind.
Die AND-Verknüpfung mit der Zahl Acht ergibt jedoch wieder die Acht. Der Wert zwei, der für die Auslenkung nach oben addiert wird, fällt weg: Das entsprechende Bit ist im zweiten Operanden nicht gesetzt.

AND verkürzt die Abfrage

Um eine Joystick-Abfrage nach den eingangs gemachten Vorgaben zu realisieren, benötigen Sie nach dieser Methode fünf Programmzeilen. Geben wir ihnen die Nummern 1 bis 5, so ergibt sich folgende Routine:

```
1 IF JOY(0) AND 1 THEN
  y=y-yschritt
2 IF JOY(0) AND 2 THEN
  y=y+yschritt
3 IF JOY(0) AND 4 THEN
  x=x+xschritt
4 IF JOY(0) AND 8 THEN
  x=x-xschritt
5 IF JOY(0) AND 16 THEN
  aktion=-1 ELSE aktion=0
```

Beachten Sie, daß jede dieser Zeilen abgearbeitet werden **muß**. Damit werden auch diagonale Auslenkungen des Joysticks erfaßt.
Die Bedingung

```
IF JOY(0) AND 1
```

bedeutet dasselbe wie

```
IF (JOY(0) AND 1) <> 0
```

und verkürzt dadurch den Programmtext. Die AND-Funktion liefert in diesem Fall den Wert eins, wenn der Joystick nach unten gelenkt wird; in jedem anderen Fall erhalten Sie eine Null.
Anstelle von

```
IF (JOY(0) AND 1) = 1
```

können Sie daher die Kurzform verwenden.
Es ist der Mühe wert, die fünf Programm-Zeilen mit allen möglichen

Wahrheitstabelle der AND-Funktion

Bit A	Bit B	Ergebnis
0	0	0
0	1	0
1	0	0
1	1	1

AND erleichtert die Joystick-Abfrage

Werten auf einem Blatt Papier durchzurechnen, und zwar in binärer Darstellung. Mit Hilfe von Abbildung 1 dürfte Ihnen das leicht gelingen.

Wahr oder falsch?

Die Eigenschaft des CPC-BASIC, in IF-Bedingungen keinen Operator „<>0" zu verlangen, gibt Ihnen erste Hinweise darauf, wie solche arithmetischen Ausdrücke in Ihrem Rechner ausgewertet werden. Geben Sie folgende Zeile ein:

```
PRINT 1=1
```

Für Mathematiker scheint dies keinen Sinn zu geben. Ihr CPC liefert dennoch ein Ergebnis. Auf dem Bildschirm erscheint

```
-1
```

Versuchen Sie es auch mit dem Befehl

```
PRINT 1=2
```

Der CPC liefert das Ergebnis

```
0
```

Wenn Sie Ihrem Rechner eine vollständige Gleichung oder Ungleichung als Bestandteil eines arithmetischen Ausdrucks angeben, so wird diese Aussage auf ihre Korrektheit hin überprüft. Die Aussage „1=1" ist wahr. Dies wird durch das Ergebnis −1 angegeben.
Die Aussage „1=2" hingegen ist falsch. In diesem Fall liefert der CPC eine Null.

Ein wenig Mathematik

Dieses Verhalten können Sie sich zunutze machen. Betrachten wir, was geschieht, wenn Ihr Computer die Aussage

```
(JOY(0) AND 8) <> 0
```

berechnet.
Die Acht ist das vierte Bit, das vom Joystick-Eingang geliefert wird. Es symbolisiert eine Auslenkung nach rechts.
Steuern Sie tatsächlich nach rechts, so liefert die AND-Verknüpfung mit dem Wert acht das Resultat acht. Die Aussage „acht ist ungleich null" ist wahr. Der CPC ermittelt −1 als Wahrheits-Wert der Ungleichung.
Lenken Sie nicht nach rechts, erhalten Sie durch die AND-Funktion das Ergebnis null. Im Klartext machen Sie also die Aussage „null ist ungleich null". Offensichtlich ist dies falsch: Der Computer gibt eine Null aus.
Nach den eingangs gemachten Vorgaben soll der vorgegebene x-Wert anhand der Joystick-Auslenkung berichtigt werden. Wird nach rechts gelenkt, so liefert die Ungleichung den Wert −1. Subtrahieren Sie diese Zahl vom aktuellen x-Wert, so wird eins addiert:

```
-(-1) = +1
```

Damit zusätzlich die Schrittweite berücksichtigt wird, müssen Sie diesen Wert noch mit **xschritt** multiplizieren. Wir erhalten die Formel:

```
x=x-xschritt*((JOY(0) AND 8)
  <> 0)
```

Für eine Bewegung nach links müssen Sie dieselben Überlegungen mit der Zahl Vier durchführen. Da diese Richtung einen kleineren x-Wert bedingt, muß jeweils der Wert von **xschritt** subtrahiert werden. Aus dem Minus-Zeichen unserer Formel wird daher ein Plus:

```
x=x+schritt*((JOY(0) AND 4
  <> 0)
```

Da eine Bewegung nach links eine gleichzeitige Auslenkung des Joysticks nach rechts ausschließt, können Sie die beiden Formeln auch zusammenfassen:

```
x=x-xschritt*((JOY(0) AND 8)
  <> 0)
   +xschritt*((JOY(0) AND 4)
  <> 0)
```

Der Faktor **xschritt** ist jetzt zweimal vorhanden. Durch Klammerung können Sie den Ausdruck noch verkürzen. Beachten Sie, daß vor der Klammer ein Minus steht. Innerhalb der Klammer muß also ein Vorzeichen-Wechsel stattfinden:

```
x=x-xschritt*(((JOY(0) AND
  <> 0) -((JOY(0) AND 4)
  <> 0))
```

Der Rest ist einfach

Für die Bewegung in vertikaler Richtung können Sie dieselben Berechnungen durchführen. Die

JOY(0):	00001000 = 8
Vergleich:	00001000 = 8
AND:	00001000 = 8
JOY(0):	00001010 = 16
Vergleich:	00001000 = 8
AND:	00001000 = 8

Zwei Beispiele für die Wirkung der AND-Funktion

endgültige Formel zur Berechnung des y-Wertes lautet:

```
y=y-yschritt*(((JOY(0) AND 1)
  <> 0) -((JOY(0) AND 2)
  <> 0))
```

Das Flag **aktion** soll, je nachdem, ob der Feuerknopf betätigt wurde oder nicht, den Wert –1 oder null annehmen. Eine einfache logische Auswertung bringt das gewünschte Ergebnis:

```
aktion=((JOY(0) AND 16)<> 0)
```

Mit diesen drei Berechnungen verfügen Sie über ein komplettes Unterprogramm zur Steuerung einer Bewegung auf dem Bildschirm. Unser Listing zeigt ein Beispiel. Für den Einsatz in eigenen Programmen können Sie alle Zeilen unter 10000 löschen und die Subroutine mit

```
SAVE"JOY.MRG",A
```

abspeichern. Der Befehl

```
MERGE"JOY.MRG"
```

bindet das Unterprogramm in Ihr Listing ein. Die Kommentar-Zeilen im Kopf der Routine sollen Sie an die benötigten Variablen erinnern. *AE*□

```
100 'Demo-Hauptprogramm                 <1923>
110 'zur Joystick-Abfrage               <1B2D>
120 INK 0,0:PAPER 0:INK 1,6:PEN 1:
BORDER 0:MODE 2                         <1A53>
130 PRINT CHR$(23);CHR$(1);:TAG         <157C>
140 x=320:y=200:xschritt=1:yschrit
t=2                                     <3181>
150 MOVE x,y:PRINT"Test";               <1813>
160 SOUND 1,y,20,7:MOVE x,y:PRINT"
Test";                                  <25A1>
170 GOSUB 10000                         <0946>
180 IF NOT aktion THEN 150              <14F0>
190 TAGOFF:PRINT CHR$(23);CHR$(0);
:CLS:END                                <19D7>
10000 'Unterprogramm Joystick-Abfr
age                                     <251B>
10010 'Uebergabe:                       <115B>
10020 'x = aktuelle x-Position          <1E80>
10030 'y = aktuelle y-Position          <1EA8>
10040 'xschritt = Schrittweite x        <2096>
10050 'yschritt = Schrittweite y        <201A>
10060 'Rueckgabe:                       <111F>
10070 'x = neue x-Position              <1AE1>
10080 'y = neue y-Position              <1A00>
10090 'aktion = Flag fuer Feuertas
te,                                     <25B4>
10100 '          -1:gedrueckt,0:nic
ht                                      <247E>
10110 x=x-xschritt*(((JOY(0)AND 8)
<>0)-((JOY(0)AND 4)<>0)):y=y-yschr
itt*(((JOY(0)AND 1)<>0)-((JOY(0)AN
D 2)<>0)):aktion=((JOY(0)AND 16)<>
0):RETURN                               <8719>
```



RSX Extended BASIC

Das Programm RSX Extended BASIC ist eine äußerst leistungsstarke Befehlserweiterung für den Schneider CPC 464. Dem User stehen 58(!) neue Befehle zur Verfügung, die als RSX in das Betriebssystem eingebunden wurden.

Eine Befehlserweiterung für den CPC 464 hat unser Autor Andre Schröder aus Hannover geschrieben.
Die 58 neuen Befehle umfassen die Gebiete Grafik, Schriftvergrößerung und Manipulation, Disketten- und Kassettenroutinen, Bildschirm- und Speicheroperationen sowie mehrere Toolkits. Extended BASIC läuft sowohl mit Kassette als auch mit Diskette. Für eine sinnvolle Anwendung empfiehlt sich der gleichzeitige Betrieb beider Speichermedien.
Tippen Sie zunächst den BASIC-Lader ein und speichern Sie ihn vorsichtshalber ab. Tippfehler und der damit verbundene Absturz des Computers werden dadurch nicht zur Katastrophe. Danach starten Sie den BASIC-Lader. Es wird nun automatisch die Binärdatei EXTEN.BIN erzeugt und abgespeichert. Geladen wird die MC-Datei künftig durch das kurze BASIC-Programm EXTENDED.BAS, das die Kassettenbesitzer vor das Binärfile setzen sollten.
Allen Befehlen muß der RSX-Strich vorangestellt werden. Sie finden dieses Zeichen rechts neben P auf der Tastatur und erreichen es durch SHIFT und Klammeraffe. Folgende Befehle stehen Ihnen zur Verfügung:

1. Toolkits:

GETKEY,string
Wartet auf Taste und lädt diese in den angegebenen Strings (zum Beispiel !GETKEY,*a$).
* steht für den Klammeraffen.

BREAK.ON/.OFF
Bestimmt den Zustand der ESC-Taste.

ERROR.ON/.OFF
Ein- und Ausschalten der Fehlerabfangroutine. Nützlich zum Beispiel bei Memory full oder File Type Error. Der Befehl ist allerdings mit Vorsicht zu genießen, da ein Absturz nicht ausgeschlossen ist.

HELP
Listet alle neuen Befehle auf.

SCRLOAD,string
Lädt den Bildschirmspeicher mit dem Namen string ein.

SCRSAVE,string
Speichert den Bildschirmspeicher unter dem Namen string ab.

ONLOWRES
In Mode 1 und 2 kann jetzt mit der Schriftgröße von Mode 0 geschrieben werden.

NORMAL
Schaltet wieder auf normale Textausgabe um.

TEXTGET,adresse,x,xl,y,yl
Liest das durch x, xl, y, yl bestimmte Fenster charakterweise in den ab adresse beginnenden Speicherbereich ein.

TEXTPUT,adresse,x,xl,y,yl
Der mit TEXTGET eingelesene Bereich wird wieder ausgegeben.

GETCHR,x,y,string
Das sich auf x, y befindende Zeichen wird in string eingelesen.

PROTECT
Schützt ein BASIC-Programm gegen List und Edit.

UNPROTECT
Hebt den Schutz wieder auf.

2. Disketten-Befehle:

GETFORMAT
Ermittelt das Format der Diskette im Laufwerk A und übergibt es an die Befehle READ/WRITE-SECTOR und BACKUP/RESTORE.

DATA/IBM/SYSTEM
Setzt das jeweilige Format für die obengenannten Befehle.

DISCSPEED,s
Setzt den Wert für die Warteschleife beim Laden von Programmen. Je kleiner der Wert, um so größer die Zeitersparnis (1=25%).

READSECTOR,drive,track,sector, adresse
Ein Sektor wird ab adresse gelesen.

WRITESECTOR,drive,track,sector

adresse
Ein Sektor wird ab adresse geschrieben.

FORMAT,drive
Formatiert die Diskette in drive (0/1) im vorher bestimmten Format.

GETHEADER,name,anfangsstring, längestring,startstring
Der Header des Programms name wird in den drei Strings als Hexzahl abgelegt, die vorher mit vier Characters definiert werden müssen.

FILE.DISC,name
Kopiert das File name von Disk auf Disk.

FILE.TAPE,name
Kopiert das File name von Disk auf Tape.

BACKUP
Speichert eine komplette Diskettenseite im Headerless-Format auf Kassette ab. Es werden jeweils fünf Spuren gelesen.

RESTORE
Eine mit BACKUP abgespeicherte Disk wird wieder eingelesen.

3. Kassetten-Befehle:

TAPESPEED,s
Setzt die Baud-Rate (Speed Write). Je kleiner s, um so größer ist die Baud-Rate (s=0 entspricht 4440 Baud). Der Wert von s muß zwischen null und 150 liegen.

HEADERLESS.OUT,anfang,länge
Speichert den mit anfang und länge definierten Bereich im Headerless-Format ab.

HEADERLESS.IN,anfang,länge
Liest den Bereich wieder ein.

4. Grafik-Befehle:

FOCUS,x,y
Bestimmt die Vergrößerung in x- und y-Richtung.

TEXTPOS,x,y
Bestimmt die Anfangsposition für die vergrößerte Schrift (Grafikposition).

TYP,s
Wählt den entsprechenden Typ für die vergrößerte Schrift (s=1 bis 4).

INVERSE
Invertiert die Schrift.

REVERSE
Schaltet die Invertierung wieder ab.

MIRROR.ON/.OFF
Schaltet Spiegelung ein beziehungsweise aus.

STEP,s
Mit s wird der pixelweise Abstand zweier Zeichen festgelegt.

TEXT,string
Vergrößert string mit den eingestellten Faktoren.

FILL,x,y
Eine Fläche wird ab x, y ausgemalt.

CIRCLE,flag,x,y,xradius,yradius
Ein Kreis oder eine Ellipse wird gezeichnet.
flag=0: normale Zeichnung
flag=1: ausgefüllte Zeichnung

BOX,flag,x,y,xversatz,yversatz
Ein Rechteck wird gezeichnet. Für flag gilt das gleiche wie bei CIRCLE.

5. Speicheroperationen:

HEXDUMP,adresse
Gibt Hexdump des ab adresse beginnenden Speicherbereiches aus. Es empfiehlt sich Mode 2.

DOKE,adresse,wert
Poked einen 16-Bit-Wert an adresse.

ROMDUMP,adresse
Wie HEXDUMP, nur bezieht sich adresse auf einen ROM-Bereich.

DEEK,adresse
Liest einen 16-Bit-Wert von adresse.

DEEK,adresse,string
Lädt einen 16-Bit-Wert in den String string.

STROKE,textstring,adresse,textlänge
Legt den Textstring ab adresse im RAM ab.

STREEK,textstring,adresse,textlänge
Liest RAM ab adresse in textstring ein.

6. Bildschirm-Manipulationen:

FREEZE
Warteschleife auf Rasterstrahlrücklauf.

SCROMPRESS,adresse,längestring
Komprimiert den Bildschirmspeicher und legt ihn ab adresse ab. Ein komprimierter Screen ist nur noch fünf bis 14 KByte lang. Die genaue Länge wird in längestring übergeben.

SCREXPAND,adresse
Lädt einen mit SCROMPRESS verkürzten Bildschirm von adresse ein. Der alte Screen wird hinausgeschoben.

SCRSCROLL,adresse
Scrollt einen sich bei adresse befindenden Screen in den Bildschirm, wobei der alte Bildschirm hinausgescrollt wird.

LINESCROLL,zeile,textstring
Scrollt textstring in der angegebenen Zeile über den Bildschirm.

SHAPE,x,y,xversatz,yversatz, shapeadresse,oldscreenadresse, flag
flag=0: Der Inhalt des Speicherbereiches wird in dem durch xversatz und yversatz be-

stimmten Bildschirmausschnitt dargestellt, während der Originalausschnitt bei oldscreenadresse abgelegt wird

flag=1: Der alte Ausschnitt wird wieder zurückgeholt

Der SHAPE-Befehl ist nicht grafik-, sondern bildschirmorientiert. Das heißt, x+xver. darf nicht größer als 80, x+yversatz nicht größer als 200 sein.

TRANS.ON
Transparente Darstellung der Shapes.

TRANS.OFF
Wieder normale Darstellung der Shapes

7. Sonstiges:

FIND,suchadresse,suchtextstring, übergabestring
Der Speicher wird ab suchadressestring nach suchtextstring abgesucht und die entsprechende Adresse in übergabestring übergeben

Wird ein falscher Parameter eingegeben, so erscheint eine Fehlermeldung. Bei den Befehlen GETHEADER, SCROMPRESS, FIND und DEEK muß vor der Anwendung der entsprechende String mit vier Bytes vordefiniert werden. Jedem String muß ein Klammeraffe vorangestellt werden. *TB* □

```
1 '******************************  (24B1)
2 '*      EXTENDED-BASIC        *  (247D)
3 '*            VON             *  (24F2)
4 '*      ANDRE SCHROEDER       *  (2477)
5 '*            FUER            *  (2405)
6 '*     SCHNEIDER CPC-WELT     *  (2476)
7 '*          CPC 464        TB*  (2402)
8 '******************************  (24BF)
10 MEMORY &9564                   (0998)
20 LOAD"EXTEN.BIN"                (113B)
40 CALL &9565                     (0902)
50 MODE 1                         (070C)
60 PRINT SPC(10);"EXTENDED-BASIC 2
.0"                               (20AE)
70 PRINT                          (068B)
71 PRINT SPC(11);"37877 Bytes frei
"                                 (1EAA)
72 PRINT                          (068F)
80 PRINT"/HELP zur Befehls";      (1A6E)
90 PRINT"uebersicht eingeben."    (1C31)
100 NEW                           (068F)
```

```
10 '******************************  (24C3)
20 '*         GRAFIK-DEMO        *  (2417)
30 '*     ZU EXTENDED BASIC      *  (24F8)
40 '*            VON             *  (243D)
50 '*      ANDRE SCHROEDER       *  (24D3)
60 '*            FUER            *  (2492)
65 '*     SCHNEIDER CPC-WELT     *  (24EC)
70 '*          CPC 464        TB*  (2480)
80 '******************************  (244E)
90 '                              (07BF)
100 MEMORY 9999:DEFINT a-z        (0F4A)
110 '                             (07E7)
120 '**** AUFBAU DES SCHLUSSBILDES
                                  (24A3)
130 '                             (070E)
140 BORDER 0:FOR a=0 TO 3:INK a,0:
NEXT                              (1BCA)
150 CALL &BC07,&40:MODE 1:/INVERSE
                                  (1AA4)
160 GOSUB 1060:f=2:t=3:y=350      (2038)
170 GOSUB 1070:/REVERSE           (13A8)
180 GOSUB 1060:f=0:t=2:GOSUB 1070 (1C58)
190 LOCATE 1,25:PRINT STRING$(15,3
2);                               (1630)
200 CALL &BC07,&C0:MODE 1         (1018)
210 '                             (07AE)
220 '**** AUFBAU DES SHAPES       (1D26)
230 '                             (07D6)
240 PLOT-2,-2,2:/BOX,1,10,375,204,
80                                (229D)
250 PLOT-2,-2,3:/BOX,1,2,387,204,8
0                                 (2169)
260 GOSUB 1060:GOSUB 1060         (0E29)
270 f=2:t=3:GOSUB 1070            (179E)
280 PLOT-2,-2,1:/BOX,0,2,387,204,8
0                                 (21FF)
290 /SHAPE,0,5,27,44,&C000,10000,0
                                  (20D6)
300 '                             (0764)
310 MODE 1:INK 1,25:INK 3,15      (13C7)
320 INK 0,18:BORDER 18            (0EBE)
330 FOR a=500 TO 10 STEP-20       (15A9)
340 FOR b=0 TO a:NEXT             (136A)
350 x=RND*50+1:y=RND*150+1        (1EE0)
360 /SHAPE,x,y,27,44,&C000,10000,1
                                  (2619)
370 NEXT:GOSUB 1040               (0BC4)
380 '                             (0703)
390 MODE 1:INK 2,18:PEN 2         (1083)
400 /REVERSE                      (0EF8)
410 GOSUB 1060                    (093B)
420 /MIRROR.ON:y=200:t=2:f=3      (26B8)
430 GOSUB 1070                    (09B3)
440 /MIRROR.OFF:GOSUB 1040        (1698)
450 '                             (078F)
460 MODE 1:GOSUB 1060             (0CF6)
470 F=1:x=20:y=366:t=4:yv=5       (2BE8)
```

```
480 GOSUB 1070                          (2918)
490 GOSUB 1060                          (29DB)
500 f=2:t=3                             (12B9)
510 FOR x=30 TO 34 STEP 2               (12DB)
520 GOSUB 1070:NEXT                     (0B36)
530 |INVERSE                            (0E88)
540 GOSUB 1060                          (0941)
550 f=2:t=2:GOSUB 1070                  (1757)
560 |REVERSE                            (0E39)
570 f=2:t=2:GOSUB 1070                  (177F)
580 t=1:f=3:x=1:GOSUB 1070              (1E3F)
590 LOCATE 1,25                         (0AD6)
600 PRINT STRING$(15,32);               (10E5)
610 INK 2,0:PEN 1:GOSUB 1040            (117E)
620 '                                   (07E5)
630 FOR a=1 TO 20                       (0FFD)
640 x=RND*50+1:y=RND*150+1              (1E24)
650 |SHAPE,x,y,27,44,10000,11500,0
                                        (26A4)
660 FOR b=1 TO 1000:NEXT                (1259)
670 |SHAPE,x,y,27,44,10000,11500,1
                                        (26D0)
680 NEXT:|TRANS.ON                      (11FC)
690 |SHAPE,2,50,70,67,32800,11189,
0                                       (2424)
700 '                                   (0784)
710 a$="   Extended Basic Version
2.0"                                    (2907)
720 a$=a$+STRING$(9,32)                 (179A)
730 FOR a=8 TO 15                       (0F91)
740 |LINESCROLL,a,@a$                   (1C4E)
750 PEN INT(RND*3+1):NEXT               (12E5)
760 GOSUB 1040                          (0958)
770 |CIRCLE,1,275,230,260,67:GOSUB
 1040                                   (2208)
780 |SHAPE,2,50,70,67,32800,11189,
1                                       (24DD)
790 PLOT-2,-2,2                         (0DC1)
800 |FILL,240,260:|FILL,110,220         (1F9E)
810 |FILL,240,200:|FILL,520,220         (1FE6)
820 '                                   (0776)
830 FOR a=40 TO 600 STEP 80             (1496)
840 PLOT-2,-2,INT(RND*3+1)              (164E)
850 x=RND*50+20:y=RND*50+20             (205A)
860 |CIRCLE,1,a,130,x,y                 (21B6)
870 |CIRCLE,1,640-a,310,x,y             (2605)
880 NEXT                                (06A6)
890 FOR a=1 TO 50                       (0F7D)
900 FOR b=1 TO 300:NEXT                 (12A2)
910 FOR c=1 TO 3                        (0EDD)
920 INK c,RND*25+1                      (120E)
930 NEXT:NEXT:INK 1,25                  (0E65)
940 '                                   (0765)
950 INK 2,0:INK 3,15:GOSUB 1040:CL
S                                       (16E1)
960 |SHAPE,0,3,70,67,32800,11189,1
                                        (23E0)
970 |SHAPE,0,122,70,67,32800,11189
,1                                      (2424)
980 GOSUB 1040                          (0911)
990 FOR y=1 TO 151 STEP 50              (12CF)
1000 FOR x=5 TO 40 STEP 35              (12E1)
1010 |SHAPE,x,y,27,44,&C000,12000,
1                                       (2630)
1020 NEXT:NEXT:GOSUB 1040               (0DAB)
1030 |SCRSCROLL,16384:PEN 1:END          (194F)
1040 FOR a=1 TO 5000:NEXT:RETURN        (14DF)
1050 '                                  (0743)
1060 READ t$,x,y,xu,yu,t,f              (2A72)
1070 |TEXTPOS,x,y                       (184F)
1080 |TYP,t:PLOT-2,-2,f                 (1B54)
1090 |FOCUS,xu,yu                       (18B8)
1100 |TEXT,@t$                          (11B6)
1110 RETURN                             (06D7)
1120 '                                  (07CF)
1130 DATA * * * * * * * *,2,150,6,10
,4,3                                    (232F)
1140 DATA ENDE DER DEMO,2,250,6,10
,1,1                                    (23EA)
1150 DATA EXTENDED,6,370,3,3,4,1        (1DC0)
1160 DATA BASIC,2,340,5,4,1,1           (1AD6)
1170 DATA BY ANDRE S.,2,250,7,8,4,
1                                       (20DB)
1180 DATA DAS IST EINE,20,370,6,6,
2,3                                     (22E0)
1190 DATA DEMO,30,250,17,18,1,1         (1C8F)
1200 DATA NICHT WAHR ?,10,80,6,8,1
,1                                      (21E5)
```

```
100 '*****************************    (2234)
110 '*  DATALADER ERZEUGT DAS  *       (22BE)
120 '*  BINAERFILE (EXTEN.BIN) *       (22C6)
130 '*          VON            *       (2293)
140 '*     ANDRE SCHROEDER     *       (22FE)
150 '*          FUER           *       (2228)
160 '*    SCHNEIDER CPC-WELT   *       (22F7)
170 '*        CPC 464        TB*       (2251)
180 '*****************************    (22D4)
200 MEMORY &9564                        (0914)
650 a=&9565:e=&A672:zb=1000:e=e+1       (2C4D)
660 FOR i=a TO e:IF i=e THEN SAVE"
EXTEN.BIN",B,&9565,&110D:END             (3BEF)
670 READ d$:POKE i,VAL("&"+d$)          (1D5A)
730 IF i<e THEN NEXT i                  (1526)
1001 DATA 3A,CE,97,FE,01,C8,3E,01       (1E80)
1002 DATA 32,CE,97,01,79,95,21,CA       (1EB8)
1003 DATA 97,C3,D1,BC,29,96,C3,6D       (1EAA)
1004 DATA A2,C3,7D,A2,C3,CA,A4,C3       (1EAD)
1005 DATA BE,A4,C3,C4,A4,C3,5F,A2       (1EAC)
1006 DATA C3,38,A2,C3,97,A0,C3,2E       (1E81)
1007 DATA 9F,C3,51,A3,C3,9B,A3,C3       (1EA9)
1008 DATA EA,A2,C3,3E,A4,C3,43,A4       (1E22)
1009 DATA C3,B8,A3,C3,D0,A2,C3,E5       (1E79)
1010 DATA A2,C3,D2,A1,C3,D8,A1,C3       (1EE2)
1011 DATA DE,A1,C3,E4,A1,C3,F1,A1       (1ECB)
1012 DATA C3,1F,A4,C3,12,A4,C3,0C       (1E19)
1013 DATA A0,C3,24,A0,C3,D6,A5,C3       (1EB2)
1014 DATA 49,9E,C3,4E,9E,C3,54,9E       (1E1A)
1015 DATA C3,59,9E,C3,F9,9E,C3,FF       (1E61)
1016 DATA 9E,C3,02,9E,C3,0F,9E,C3       (1E98)
1017 DATA E3,9D,C3,CF,9D,C3,05,9F       (1E4A)
1018 DATA C3,16,9F,C3,6B,9D,C3,F2       (1E75)
```

```
1019 DATA 9E,C3,EE,9E,C3,67,9B,C3 <1EC1>
1020 DATA AE,9B,C3,BD,9B,C3,D0,9B <1EF0>
1021 DATA C3,41,9B,C3,47,9B,C3,4D <1ECB>
1022 DATA 9B,C3,7A,9B,C3,ED,9B,C3 <1E92>
1023 DATA 05,9B,C3,ED,9A,C3,23,A6 <1E88>
1024 DATA C3,21,9B,C3,4F,99,C3,CF <1E85>
1025 DATA 97,C3,6A,98,54,45,58,54 <1E8A>
1026 DATA 47,45,D4,54,45,58,54,50 <1E6F>
1027 DATA 55,D4,53,48,41,50,C5,54 <1E03>
1028 DATA 52,41,4E,53,2E,4F,CE,54 <1E76>
1029 DATA 52,41,4E,53,2E,4F,46,C6 <1E43>
1030 DATA 47,45,54,4B,45,D9,47,45 <1EAB>
1031 DATA 54,43,48,D2,53,43,52,53 <1E53>
1032 DATA 43,52,4F,4C,CC,4C,49,4E <1E5A>
1033 DATA 45,53,43,52,4F,4C,CC,53 <1E33>
1034 DATA 43,52,43,4F,4D,50,52,45 <1EE5>
1035 DATA 53,D3,53,43,52,45,58,50 <1ED1>
1036 DATA 41,4E,C4,48,45,58,44,55 <1E48>
1037 DATA 4D,D0,50,52,4F,54,45,43 <1E61>
1038 DATA D4,55,4E,50,52,4F,54,45 <1E0E>
1039 DATA 43,D4,47,45,54,48,45,41 <1EE3>
1040 DATA 44,45,D2,54,41,50,45,53 <1E8C>
1041 DATA 50,45,45,C4,44,49,53,43 <1EF8>
1042 DATA 53,50,45,45,C4,44,41,54 <1E71>
1043 DATA C1,53,59,53,54,45,CD,49 <1ED7>
1044 DATA 42,CD,52,45,41,44,53,45 <1E61>
1045 DATA 43,54,4F,D2,57,52,49,54 <1E0B>
1046 DATA 45,53,45,43,54,4F,D2,48 <1EC4>
1047 DATA 45,41,44,45,52,4C,45,53 <1EEB>
1048 DATA 53,2E,4F,55,D4,48,45,41 <1E13>
1049 DATA 44,45,52,4C,45,53,53,2E <1EEC>
1050 DATA 49,CE,46,49,4C,45,2E,54 <1E04>
1051 DATA 41,50,C5,46,49,4C,45,2E <1EB6>
1052 DATA 44,49,53,C3,46,4F,52,4D <1EF9>
1053 DATA 41,D4,42,52,45,41,4B,2E <1E47>
1054 DATA 4F,CE,42,52,45,41,4B,2E <1E1C>
1055 DATA 4F,46,C6,42,41,43,4B,55 <1EE0>
1056 DATA D0,52,45,53,54,4F,52,C5 <1E23>
1057 DATA 45,52,52,4F,52,2E,4F,CE <1E22>
1058 DATA 45,52,52,4F,52,2E,4F,46 <1E71>
1059 DATA C6,53,54,52,45,45,CB,53 <1EB3>
1060 DATA 54,52,4F,4B,C5,44,45,45 <1E95>
1061 DATA CB,44,4F,4B,C5,4F,4E,4C <1E1D>
1062 DATA 4F,57,52,45,D3,4E,4F,52 <1E3E>
1063 DATA 4D,41,CC,46,49,4E,C4,52 <1ED8>
1064 DATA 4F,4D,44,55,4D,D0,46,52 <1EA9>
1065 DATA 45,45,5A,C5,46,4F,43,55 <1EE3>
1066 DATA D3,4D,49,52,52,4F,52,2E <1E0F>
1067 DATA 4F,CE,4D,49,52,52,4F,52 <1EF8>
1068 DATA 2E,4F,46,C6,53,54,45,D0 <1E3B>
1069 DATA 49,4E,56,45,52,53,C5,52 <1E0C>
1070 DATA 45,56,45,52,53,C5,54,45 <1E3F>
1071 DATA 58,54,50,4F,D3,54,59,D0 <1ED3>
1072 DATA 54,45,58,D4,53,43,52,4C <1E1C>
1073 DATA 4F,41,C4,53,43,52,53,41 <1E52>
1074 DATA 56,C5,48,45,4C,D0,47,45 <1E45>
1075 DATA 54,46,4F,52,4D,41,D4,43 <1EE6>
1076 DATA 49,52,43,4C,C5,42,4F,D8 <1E02>
1077 DATA 46,49,4C,CC,00,FC,A6,79 <1E95>
1078 DATA 95,00,06,05,CD,5B,A1,FE <1E6B>
1079 DATA 00,C8,ED,53,68,98,DD,66 <1EC2>
1080 DATA 03,DD,6E,02,22,66,98,DD <1E70>
1081 DATA 66,05,DD,6E,04,DD,56,07 <1EFB>
1082 DATA DD,5E,06,DD,7E,08,FE,01 <1E34>
1083 DATA CA,22,98,CD,58,98,ED,4B <1E03>
1084 DATA 66,98,EB,09,EB,CD,4E,98 <1E84>
1085 DATA ED,4B,68,98,ED,42,CD,4E <1EDC>
1086 DATA 98,ED,4B,66,98,EB,ED,42 <1ECA>
1087 DATA EB,CD,4E,98,ED,4B,68,98 <1E41>
1088 DATA 09,CD,4E,98,C9,ED,53,62 <1E7F>
1089 DATA 98,ED,4B,66,98,EB,09,EB <1E07>
1090 DATA ED,53,64,98,ED,4B,68,98 <1E26>
1091 DATA ED,5B,62,98,CD,58,98,ED <1E55>
1092 DATA 5B,64,98,CD,4E,98,2B,2B <1E92>
1093 DATA 0B,0B,78,FE,FF,C8,C3,35 <1E90>
1094 DATA 98,C5,E5,D5,CD,F6,BB,D1 <1E24>
1095 DATA E1,C1,C9,C5,E5,D5,CD,C0 <1EF3>
1096 DATA BB,D1,E1,C1,C9,00,00,00 <1EF3>
1097 DATA 00,00,00,00,00,06,02,CD <1E71>
1098 DATA 5B,A1,FE,00,C8,ED,53,47 <1ED4>
1099 DATA 99,DD,66,03,DD,6E,02,22 <1EE1>
1100 DATA 45,99,3A,C8,B1,FE,00,C2 <1E68>
1101 DATA 8C,98,3E,04,C3,98,98,FE <1E96>
1102 DATA 01,C2,96,98,3E,02,C3,98 <1EF5>
1103 DATA 98,3E,01,32,49,99,3E,00 <1E80>
1104 DATA 32,4B,99,2A,47,99,22,41 <1EA9>
1105 DATA 99,21,23,23,22,07,99,2A <1ECE>
1106 DATA 45,99,22,43,99,3E,00,32 <1EA9>
1107 DATA 4C,99,EB,2A,41,99,CD,F0 <1EB7>
1108 DATA BB,FE,00,C2,25,99,2A,43 <1E83>
1109 DATA 99,ED,4B,49,99,3A,4C,99 <1EFA>
1110 DATA FE,01,20,04,ED,42,18,01 <1EA9>
1111 DATA 09,22,43,99,EB,2A,41,99 <1E8C>
1112 DATA CD,F0,BB,FE,00,20,06,7C <1E03>
1113 DATA FE,03,D8,18,D9,3A,4C,99 <1E1A>
1114 DATA FE,01,C2,0F,99,ED,5B,43 <1E49>
1115 DATA 99,2A,41,99,E5,CD,C0,BB <1E47>
1116 DATA ED,5B,4D,99,E1,E5,CD,F6 <1E33>
1117 DATA BB,E1,23,23,22,41,99,C3 <1E6B>
1118 DATA AC,98,3E,01,32,4C,99,2A <1E7F>
1119 DATA 43,99,22,4D,99,ED,5B,45 <1EB6>
1120 DATA 99,ED,53,43,99,C3,DA,98 <1E04>
1121 DATA 3A,4B,99,FE,01,C8,21,2B <1E39>
1122 DATA 2B,22,07,99,3E,01,32,4B <1E95>
1123 DATA 99,2A,47,99,2B,2B,22,41 <1E41>
1124 DATA 99,C3,AC,98,00,00,00,00 <1E0F>
1125 DATA 00,00,00,00,00,00,00,00 <1E70>
1126 DATA 00,00,06,05,CD,5B,A1,FE <1E0B>
1127 DATA 00,C8,21,00,80,22,E5,9A <1EDC>
1128 DATA ED,53,DD,9A,DD,66,03,DD <1E19>
1129 DATA 6E,02,22,DF,9A,DD,66,05 <1ED2>
1130 DATA DD,6E,04,22,E3,9A,DD,66 <1EFF>
1131 DATA 07,DD,6E,06,22,E1,9A,DD <1ED7>
1132 DATA 7E,08,FE,01,CA,89,99,3E <1EA8>
1133 DATA E9,C3,8B,99,3E,F6,32,0D <1EB3>
1134 DATA 9A,32,36,9A,32,61,9A,32 <1E2F>
1135 DATA 8C,9A,06,0A,2A,DF,9A,CD <1E11>
1136 DATA D3,9A,ED,5B,DD,9A,CD,C1 <1EBC>
1137 DATA BD,22,EB,9A,21,00,00,22 <1E57>
1138 DATA DF,9A,2A,DF,9A,ED,5B,DD <1E07>
1139 DATA 9A,7C,BA,C2,C0,99,13,13 <1E71>
1140 DATA 7D,BB,D0,2A,DF,9A,3A,EB <1E8F>
1141 DATA 9A,47,CD,D3,9A,11,0A,00 <1E60>
1142 DATA CD,C1,BD,22,E7,9A,2A,DD <1E87>
```

```
1143 DATA 9A,3A,EB,9A,47,CD,D3,9A <1E5D>
1144 DATA 11,0A,00,CD,C1,BD,22,E9 <1EDD>
1145 DATA 9A,2A,E1,9A,ED,5B,E7,9A <1E35>
1146 DATA 19,EB,2A,E3,9A,ED,4B,DD <1E70>
1147 DATA 9A,09,CD,E9,BB,2A,E1,9A <1E07>
1148 DATA ED,5B,E7,9A,ED,52,EB,2A <1EAA>
1149 DATA E3,9A,ED,4B,DD,9A,09,CD <1E83>
1150 DATA F6,BB,2A,E1,9A,ED,5B,E9 <1EFE>
1151 DATA 9A,19,EB,2A,E3,9A,ED,4B <1E86>
1152 DATA DF,9A,09,CD,E9,BB,2A,E1 <1EB9>
1153 DATA 9A,ED,5B,E9,9A,ED,52,EB <1E8F>
1154 DATA 2A,E3,9A,ED,4B,DF,9A,09 <1EEB>
1155 DATA CD,F6,BB,2A,E1,9A,ED,5B <1E71>
1156 DATA E7,9A,ED,52,EB,2A,E3,9A <1E26>
1157 DATA ED,4B,DD,9A,ED,42,CD,E9 <1EFE>
1158 DATA BB,2A,E1,9A,ED,5B,E7,9A <1E73>
1159 DATA 19,EB,2A,E3,9A,ED,4B,DD <1E8A>
1160 DATA 9A,ED,42,CD,F6,BB,2A,E1 <1EBA>
1161 DATA 9A,ED,5B,E9,9A,ED,52,EB <1E9F>
1162 DATA 2A,E3,9A,ED,4B,DF,9A,ED <1ECE>
1163 DATA 42,CD,E9,BB,2A,E1,9A,ED <1EA3>
1164 DATA 5B,E9,9A,19,EB,2A,E3,9A <1E77>
1165 DATA ED,4B,DF,9A,ED,42,CD,F6 <1E0F>
1166 DATA BB,2A,E5,9A,7C,FE,80,D2 <1EBD>
1167 DATA AE,9A,2A,DF,9A,06,04,CD <1E36>
1168 DATA D3,9A,01,06,00,09,EB,2A <1EF5>
1169 DATA E5,9A,19,22,E5,9A,C3,CC <1E29>
1170 DATA 9A,2A,DF,9A,ED,5B,DD,9A <1E6E>
1171 DATA ED,52,06,04,CD,D3,9A,01 <1E43>
1172 DATA 0A,00,09,EB,2A,E5,9A,19 <1ECC>
1173 DATA 22,E5,9A,21,DD,9A,35,21 <1E5E>
1174 DATA DF,9A,34,C3,AF,99,54,5D <1E7E>
1175 DATA 21,00,00,0B,19,10,FD,C9 <1E80>
1176 DATA 45,00,47,00,40,01,C8,00 <1EA7>
1177 DATA 64,80,46,00,46,00,0A,00 <1ED4>
1178 DATA CD,15,9B,CD,8C,BC,CD,7A <1E5D>
1179 DATA BC,21,00,C0,11,00,40,3E <1E43>
1180 DATA 02,CD,98,BC,CD,8F,BC,C9 <1EF4>
1181 DATA CD,15,9B,CD,77,BC,21,00 <1EB5>
1182 DATA C0,CD,83,BC,CD,7A,BC,C9 <1E46>
1183 DATA 1A,47,13,1A,6F,13,1A,67 <1E2E>
1184 DATA 11,00,C0,C9,3E,C9,32,5A <1E83>
1185 DATA BB,21,3E,9B,CD,D4,BC,D0 <1E28>
1186 DATA AF,CD,1B,00,3A,9F,A8,3D <1E49>
1187 DATA 32,37,A2,3E,CF,32,5A,BB <1ED8>
1188 DATA C9,44,49,D2,3E,C2,32,6A <1E4E>
1189 DATA 9C,C9,3E,CA,32,6A,9C,C9 <1EEB>
1190 DATA 06,02,CD,5B,A1,FE,00,C8 <1EC4>
1191 DATA ED,53,E0,9B,ED,53,E2,9B <1E64>
1192 DATA DD,66,03,DD,6E,02,22,DE <1E6C>
1193 DATA 9B,C9,06,02,CD,5B,A1,FE <1EB3>
1194 DATA 00,C8,ED,53,E4,9B,DD,7E <1E4B>
1195 DATA 02,32,E6,9B,C9,06,01,CD <1E25>
1196 DATA 5B,A1,FE,00,C8,7B,FE,01 <1E9A>
1197 DATA C2,8F,9B,21,F5,9C,22,EB <1E5F>
1198 DATA 9C,C9,FE,02,C2,9B,9B,21 <1E45>
1199 DATA 14,9D,22,EB,9C,C9,FE,03 <1E93>
1200 DATA C2,A7,9B,21,29,9D,22,EB <1EC9>
1201 DATA 9C,C9,21,48,9D,22,EB,9C <1EA0>
1202 DATA C9,3E,C6,32,7D,9C,3E,0F <1EA3>
1203 DATA 32,79,9C,3E,01,C3,C9,9B <1EA8>
1204 DATA 3E,D6,32,7D,9C,3E,01,32 <1E8A>
1205 DATA 79,9C,3E,0F,32,F7,9B,32 <1EF8>
1206 DATA 86,9C,C9,7B,32,D5,9B,C9 <1ED6>
1207 DATA 00,00,0F,00,00,00,20,22 <1EB7>
1208 DATA 48,02,00,C8,00,4D,22,08 <1E73>
1209 DATA 00,08,00,30,20,27,44,42 <1EFB>
1210 DATA 06,01,CD,5B,A1,FE,00,C8 <1E96>
1211 DATA 00,3E,0F,32,D7,9B,1A,47 <1E49>
1212 DATA 3D,32,DC,9B,3E,00,32,DD <1E5D>
1213 DATA 9B,32,DB,9B,32,D9,9B,3A <1E2D>
1214 DATA C8,B1,FE,00,C2,20,9C,21 <1EBA>
1215 DATA 34,34,22,8D,9C,22,8F,9C <1E5D>
1216 DATA C3,40,9C,FE,01,C2,34,9C <1E29>
1217 DATA 21,34,34,22,8D,9C,21,00 <1E9A>
1218 DATA 00,22,8F,9C,C3,40,9C,21 <1E19>
1219 DATA 34,00,22,8D,9C,21,00,00 <1E59>
1220 DATA 22,8F,9C,13,1A,6F,13,1A <1E9C>
1221 DATA 67,3E,1F,CD,5A,BB,3E,01 <1E43>
1222 DATA CD,5A,BB,3E,19,CD,5A,BB <1E16>
1223 DATA 7E,CD,5A,BB,23,10,F9,2A <1E69>
1224 DATA D7,9B,ED,5B,D9,9B,E5,D5 <1EF2>
1225 DATA CD,F0,BB,FE,00,CA,70,9C <1EC7>
1226 DATA CD,CE,9C,CD,56,9D,D1,E1 <1EFC>
1227 DATA 3A,D7,9B,FE,01,CA,85,9C <1E12>
1228 DATA D6,02,32,D7,9B,C3,5C,9C <1EDB>
1229 DATA 3E,0F,32,D7,9B,21,D9,9B <1ECA>
1230 DATA 34,34,00,00,2A,DE,9B,ED <1EB3>
1231 DATA 5B,E6,9B,19,22,DE,9B,2A <1E91>
1232 DATA E2,9B,22,E0,9B,3A,DB,9B <1E5C>
1233 DATA 3C,32,DB,9B,FE,08,C2,5C <1E50>
1234 DATA 9C,3A,DC,9B,47,3A,DD,9B <1ED3>
1235 DATA B8,C8,3C,32,DD,9B,3E,00 <1E27>
1236 DATA 32,DB,9B,2A,DE,9B,ED,4B <1E73>
1237 DATA D5,9B,09,22,DE,9B,C3,5C <1E8C>
1238 DATA 9C,ED,5B,DE,9B,2A,E0,9B <1E62>
1239 DATA 3A,E6,9B,06,00,4F,CD,63 <1E93>
1240 DATA 9D,3A,E4,9B,4F,09,22,EA <1E3D>
1241 DATA 9B,ED,53,E8,9B,CD,F5,9C <1E78>
1242 DATA C9,2A,EA,9B,ED,5B,E8,9B <1EF7>
1243 DATA E5,D5,06,00,09,C5,CD,C0 <1E2D>
1244 DATA BB,3A,E6,9B,4F,CD,63,9D <1ED6>
1245 DATA CD,F6,BB,C1,D1,E1,0B,0B <1E44>
1246 DATA 79,FE,FE,D0,C3,F5,9C,CD <1EB7>
1247 DATA C0,BB,3A,E6,9B,06,00,4F <1EC6>
1248 DATA CD,63,9D,3A,E4,9B,4F,09 <1EF0>
1249 DATA CD,F6,BB,C9,CD,C0,BB,3A <1E48>
1250 DATA E6,9B,06,00,4F,CD,63,9D <1EBB>
1251 DATA E5,D5,CD,F6,BB,D1,E1,CD <1E45>
1252 DATA C0,BB,3A,E4,9B,4F,09,CD <1E29>
1253 DATA F6,BB,C9,CD,C0,BB,3A,E4 <1E1E>
1254 DATA 9B,06,00,4F,09,CD,F6,BB <1E10>
1255 DATA C9,2A,E0,9B,ED,5B,E4,9B <1EA9>
1256 DATA ED,52,22,E0,9B,C9,E5,62 <1E49>
1257 DATA 6B,09,54,5D,E1,C9,06,03 <1E74>
1258 DATA CD,5B,A1,FE,00,C8,ED,53 <1E8B>
1259 DATA C8,9D,DD,66,05,DD,6E,04 <1EAA>
1260 DATA 22,CD,9D,DD,66,03,DD,6E <1E49>
1261 DATA 02,7E,32,CA,9D,23,7E,5F <1E21>
1262 DATA 23,7E,57,ED,53,CB,9D,7C <1EC9>
1263 DATA FE,C0,C8,C3,A5,9D,2A,CD <1E64>
1264 DATA 9D,23,22,CD,9D,C3,94,9D <1E35>
1265 DATA 3A,CA,9D,47,ED,5B,CB,9D <1E28>
1266 DATA 2A,CD,9D,1A,BE,C2,9B,9D <1E8F>
```

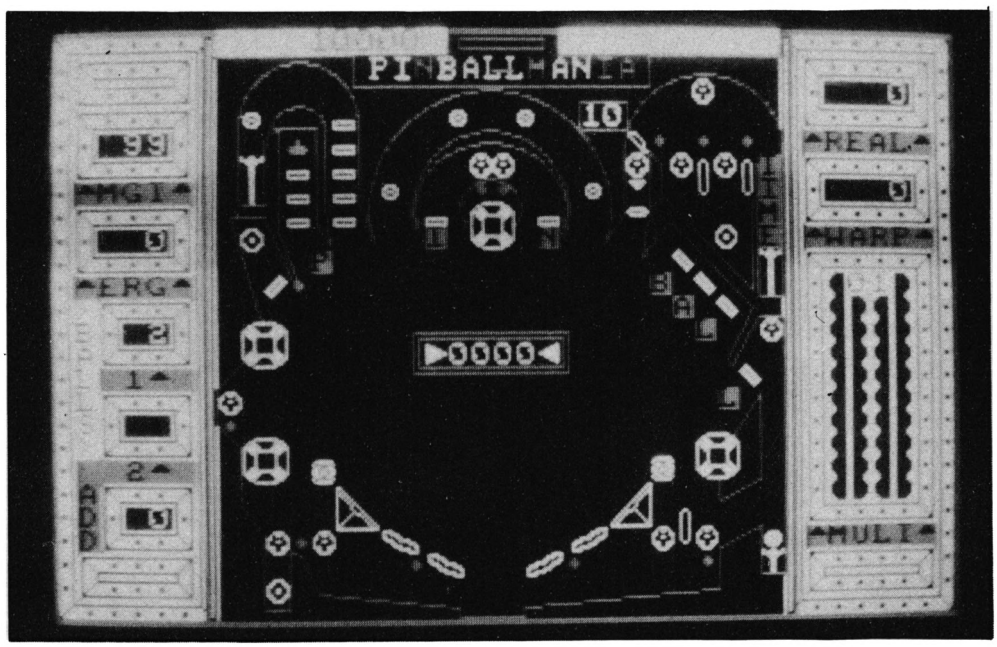
```
1267 DATA 23,13,05,78,FE,00,C2,B0 <1EA4>
1268 DATA 9D,ED,5B,C8,9D,2A,CD,9D <1EB0>
1269 DATA C3,2B,A1,07,02,05,0C,27 <1E41>
1270 DATA 0C,27,06,02,CD,5B,A1,FE <1E04>
1271 DATA 00,C8,DD,66,03,DD,6E,02 <1E10>
1272 DATA 7B,77,23,7A,77,C9,FE,02 <1EAA>
1273 DATA C2,F7,9D,DD,46,03,DD,4E <1E4D>
1274 DATA 02,0A,6F,03,0A,67,CD,2B <1EFB>
1275 DATA A1,C9,13,1A,CD,28,A3,1B <1EA1>
1276 DATA 1A,CD,28,A3,C9,06,03,CD <1ED4>
1277 DATA 5B,A1,FE,00,C8,3E,01,C3 <1ED0>
1278 DATA 19,9E,06,03,CD,5B,A1,FE <1E00>
1279 DATA 00,C8,3E,00,32,48,9E,7B <1EE7>
1280 DATA 32,47,9E,DD,66,03,DD,6E <1E45>
1281 DATA 02,DD,56,05,DD,5E,04,12 <1E73>
1282 DATA 13,1A,4F,13,1A,57,79,5F <1EE7>
1283 DATA 3A,48,9E,FE,00,C2,3E,9E <1EB5>
1284 DATA EB,3A,47,9E,06,00,4F,ED <1E7E>
1285 DATA B0,C9,00,00,3E,C3,C3,50 <1E11>
1286 DATA 9E,3E,C9,32,EE,BD,C9,3E <1EF3>
1287 DATA 00,C3,5B,9E,3E,01,32,E9 <1E88>
1288 DATA 9E,06,08,3E,00,32,EA,9E <1E44>
1289 DATA C5,3A,E9,9E,FE,01,CA,7E <1EB9>
1290 DATA 9E,C3,75,9E,C1,10,F1,C9 <1EB1>
1291 DATA CD,87,9E,CD,D8,9E,C3,71 <1E60>
1292 DATA 9E,CD,D3,9E,CD,87,9E,C3 <1E44>
1293 DATA 71,9E,21,9D,41,3A,EA,9E <1E01>
1294 DATA 57,C6,05,32,EA,9E,0E,09 <1E29>
1295 DATA 3A,37,A2,81,4F,CD,BB,9E <1E4F>
1296 DATA CD,AB,9E,3A,37,A2,3C,B9 <1E7C>
1297 DATA 28,0B,0D,C3,9A,9E,D5,11 <1E98>
1298 DATA 00,02,19,D1,C9,14,3A,EA <1E27>
1299 DATA 9E,BA,C8,C3,93,9E,1E,00 <1EE0>
1300 DATA 3A,E9,9E,FE,01,CA,CA,9E <1EFA>
1301 DATA 3E,3C,C3,CC,9E,3E,3F,32 <1E30>
1302 DATA EB,9E,DF,EB,9E,C9,3E,A1 <1E73>
1303 DATA C3,DA,9E,3E,9E,32,E6,9E <1E3E>
1304 DATA 3E,16,21,9D,41,11,00,5A <1E48>
1305 DATA CD,00,BC,C9,00,00,00,C0 <1E28>
1306 DATA 07,CD,19,BD,C9,DF,F6,9E <1E23>
1307 DATA C9,EA,A2,FC,3E,C9,32,04 <1E29>
1308 DATA AC,C9,3E,F1,32,04,AC,C9 <1E03>
1309 DATA 3A,C8,B1,32,2C,9F,3E,00 <1EAF>
1310 DATA 32,C8,B1,3E,FF,32,D0,B1 <1E31>
1311 DATA C9,3A,2C,9F,32,C8,B1,FE <1E32>
1312 DATA 01,C2,26,9F,3E,44,C3,28 <1E24>
1313 DATA 9F,3E,40,32,D0,B1,C9,00 <1ED4>
1314 DATA 00,06,02,CD,5B,A1,FE,00 <1EB7>
1315 DATA C8,1A,32,0A,A0,13,1A,6F <1E4C>
1316 DATA 13,1A,67,22,FD,9F,DD,7E <1E96>
1317 DATA 02,CD,BE,9F,3A,C8,B1,47 <1EB5>
1318 DATA CD,D2,9F,3E,08,32,FF,9F <1E6E>
1319 DATA CD,19,BD,3A,09,A0,47,3A <1EF5>
1320 DATA FB,9F,B8,CA,90,9F,3C,32 <1E86>
1321 DATA FB,9F,2A,05,A0,22,00,A0 <1EAD>
1322 DATA 2A,00,A0,11,00,08,19,22 <1ED7>
1323 DATA 00,A0,54,5D,23,01,4F,00 <1E4E>
1324 DATA ED,B0,36,00,3A,FF,9F,3D <1EFF>
1325 DATA FE,00,CA,50,9F,32,FF,9F <1E11>
1326 DATA C3,6D,9F,3E,00,32,FB,9F <1EE4>
1327 DATA 3E,1F,CD,5A,BB,3A,07,A0 <1E21>
1328 DATA CD,5A,BB,3A,08,A0,CD,5A <1E27>
1329 DATA BB,3A,0A,A0,FE,00,C8,3D <1EBA>
1330 DATA 32,0A,A0,2A,FD,9F,7E,CD <1E1D>
1331 DATA 5A,BB,23,22,FD,9F,C3,67 <1E62>
1332 DATA 9F,32,08,A0,21,B0,B7,11 <1E3B>
1333 DATA 50,00,19,3D,FE,00,C2,C7 <1E49>
1334 DATA 9F,22,05,A0,C9,3E,13,32 <1E6A>
1335 DATA 07,A0,3E,03,32,09,A0,78 <1E28>
1336 DATA FE,01,C2,ED,9F,3E,27,32 <1E62>
1337 DATA 07,A0,3E,01,32,09,A0,78 <1EA9>
1338 DATA FE,02,C0,3E,4F,32,07,A0 <1EBF>
1339 DATA 3E,00,32,09,A0,C9,44,4C <1E5E>
1340 DATA 4F,4B,42,49,4E,01,00,FF <1E81>
1341 DATA 53,50,49,52,49,54,32,D5 <1EF5>
1342 DATA 21,1C,A0,CD,D4,BC,D0,AF <1E48>
1343 DATA CD,1B,00,D1,C3,24,A0,54 <1EA1>
1344 DATA 41,50,45,2E,4F,55,D4,1A <1E30>
1345 DATA 32,58,AF,13,1A,6F,13,1A <1ECE>
1346 DATA 67,11,00,BF,3A,58,AF,06 <1E33>
1347 DATA 00,4F,ED,B0,21,47,A0,11 <1E31>
1348 DATA 00,AF,01,50,00,ED,B0,C3 <1E7E>
1349 DATA 00,AF,21,00,BF,3A,58,AF <1E4E>
1350 DATA 47,11,00,A7,CD,77,BC,2A <1E9F>
1351 DATA 6D,A7,22,50,AF,2A,6F,A7 <1EFA>
1352 DATA 22,52,AF,3A,67,A7,32,54 <1EFE>
1353 DATA AF,2A,6A,A7,22,56,AF,CD <1E87>
1354 DATA 83,BC,CD,7A,BC,CD,06,BB <1EE5>
1355 DATA 21,00,BF,3A,58,AF,47,11 <1E71>
1356 DATA 00,C0,CD,8C,BC,2A,56,AF <1E9E>
1357 DATA ED,5B,50,AF,ED,4B,52,AF <1E4C>
1358 DATA 3A,54,AF,CD,98,BC,CD,8F <1E81>
1359 DATA BC,C9,06,01,CD,5B,A1,FE <1EBC>
1360 DATA 00,C8,21,00,C0,ED,52,22 <1E3C>
1361 DATA 28,A1,3E,00,32,27,A1,3E <1EEF>
1362 DATA 08,32,26,A1,21,B0,BF,22 <1E7E>
1363 DATA 22,A1,22,20,A1,CD,F6,A0 <1E55>
1364 DATA 54,5D,23,01,4F,00,ED,B0 <1EB3>
1365 DATA 2B,44,4D,2A,22,A1,ED,5B <1E87>
1366 DATA 28,A1,ED,52,3A,27,A1,16 <1E80>
1367 DATA 00,5F,19,7E,02,2A,22,A1 <1E88>
1368 DATA 7C,FE,FF,C2,BA,A0,7D,FE <1EA3>
1369 DATA 80,C2,BA,A0,3A,27,A1,FE <1ED3>
1370 DATA 50,C8,3C,32,27,A1,C3,AC <1E0C>
1371 DATA A0,3A,26,A1,2A,22,A1,FE <1EF7>
1372 DATA 08,C2,14,A1,3E,01,32,26 <1ECD>
1373 DATA A1,11,50,00,2A,20,A1,19 <1EC4>
1374 DATA 22,22,A1,22,20,A1,C9,11 <1E2D>
1375 DATA 00,08,19,22,22,A1,3C,32 <1E3D>
1376 DATA 26,A1,C9,00,00,00,00,00 <1E05>
1377 DATA 00,00,00,00,00,00,3E,04 <1EEE>
1378 DATA 12,13,1A,4F,13,1A,47,7C <1EE2>
1379 DATA 0F,0F,0F,0F,CD,4D,A1,7C <1E8F>
1380 DATA CD,4D,A1,7D,0F,0F,0F,0F <1E0E>
1381 DATA CD,4D,A1,7D,CD,4D,A1,C9 <1E4F>
1382 DATA E6,0F,FE,0A,DA,56,A1,C6 <1E76>
1383 DATA 07,C6,30,02,03,C9,B8,DA <1E2A>
1384 DATA 69,A1,C2,6F,A1,C9,21,B9 <1E17>
1385 DATA A1,C3,72,A1,21,7D,A1,C3 <1E17>
1386 DATA 72,A1,21,9A,A1,7E,FE,00 <1E44>
1387 DATA C8,CD,5A,BB,23,C3,72,A1 <1E12>
1388 DATA 0A,0D,2A,2A,2A,20,50,41 <1EB0>
1389 DATA 52,41,4D,45,54,45,52,20 <1E74>
1390 DATA 4D,49,53,53,49,4E,47,20 <1E51>
```

```
1391 DATA 2A,2A,2A,07,00,0A,0D,2A <1E8F>
1392 DATA 2A,2A,20,54,4F,4F,20,4D <1E0E>
1393 DATA 41,4E,59,20,50,41,52,41 <1EEF>
1394 DATA 4D,45,54,45,52,53,20,2A <1E5B>
1395 DATA 2A,2A,07,00,0A,0D,2A,2A <1E3A>
1396 DATA 2A,20,42,41,44,20,50,41 <1EAD>
1397 DATA 52,41,4D,45,54,45,52,20 <1E84>
1398 DATA 2A,2A,2A,07,00,3E,C0,32 <1EBD>
1399 DATA 37,A2,C9,3E,40,32,37,A2 <1E14>
1400 DATA C9,3E,00,32,37,A2,C9,06 <1E39>
1401 DATA 04,CD,5B,A1,FE,00,C8,3E <1E8F>
1402 DATA 84,C3,FB,A1,06,04,CD,5B <1E91>
1403 DATA A1,FE,00,C8,3E,85,32,33 <1EB4>
1404 DATA A2,ED,53,2D,A2,DD,7E,02 <1E3D>
1405 DATA 21,37,A2,86,32,2B,A2,DD <1E9F>
1406 DATA 7E,04,32,29,A2,DD,7E,06 <1EF2>
1407 DATA 32,27,A2,21,33,A2,CD,D4 <1E5B>
1408 DATA BC,D0,22,34,A2,79,32,36 <1E2A>
1409 DATA A2,1E,00,16,03,0E,41,21 <1E70>
1410 DATA 30,75,DF,34,A2,C9,84,3C <1ED7>
1411 DATA C0,07,40,06,03,CD,5B,A1 <1E00>
1412 DATA FE,00,C8,3E,01,12,13,1A <1E37>
1413 DATA 6F,13,1A,67,3E,1F,CD,5A <1EBF>
1414 DATA BB,DD,7E,04,CD,5A,BB,DD <1E84>
1415 DATA 7E,02,CD,5A,BB,CD,60,BB <1E1F>
1416 DATA 77,C9,3E,01,12,13,1A,6F <1E29>
1417 DATA 13,1A,67,CD,06,BB,77,C9 <1EB4>
1418 DATA 06,05,CD,5B,A1,FE,00,C8 <1EDD>
1419 DATA 3E,00,32,CF,A2,C3,8A,A2 <1E74>
1420 DATA 06,05,CD,5B,A1,FE,00,C8 <1EE1>
1421 DATA 3E,01,32,CF,A2,DD,6E,08 <1E2D>
1422 DATA DD,66,09,DD,7E,06,47,32 <1E8D>
1423 DATA CE,A2,DD,4E,04,DD,56,02 <1E24>
1424 DATA 3E,1F,CD,5A,BB,78,CD,5A <1E2D>
1425 DATA BB,7A,CD,5A,BB,3A,CF,A2 <1E54>
1426 DATA B7,C2,C7,A2,CD,60,BB,77 <1E5D>
1427 DATA 23,78,04,B9,C2,9D,A2,3A <1EDC>
1428 DATA CE,A2,47,7A,14,BB,C2,9D <1E19>
1429 DATA A2,C9,7E,CD,5A,BB,C3,B5 <1E8A>
1430 DATA A2,00,00,7B,FE,A4,DA,DA <1EC5>
1431 DATA A2,CD,63,A1,C9,C6,5C,26 <1ED1>
1432 DATA 00,6F,3E,0A,CD,68,BC,C9 <1EF4>
1433 DATA 7B,32,49,BE,C9,0E,10,D5 <1E79>
1434 DATA 7A,CD,28,A3,7B,CD,28,A3 <1EE1>
1435 DATA CD,47,A3,06,10,1A,CD,28 <1EAA>
1436 DATA A3,3E,20,CD,5A,BB,13,10 <1EC8>
1437 DATA F4,CD,47,A3,D1,06,10,1A <1E1B>
1438 DATA FE,20,DC,25,A3,FE,7F,D4 <1E4B>
1439 DATA 25,A3,CD,5A,BB,13,10,EF <1EA1>
1440 DATA 3E,00,0D,B9,C8,C3,EC,A2 <1E32>
1441 DATA 3E,2E,C9,6F,0F,0F,0F,0F <1EB0>
1442 DATA E6,0F,CD,3A,A3,7B,7D,E6 <1E2C>
1443 DATA 0F,CD,3A,A3,C9,FE,0A,DA <1EF7>
1444 DATA 41,A3,C6,07,C6,30,CD,5A <1E9D>
1445 DATA BB,C9,06,06,3E,20,CD,5A <1E72>
1446 DATA BB,10,FB,C9,06,02,CD,5B <1E8F>
1447 DATA A1,FE,00,C8,DD,66,03,DD <1E41>
1448 DATA 6E,02,22,B6,A3,D5,11,00 <1E01>
1449 DATA C0,06,01,1A,4F,13,1A,B9 <1E2F>
1450 DATA C2,8D,A3,3E,FF,B8,CA,8D <1EA0>
1451 DATA A3,04,13,3E,00,BA,C2,6B <1EA4>
1452 DATA A3,23,3E,00,77,ED,5B,B6 <1E16>
1453 DATA A3,ED,52,D1,CD,2B,A1,C9 <1E29>
1454 DATA 79,77,23,78,77,23,3E,01 <1E7F>
1455 DATA 47,1A,4F,C3,77,A3,62,6B <1EDC>
1456 DATA 11,00,C0,7E,4F,23,7E,47 <1EAA>
1457 DATA 23,FE,00,C8,79,12,13,05 <1E7C>
1458 DATA 78,FE,00,C2,A9,A3,C3,A0 <1E73>
1459 DATA A3,00,00,06,04,CD,5B,A1 <1E68>
1460 DATA FE,00,C8,ED,53,0C,A4,DD <1E35>
1461 DATA 6E,02,DD,66,03,22,0E,A4 <1EC0>
1462 DATA DD,6E,04,DD,66,05,22,10 <1E24>
1463 DATA A4,DD,5E,06,DD,56,07,1A <1E8F>
1464 DATA 47,13,1A,6F,13,1A,67,11 <1E02>
1465 DATA 00,C0,CD,77,BC,CD,7A,BC <1EB7>
1466 DATA 2A,6F,A7,ED,5B,0C,A4,CD <1EF7>
1467 DATA 2B,A1,2A,6D,A7,ED,5B,0E <1ED0>
1468 DATA A4,CD,2B,A1,2A,6A,A7,ED <1E04>
1469 DATA 5B,10,A4,CD,2B,A1,C9,00 <1EA2>
1470 DATA 00,00,00,00,00,06,02,CD <1E5B>
1471 DATA 5B,A1,FE,00,C8,3E,A1,C3 <1E89>
1472 DATA 29,A4,06,02,CD,5B,A1,FE <1ED6>
1473 DATA 00,C8,3E,9E,32,3B,A4,DD <1E1B>
1474 DATA 66,03,DD,6E,02,DD,56,01 <1EC5>
1475 DATA DD,5E,00,3E,10,CD,9E,BC <1E7E>
1476 DATA C9,3E,00,C3,45,A4,3E,01 <1EB3>
1477 DATA 32,BD,A4,21,70,01,22,BB <1E3E>
1478 DATA A4,7E,32,B9,A4,FE,00,C8 <1EE5>
1479 DATA 01,20,4E,11,03,00,19,3A <1E61>
1480 DATA BD,A4,FE,01,CA,A5,A4,09 <1E37>
1481 DATA E5,11,09,00,2A,BB,A4,19 <1E8D>
1482 DATA 09,54,5D,E1,ED,B8,01,04 <1E9C>
1483 DATA 00,2A,BB,A4,09,54,5D,21 <1E51>
1484 DATA B3,A4,01,06,00,ED,B0,2A <1EB2>
1485 DATA BB,A4,ED,5B,B9,A4,3A,BD <1EEB>
1486 DATA A4,FE,01,CA,9C,A4,7B,C6 <1E82>
1487 DATA 06,77,5F,19,C3,4B,A4,7B <1E58>
1488 DATA D6,06,77,5F,19,C3,4B,A4 <1ED0>
1489 DATA 23,54,5D,D5,11,06,00,19 <1E7C>
1490 DATA D1,ED,B0,C3,84,A4,01,00 <1E3B>
1491 DATA 00,2C,2C,2C,00,00,FA,01 <1E25>
1492 DATA 01,3E,01,32,D5,A5,C9,3E <1E1E>
1493 DATA 00,32,D5,A5,C9,06,07,CD <1E05>
1494 DATA 5B,A1,FE,00,C8,7B,32,D2 <1E59>
1495 DATA A5,DD,66,03,DD,6E,02,22 <1E9F>
1496 DATA CA,A5,22,CC,A5,DD,66,05 <1E10>
1497 DATA DD,6E,04,22,CE,A5,DD,7E <1E4D>
1498 DATA 06,32,D1,A5,DD,7E,08,32 <1EAB>
1499 DATA D0,A5,DD,7E,0C,32,D3,A5 <1E54>
1500 DATA 21,D0,A5,86,FE,51,D2,BF <1EA0>
1501 DATA A5,DD,7E,0A,21,D1,A5,86 <1E6A>
1502 DATA FE,C9,D2,BF,A5,21,00,C0 <1EC9>
1503 DATA 22,C5,A5,22,C3,A5,3E,01 <1ED9>
1504 DATA 32,C9,A5,DD,7E,0A,3D,F5 <1ED9>
1505 DATA CD,95,A5,F1,FE,00,C2,23 <1EDD>
1506 DATA A5,3A,D3,A5,5F,16,00,19 <1E24>
1507 DATA 22,C5,A5,2A,C3,A5,19,22 <1E83>
1508 DATA C3,A5,3A,D1,A5,4F,ED,5B <1EB3>
1509 DATA CE,A5,3A,D2,A5,FE,01,C2 <1E8F>
1510 DATA 53,A5,ED,5B,CA,A5,D5,CD <1E46>
1511 DATA 95,A5,D1,3A,D0,A5,47,3A <1E73>
1512 DATA D2,A5,FE,01,CA,90,A5,7E <1EFF>
1513 DATA E5,2A,CC,A5,77,23,22,CC <1E8E>
1514 DATA A5,E1,1A,FE,00,C2,7E,A5 <1E90>
```

```
1515 DATA 3A,D5,A5,FE,01,CA,7F,A5 <1E1C>
1516 DATA 1A,77,23,13,78,FE,00,05 <1EC4>
1517 DATA C2,5C,A5,79,FE,00,C8,0D <1ECE>
1518 DATA C3,53,A5,1A,77,C3,7F,A5 <1E20>
1519 DATA 3A,C9,A5,2A,C5,A5,FE,08 <1E4F>
1520 DATA C2,B3,A5,3E,01,32,C9,A5 <1EC8>
1521 DATA 11,50,00,2A,C3,A5,19,22 <1EAD>
1522 DATA C5,A5,22,C3,A5,C9,11,00 <1ED0>
1523 DATA 08,19,22,C5,A5,3C,32,C9 <1E97>
1524 DATA A5,C9,CD,63,A1,C9,00,00 <1E6D>
1525 DATA 00,00,00,00,22,00,00,00 <1E92>
1526 DATA 00,00,00,00,00,00,00,00 <1E94>
1527 DATA 00,16,00,21,FF,A5,01,04 <1E25>
1528 DATA 00,AF,F5,7A,77,09,F1,3C <1ECA>
1529 DATA FE,09,20,F6,CD,F3,A5,14 <1EAA>
1530 DATA 3E,28,BA,C8,18,E5,0E,C1 <1E14>
1531 DATA 21,FF,A5,DF,FC,A5,C9,52 <1E8A>
1532 DATA C6,07,00,00,C1,02,00,00 <1E3E>
1533 DATA C3,02,00,00,C5,02,00,00 <1EF7>
1534 DATA C7,02,00,00,C9,02,00,00 <1E9B>
1535 DATA C2,02,00,00,C4,02,00,00 <1E27>
1536 DATA C6,02,00,00,C8,02,21,29 <1E24>
1537 DATA 96,01,00,00,3E,04,CD,5A <1E9E>
1538 DATA BB,3E,02,CD,5A,BB,7D,FE <1EA8>
1539 DATA C9,C2,3D,A6,7C,FE,97,C8 <1E32>
1540 DATA 7E,FE,80,D2,4B,A6,CD,5A <1EEA>
1541 DATA BB,04,23,C3,33,A6,D6,80 <1ECD>
1542 DATA CD,5A,BB,3E,12,90,47,3E <1EE9>
1543 DATA 20,CD,5A,BB,10,FB,0C,79 <1E3E>
1544 DATA FE,03,CC,65,A6,C3,47,A6 <1E7C>
1545 DATA 0E,00,3E,0A,CD,5A,BB,3E <1E29>
1546 DATA 0D,CD,5A,BB,C9,00       <1877>
```

PINBALLMANIA

Ein rachsüchtiger Professor hat den jungen Pinball-Meister in die Vergangenheit verbannt. Helfen Sie Tiddy, ins Jahr 2111 zurückzukehren. Ein spannendes Flipper-Spiel für alle CPC-Typen.



In einer kleinen, irgendwo in Mitteleuropa gelegenen Grafschaft namens Zobelonien finden alljährlich die großen Pinball-Meisterschaften statt. So auch im Spätsommer des Jahres 2111. Professor Zorkowitsch ist seit 14 Jahren unbestrittener Rekordhalter.
Dieses Mal aber tritt ein junger Student namens Tiddy Tilt gegen ihn an und besiegt ihn schon in der zweiten Runde. Der Professor schwört wütend Rache.
Er konstruiert eine Zeitmaschine und koppelt diese mit dem Pinball-Automaten. Drei Monate später ist es soweit: Er sperrt Tiddy Tilt in seine Zeitmaschine und startet sie. Unter lautem Zischen verschwindet der Student in der Vergangenheit. Als Tiddy wieder erwacht, findet er in der Zeitmaschine einen Brief des Professors vor:
„Mein lieber junger Feind!
Da Ihr nun Pinball-Meister seid, könnt Ihr gleich unter Beweis stellen, ob Ihr diesen Titel auch wirklich verdient. Ihr befindet Euch in einer meiner Erfindungen, dem Zeitzylinder, mit

dem ich Euch in das Jahr Null zurückbefördert habe.
Die Tür des Zeitzylinders wird sich erst wieder öffnen, wenn Ihr es schafft, ins Jahr 2111 zurückzukommen.
Vor Euch steht ein Flipper. Rechts seht Ihr einige Kontrollanzeigen. Die erste von oben zeigt Euch das Jahr an, in dem Ihr Euch befindet. Darunter ist die Zeitkontrolle, die Euch anzeigt, in welches Jahr Ihr nach dem nächsten Zeitsprung gelangt. Diese ‚Warp'-Zeit ist mit dem Flipper verbunden. Je besser Ihr spielt, desto schneller kommt Ihr in der Zeit voran. Darunter seht Ihr noch die Anzeige für den Zeitmultiplikator.
Auf der linken Seite des Flippers habt Ihr, von oben nach unten betrachtet, zunächst die Anzeige für die Energie, die Ihr für die Magnetfelder zum Auffangen des Balles braucht, bevor dieser durch den rechten oder linken Ausgang verschwindet. Diese Taste aber nicht nur einmal drücken, sondern gedrückt halten!
Darunter seht Ihr die Energie, die Euch für den nächsten Zeitsprung zur Verfügung steht. Außerdem wird die Zahl der Bälle angezeigt.
Ganz unten findet Ihr den Multiplikatorbonus.
Und jetzt, mein lieber junger Feind . . . Gut Flipp und auf Nimmerwiedersehen!
Prof. Zorkowitsch"
Dieses Spiel lohnt die Tipparbeit wirklich, Sie werden nicht enttäuscht sein. Es läuft auf allen CPC-Typen mit Kassetten- oder Diskettenlaufwerk und bietet alles, was ein richtiger Flipper auch hat. Es gilt folgende Tastenbelegung:
Schläger links = Z-Taste;
Schläger rechts = Backslash-Taste, links neben der rechten Shift-Taste;
Magnetfeld an = eckige Klammer zu.

Abtipp-Hilfe

Tippen Sie zunächst das Pinball-Vorprogramm ab und speichern Sie es mit
SAVE "PIN"
Das Hauptprogramm legen Sie auf Ihrer Diskette oder Kassette mit dem Befehl
SAVE "PINBALL"
ab. Gestartet wird das Spiel mit
RUN "PIN"
Wir wünschen Ihnen viel Freude beim Flippern.

ME □

```
1 '*****************************  <24B1>
2 '*    PINBALL-VORPROGRAMM    *  <2440>
3 '*          VON              *  <24D0>
4 '*      ERNST KLASAREK       *  <2491>
5 '*          FUER             *  <2424>
6 '*    SCHNEIDER CPC-WELT     *  <24EF>
7 '*     CPC 464/664/6128   ME*  <2483>
8 '*****************************  <24BF>
10 RANDOMIZE INT(TIME/100):CALL &B
C6E:SYMBOL AFTER 64:FOR pk=(HIMEM+
1)TO(HIMEM+768):POKE pk,PEEK(pk)AN
D(PEEK(pk)*2)-2:NEXT                <53B6>
20 MODE 0:INK 0,0:BORDER 0:DEFINT
a-z:w=26:DEF FNr=INT(RND*(w-1+1)+1
):FOR i=1 TO 15:INK i,FNr,FNr:NEXT
:w=15:FOR s=0 TO 40:RESTORE:FOR i=
5 TO 16:READ a$:LOCATE i,8:PEN FNr
:PRINT a$:NEXT i,s:DATA P,I,N,B,A,
L,L,M,A,N,I,A                       <B74A>
30 SYMBOL AFTER 32:PEN FNr:LOCATE
6,11:PRINT"IS ";:PEN FNR:PRINT"LOA
DING":FOR s=0 TO 5000:NEXT:RUN"pin
ball"                               <49AE>
```

```
1 '*****************************  <24B1>
2 '*   PINBALL-HAUPTPROGRAMM   *  <2408>
3 '*          VON              *  <24D0>
4 '*      ERNST KLASAREK       *  <2491>
5 '*          FUER             *  <2424>
6 '*    SCHNEIDER CPC-WELT     *  <24EF>
7 '*     CPC 464/664/6128   ME*  <2483>
8 '*****************************  <24BF>
10 MODE 1:CALL &BC02:INK 0,6:PAPER
 3:BORDER 6:PEN 2                   <1AD5>
11 SYMBOL AFTER 47                  <098D>
12 FOR p=HIMEM+1 TO HIMEM+768:POKE
 p,PEEK(p)AND(PEEK(p)*2):NEXT       <3463>
13 LOCATE 15,5:PRINT"PINBALLMANIA"
                                    <1A98>
14 PEN 1:LOCATE 11,11:PRINT"TOUCHE
S:"                                 <1A66>
15 LOCATE 11,13:PRINT"(Z) = FLIPPE
R LEFT"                             <21C0>
16 LOCATE 11,14:PRINT"(\) = FLIPPE
R RIGHT"                            <221A>
17 LOCATE 11,15:PRINT"(J) = MAGNET
FIELDS ON"                          <2477>
18 DEFSTR a:DEFINT b-o,q-y:ON ERRO
R GOTO 552                          <19A0>
19 PEN 2:RESTORE 530:FOR i=97 TO 1
45:FOR s=0 TO 7:READ a:b(s)=VAL("&
"+a):NEXT                           <420A>
20 SYMBOL i,b(0),b(1),b(2),b(3),b(
4),b(5),b(6),b(7):NEXT              <4CDE>
21 DIM a(14),ce1(25,20),ce2(25,20)
,o1(25),o2(25)                      <3E73>
22 RESTORE 22:FOR i=0 TO 8:READ kt
(i):NEXT:DATA 1,3,5,1,1,1,1,1,1    <360E>
23 a(0)=CHR$(23)+CHR$(1):a(1)=CHR$
(23)+CHR$(0)                        <2E45>
24 a(2)="n":a(3)="o":a(4)="p":av=C
HR$(8):ay=CHR$(10):am=av+ay         <531D>
25 a(5)=CHR$(22)+CHR$(1):a(6)=CHR$
(22)+CHR$(0):ar=CHR$(141)           <3BC4>
26 a(7)=av+av+CHR$(11)+"su"+av+am+
"tu":a(8)="l"+am+"m"                <4B2C>
27 a(9)="c"+am+"d":a(10)=CHR$(134)
:a(11)=CHR$(127)+CHR$(126)          <4127>
28 a(12)=CHR$(128)+CHR$(129)        <1B62>
29 a(13)=CHR$(135)+am+CHR$(136)+CH
R$(137)                             <28DF>
30 a(14)="f"+am+CHR$(138)+av+av+CH
R$(139)                             <318D>
31 alf=a(12)+ay+a(12)+CHR$(9)+CHR$
(9)+a(11)+CHR$(11)+a(11)            <485C>
32 add=" "+am+"A"+am+"D"+am+"D"     <2D46>
33 at=a(10)+a(10)+av+am+a(10)+a(10
):an=" ":ax="X 1000"                <53F7>
34 ak(1)="e"+am+"e"+am+"e"+am+"e"+
am+"e":ak(2)="e"+am+"e"+am+"e"      <5A7C>
35 ak(3)="g"+ay+"g"+ay+"g":ak(4)="
i"+am+ay+"{"                        <429F>
36 ak(5)="c"+am+"d"                 <1B9A>
37 aff1=CHR$(131)+CHR$(130)+ay+CHR
$(131)+CHR$(130)                    <2E68>
38 aff=CHR$(132)+CHR$(133)+CHR$(11
1)+CHR$(132)+CHR$(133)              <2F00>
39 RESTORE 535:GOSUB 504:GOSUB 526
:PRINT a(5)                         <1C1B>
40 DEF FNr=INT(RND*(w-u+1)+u):DEF
FNk1=w1*COS(i):DEF FNk2=w2*SIN(i)   <53B2>
41 q=&BD19:kg(1)=4-lv:kg(2)=kg(1):
wt=0:rt(1)=0:rt(2)=0:bon=0:mt1=0:m
t2=0                                <69E2>
42 pt(1)=0:pt(2)=0:xt1=10:xt2=5:xt
3=5:pl=1:mg=119-(20*lv):ps1=0:ps2=
0:sa=0                              <6CA2>
43 mult=1:ml1=20:mlr=ml1:neu=1:pb(
1)=(1000000*lv):pb(2)=pb(1):bs(7)=
0                                   <64B8>
44 INK 2,20,6:LOCATE 15,15:PRINT"P
LEASE  WAIT":GOSUB 556:EVERY 60,1
GOSUB 551                           <3287>
45 PAPER 2:LOCATE 15,17:PRINT CHR$
(24)+")40 SECONDS<"+CHR$(24)        <2C18>
46 GOSUB 160:INK 3,6,20:GOSUB 160:
SPEED INK 10,20:KEY DEF 66,0,0,0    <295C>
47 vn=40:na=0:st=-1:GOSUB 100:MODE
 1:PAPER 0                          <2935>
48 EVERY 80,0 GOSUB 549:LOCATE 26,
8:PRINT"o":PEN 1                    <1C99>
49 ENV 1,1,122,1,122,-1,10:ENV 2,1
,12,1,12,-1,5:ENV 3,1,12,1,120,-5,
1:ENV 4,17,-1,32:ENT 1,100,-20,5:E
NT 2,1,12,1,120,-1,3:ENT 3,11,10,1
,10,-12,1                           <79D1>
50 WINDOW#1,8,17,1,1:WINDOW#2,23,3
2,1,1:WINDOW#3,8,32,1,25            <2D30>
51 FOR i=1 TO 3:PEN#i,3:PAPER#i,2:
NEXT                                <224D>
52 RESTORE 536:READ x,y,ad:WHILE x
<>-1:LOCATE x,y:PRINT ad:READ x,y,
ad:WEND                             <48B7>
53 RESTORE 53:FOR i=1 TO 5:READ x,
y:LOCATE x,y:PRINT ak(i):NEXT:DATA
 13,4,11,6,28,10,10,22,32,10        <567F>
54 PEN 2:LOCATE 17,14:PRINT CHR$(2
46);:PEN 1:PRINT"0000";:PEN 2:PRIN
T CHR$(247):GOSUB 168               <33BA>
55 PEN 1:PRINT a(5):RESTORE 546:FO
R i=1 TO 4:READ x,y:PEN 2:LOCATE x
,y:PRINT at;:PEN 1:PRINT a(7):NEXT
                                    <4EDA>
56 RESTORE 56:FOR i=1 TO 5:READ x,
y,s:LOCATE x,y:PRINT a(s):NEXT:DAT
A 13,20,13,26,20,14,32,22,9,9,6,9,
28,21,8                             <668D>
57 LOCATE 28,6:PRINT"i";a(8):LOCAT
```

```
E 15,22:PRINT a$:PEN 3:LOCATE 11,
5:PRINT"a"                          (348B)
58 PRINT a(6):PAPER 3:PEN 2:LOCATE
 12,19:PRINT"}":LOCATE 27,19:PRINT
"}"                                 (2BC0)
59 PAPER 0:PRINT a(5)               (10FA)
60 FOR i=0 TO 4 STEP 4              (10C4)
61 LOCATE 34,2+i:PRINT CHR$(144);S
TRING$(4,143);CHR$(142);            (28FC)
62 LOCATE 34,3+i:PRINT ar;"    ";a
r;                                  (2414)
63 LOCATE 34,4+i:PRINT CHR$(145);S
TRING$(4,143);CHR$(140);            (28DE)
64 NEXT                             (0643)
65 FOR i=2 TO 24                    (0FC5)
66 LOCATE 1,i:PRINT ar:LOCATE 7,i:
PRINT ar                            (2299)
67 LOCATE 33,i:PRINT ar:LOCATE 40,
i:PRINT ar;                         (25ED)
68 NEXT                             (0643)
69 RESTORE 538:FOR i=1 TO 20:READ
l1,l2,w,x,y,z                       (3526)
70 LOCATE l1,l2:PRINT CHR$(w);STRI
NG$(x,y);CHR$(z);                   (33D6)
71 NEXT                             (0651)
72 RESTORE 547:FOR i=1 TO 5:READ x
,y,z:LOCATE x,y:PRINT ar;STRING$(z
," ");ar:NEXT                       (4A89)
73 RESTORE 548:PEN 3:FOR i=1 TO 11
:READ x,y:LOCATE x,y:PRINT a(10):N
EXT:LOCATE 1,1:PRINT STRING$(40,20
8);:LOCATE 1,25:PRINT STRING$(40,2
10);:GOSUB 427                      (61D7)
74 FOR i=1 TO 25                    (0FBB)
75 LOCATE 1,i:PRINT CHR$(211):LOCA
TE 40,i:PRINT CHR$(209);            (260E)
76 NEXT:LOCATE 1,1:PRINT a(6)       (1498)
77 PEN 2:FOR i=11 TO 20:LOCATE 34,
i:PRINT ar;"    ";ar;:NEXT          (3380)
78 RESTORE 558:FOR i=0 TO 6:READ x
(i),y(i),ab(i):NEXT                 (3840)
79 LOCATE 26,7:PRINT"q":PEN 0:PAPE
R 2:LOCATE 35,11:PRINT"w01x"        (2433)
80 FOR i=12 TO 20:LOCATE 35,i:PRIN
T"wxwx":NEXT                        (2339)
81 PAPER 0:PEN 2:LOCATE 34,10:PRIN
T CHR$(144);STRING$(4,143);CHR$(14
2);                                 (2A26)
82 LOCATE 34,21:PRINT CHR$(145);ST
RING$(4,143);CHR$(140);             (24B5)
83 RESTORE 557:PEN 0:PAPER 3:FOR i
=1 TO 14:READ x,y,a:LOCATE x,y:PRI
NT a;:NEXT:LOCATE 2,19:PRINT add:I
F bs(7)=1 THEN RETURN               (5A9A)
84 RESTORE 84:PAPER 2:FOR i=12 TO
18:READ a:LOCATE 2,i:PRINT a:NEXT:
ORIGIN 302,282:RESTORE 540:DATA "}
",B,A,L,L,S,")"                     (5095)
85 DEG:WHILE z<>1:READ z,x,y,w1,w2
,s:FOR i=x TO y STEP s:PLOT FNk1,F
Nk2,3:i=i+y+ABS(x):PLOT FNk1,FNk2:
i=i-(y+ABS(x)):NEXT:WEND:READ o1,o
2:IF o1=0 THEN 86 ELSE ORIGIN o1,o
2:z=0:GOTO 85                       (CEEA)
86 ORIGIN 1,1:RESTORE 541:x=0       (15AF)
87 READ x,y                         (0F8D)
88 IF x=0 THEN READ x,y:MOVE x,y:G
OTO 87                              (2760)
89 IF x=-1 THEN 92                  (1107)
90 DRAW x,y                         (0FF9)
91 GOTO 87                          (095F)
92 CLS#1:CLS#2:INK 1,24:INK 2,20:I
NK 3,6:z=REMAIN(0):BORDER 0:INK 0,
0                                   (3013)
93 PAPER 0:PEN 1:PRINT a(6):LOCATE
 31,6:PRINT a(8):LOCATE 30,21:PRIN
T" "+am+" "                         (3810)
94 LOCATE 31,20:PRINT" "+am+" ":LO
CATE 32,6:PRINT" ":RESTORE 94:PAPE
R 3:PEN 0:FOR i=6 TO 9:READ x:LOCA
TE 32,i:PRINT CHR$(x):NEXT:PEN 1:P
APER 0:DATA 84,73,77,69             (69F4)
95 MOVE 464,48:DRAW 494,62:MOVE 46
4,80:DRAW 494,94                    (2422)
96 MOVE 494,312:DRAW 494,222:DRAW
512,222:DRAW 512,326                (2649)
97 z=REMAIN(1):vn=0:na=40:st=1:GOS
UB 100:KEY DEF 66,1,252,252:SPEED
INK 1,1                             (40DB)
98 RANDOMIZE INT(TIME/100):PLOT 64
0,0,2:GOTO 135                      (1DD9)
99 :                                (06CA)
100 FOR i=vn TO na STEP st:OUT &BC
00,1:OUT &BD00,i:SOUND 7,i+10*10,6
,15,4,3:NEXT:RETURN                 (4891)
101 :                               (06CE)
102 PEN 2:PAPER 0:FOR i=vn TO na S
TEP st:LOCATE f,i:CALL q:PRINT"j":
GOSUB 150:LOCATE f,i:PRINT" ":NEXT
:RETURN                             (51A7)
103 :                               (06D2)
104 ORIGIN 1,1:TAG:FOR i=vn TO na
STEP st:MOVE f,i:CALL q:PRINT"j";:
MOVE f,i:CALL q:PRINT an;:NEXT:TAG
OFF:an=" ":RETURN                   (63F2)
105 :                               (06D6)
106 ORIGIN 1,1:LOCATE 1,1:PRINT a(
0):TAG                              (19E9)
107 FOR i=vn TO na STEP st:f=f+w1    (2CCA)
108 MOVE f,i:CALL q:PRINT"j";:MOVE
 f,i:CALL q:PRINT"j";:NEXT          (34DF)
109 TAGOFF:PRINT a(1):RETURN        (1169)
110 :                               (06E0)
111 PRINT a(0):ORIGIN o1(v),o2(v):
TAG:t=0                             (2F30)
112 WHILE ce1(v,t)<>-1:MOVE ce1(v,
t),ce2(v,t):CALL q:PRINT"j";:MOVE
ce1(v,t),ce2(v,t):CALL q:PRINT"j";
:t=t+1:WEND:TAGOFF:PRINT a(1):ORIG
IN 1,1:RETURN                       (9C77)
113 :                               (06E6)
114 PEN 1:PAPER 3:LOCATE 34,9:PRIN
T"yWARPy"                           (1A8D)
115 PAPER 0:l1=26:l2=7:GOSUB 502:l
2=6:GOSUB 502:LOCATE 26,7:PRINT"q"
:PRINT a(5):PEN 1:LOCATE 26,6:PRIN
T"i";:PEN 2:PRINT au+"j":PRINT a(6
):SOUND 7,0,5,15,4,3,10             (7568)
116 w=100:u=70:m=FNr:t=-m*2:f=2:w=
10:u=1:PEN 2:EVERY 70,0 GOSUB 131   (4FF7)
117 FOR x=m TO 0 STEP-1:t=t+4:pt(p
l)=pt(pl)+100:GOSUB 466             (427C)
118 FOR i=2 TO 4                    (0E40)
119 LOCATE 26,8:PRINT a(i):FOR s=0
 TO t:NEXT                          (253D)
120 NEXT:SOUND 1,ABS(t)*2,2,15,4,3
                                    (1DC6)
121 LOCATE 26,8:PRINT a(3):FOR s=0
 TO t-50:NEXT                       (2547)
122 IF f=2 THEN f=3 ELSE f=2        (1BEA)
123 PEN f                           (0A1F)
124 NEXT x                          (0AA4)
125 z=REMAIN(0):GOSUB 173:sc=50:GO
SUB 459:mg=mg+20:GOSUB 170:PEN 2:L
OCATE 26,8:PRINT"p":PEN 1:kt(8)=1:
LOCATE 31,15:PRINT"g":GOSUB 550     (634B)
126 l1=26:l2=6:GOSUB 375:IF flag>6
 THEN 508                           (293D)
127 PEN 2:LOCATE 26,7:PRINT"j"      (12A9)
128 GOSUB 151                       (098B)
129 LOCATE 26,7:PRINT"q"            (0FE2)
130 RESTORE 130:GOSUB 284:LOCATE 2
6,8:PRINT"o":w1=56:st=2:GOTO 258:D
ATA 250,80,-14,390,-9               (43C9)
131 wt=wt+1:GOSUB 174               (17E4)
132 flag=FNr:LOCATE 26,7:IF flag>6
 AND x<40 THEN PEN 2:PRINT"y":SOUN
D 4,100,2,15,4,3,1 ELSE PEN 3:PRIN
T"q":SOUND 4,10,2,7                 (57B4)
133 LOCATE 26,8:PEN f:RETURN        (12B6)
134 :                               (0611)
135 u=1:w=5:vn=18:na=22:st=1:f=32:
GOSUB 102:bs(7)=1                   (44B9)
136 PAPER 0:LOCATE 32,22:PRINT"j"+
am;:PEN 1:PRINT"c":GOSUB 161        (27E5)
137 IF lo=4 THEN lo=0:GOSUB 329:da
=0:bon=0:GOSUB 151:GOSUB 167:mult=
1:GOSUB 372:GOSUB 151:GOTO 144      (4EFA)
138 IF neu=0 THEN d=6:GOSUB 559:GO
SUB 157:mult=mult+1:sc=20:GOSUB 46
0:sc=15:GOSUB 182:GOSUB 185:GOSUB
372:m=30:mg=mg+25:GOSUB 159:GOSUB
189:GOSUB 186:sc=5:GOSUB 467:GOTO
144                                 (8F2A)
139 PRINT a(1):GOSUB 403            (124C)
140 IF lo=2 OR lo=3 THEN IF mg>0 T
HEN GOSUB 405:GOSUB 406             (2887)
141 PEN 2:LOCATE 14,2:PRINT CHR$(2
4)+" PRESS ENTER "+CHR$(24)         (2CD4)
142 WHILE INKEY$<>"":WEND           (0D99)
143 WHILE INKEY$=CHR$(13):WEND      (11B8)
144 PAPER 0:LOCATE 14,2:PRINT SPC(
13):GOSUB 153:GOSUB 414:GOSUB 154:
PEN 1:LOCATE 32,22:PRINT a(9):vn=2
1:na=14:st=-1:f=32:GOSUB 102        (5ED2)
145 l1=32:l2=13:GOSUB 374:GOSUB 16
9:LOCATE 17,16:PRINT SPC(6)         (2DDB)
146 GOSUB 159:IF kt(8)=1 THEN LOCA
TE 31,15:PRINT" ":GOSUB 154:kt(8)=
0:GOSUB 159                         (3639)
147 IF neu=1 THEN neu=0:mg=119-(20
*lv):GOSUB 173:GOSUB 170:GOSUB 176
:GOSUB 179:GOSUB 181:GOSUB 393:GOS
UB 154                              (4E07)
148 l1=32:l2=13:GOSUB 375:GOSUB 56
4                                   (20AB)
149 PEN 1:LOCATE 31,15:PRINT"g":kt
(8)=1:GOSUB 154:GOTO 190            (283C)
150 FOR s=0 TO 80:NEXT:RETURN       (13C2)
151 FOR s=0 TO 150:NEXT:RETURN      (1352)
152 FOR s=100 TO 1500 STEP 50:SOUN
D 7,s,10,7,,,INT(s/100):NEXT:RETUR
N                                   (3374)
153 FOR s=1500 TO 100 STEP-100:SOU
ND 7,s,3,5,,1,INT(s/100):NEXT:RETU
RN                                  (3421)
154 FOR s=300 TO 3000 STEP 300:SOU
ND 7,s,1,7,,,INT(s/100):NEXT:RETUR
N                                   (348F)
155 FOR s=1 TO 29:SOUND 7,0,18,15,
,,s:NEXT:SOUND 7,0,300,15,,,30:SOU
ND 7,0,150,7,,2,30:GOSUB 160:INK 0
,0:BORDER 0:RETURN                  (53B2)
156 FOR i=600 TO 1 STEP-1:SOUND 7,
i,1,15,2,2:CALL &BD20:CALL &BD21:C
ALL &BD22:CALL &BD23:CALL &BD24:CA
LL &BD25:NEXT:RETURN                (4649)
157 w=26:u=1:FOR s=0 TO 6:FOR i=1
TO 3:INK i,26:FOR loop=2 TO 30 STE
P 2:SOUND 7,loop,1,4,,,loop:NEXT l
oop:INK i,0:NEXT i:BORDER FNr,FNr:
NEXT s:BORDER 0:INK 1,24:INK 2,20:
INK 3,6:GOSUB 159:RETURN            (9EFD)
158 FOR s=2 TO 80:SOUND 7,s,3,15,4
,3:NEXT:RETURN                      (2426)
159 FOR s=0 TO 2000:NEXT:RETURN     (1425)
160 FOR s=0 TO 5000:NEXT:RETURN     (1402)
161 SOUND 7,0,150,7,1,,1:RETURN     (1589)
162 SOUND 7,0,8,12,4,3,20:RETURN    (177E)
163 FOR s=100 TO 5 STEP-5:SOUND 7,
s,1,7:NEXT:RETURN                   (220A)
164 FOR s=10 TO 50:SOUND 7,s*5,10,
7,3,1:NEXT:RETURN                   (27DC)
165 FOR i=0 TO 4:FOR s=80 TO 10 ST
EP-10:SOUND 7,s,2,7,3,1:NEXT s,i:R
ETURN                               (3B8E)
166 FOR i=0 TO 4:FOR s=60 TO 10 ST
EP-10:SOUND 7,s,2,7,3,2:NEXT s,i:R
ETURN                               (3B39)
167 w=10:u=1:FOR s=10 TO 300 STEP
10:SOUND 7,s*FNr,3,5:NEXT:RETURN    (39A7)
168 IF xt1>99 THEN RETURN ELSE PEN
 2:LOCATE 24,4:PRINT USING"##";xt1
:RETURN                             (2B48)
169 anz="yREALy":x=34:y=5:GOSUB 18
7:PEN 1:LOCATE 35,3:PRINT USING"##
##";rt(pl):RETURN                   (490D)
170 anz="yMGTy":x=2:y=7:GOSUB 187   (2615)
171 IF mg>999 THEN mg=999           (1925)
172 PEN 1:LOCATE 3,5:PRINT USING"#
##";mg:RETURN                       (1C83)
173 anz="yWARPy":x=34:y=9:GOSUB 18
7                                   (285F)
174 PEN 1:IF wt>2111 THEN wt=2111:
PEN 3                               (1FFA)
175 LOCATE 35,7:PRINT USING"####";
wt:RETURN                           (1B77)
176 anz="yERGy":x=2:y=11:GOSUB 187
```

```
                                        (27CB)
177 IF eg>999 THEN PEN 3:LOCATE 3,
9:PRINT"E99":RETURN                     (2027)
178 PEN 1:LOCATE 3,9:PRINT USING"#
##";eg:RETURN                           (1C6C)
179 anz=" 1y ":x=3:y=15:GOSUB 187:
PEN 1:LOCATE 4,13:PRINT USING"##";
kg(1)                                   (3F4A)
180 IF sp=1 THEN RETURN ELSE anz="
 2y ":x=3:y=19:GOSUB 187:PEN 1:LOC
ATE 4,17:PRINT USING"##";kg(2):RET
URN                                     (4D2F)
181 anz=add:x=2:y=19:GOSUB 187:PEN
 1:LOCATE 4,21:PRINT USING"##";da:
RETURN                                  (3EE7)
182 IF da=99 THEN RETURN ELSE PAPE
R 3:PEN 1:LOCATE 2,19:PRINT add         (2591)
183 FOR i=1 TO sc:da=da+1:SOUND 7,
100,10,7,,2:IF da>99 THEN da=99         (4173)
184 LOCATE 4,21:PRINT USING"##";da
:NEXT:GOSUB 181:GOSUB 151:GOSUB 15
1:PAPER 0:RETURN                        (2D4E)
185 BORDER 3,6:PEN 1:PAPER 3:LOCAT
E 34,22:PRINT"yMULTy";:SOUND 7,30,
70,7,1:GOSUB 151:GOSUB 151:PAPER 0
:PEN 2:BORDER 0:RETURN                  (439B)
186 PEN 0:PAPER 3:LOCATE 34,22:PRI
NT"yMULTy";:PAPER 0:PEN 2:RETURN        (2416)
187 PEN 2:SOUND 4,f+100,15,6,2,2:S
OUND 1,f*7,70,6,2,2:FOR s=0 TO 12:
PAPER 1:LOCATE x,y:PRINT anz:PAPER
 3:LOCATE x,y:PRINT anz:NEXT:PEN 0
:PAPER 3:LOCATE x,y:PRINT anz:PAPE
R 0:PEN 2:GOSUB 151:RETURN              (8D33)
188 GOSUB 173:GOSUB 170:GOSUB 176:
RETURN                                  (15D7)
189 eg=eg+m:wt=wt+m:GOTO 188            (2B90)
190 :                                   (0681)
191 l1=29:l2=18:GOSUB 322:w=10:u=1
:ORIGIN 1,1                             (2F2F)
192 ON FNr GOTO 193,194,195,196,19
7,198,196,197,198,198                   (33B7)
193 v=17:GOSUB 111:GOTO 233             (162B)
194 v=18:GOSUB 111:GOTO 239             (166D)
195 v=19:GOSUB 111:f=240:GOTO 256       (1E74)
196 RESTORE 196:GOTO 199:DATA 140,
262,12,440,-10                          (234D)
197 RESTORE 197:GOTO 199:DATA 140,
260,11,440,-11                          (232B)
198 RESTORE 198:DATA 140,266,11,44
0,-13                                   (1E07)
199 READ vn,na,st,f,w1:GOSUB 106        (27CC)
200 :                                   (0695)
201 IF w1<>-10 THEN 207 ELSE l1=19
:l2=8:GOSUB 322:w=6:u=1:ORIGIN 1,1
                                        (3DC3)
202 ON FNr GOTO 203,204,205,206,20
3,203                                   (23E0)
203 vn=254:na=120:st=-12:f=308:w1=
11:GOSUB 106:GOTO 191                   (3CAD)
204 vn=254:na=106:st=-14:f=308:w1=
10:GOSUB 106:lo=3:GOTO 484              (4477)
205 vn=254:na=106:st=-16:f=308:w1=
12:GOSUB 106:lo=4:GOTO 484              (4411)
206 vn=254:na=90:st=-20:f=308:w1=8
:GOSUB 106:GOTO 249                     (3B66)
207 IF w1<>-11 THEN 213                 (133E)
208 l1=19:l2=8:GOSUB 322:w=20           (2280)
209 IF FNr=2 THEN 211 ELSE w=2:ON
FNr GOTO 210,212                        (28DC)
210 vn=254:na=70:st=-12:f=294:w1=3
:GOSUB 106:GOTO 285                     (3B42)
211 vn=250:na=40:st=-18:f=296:an="
j":lo=0:PRINT a(0):GOSUB 104:GOTO
323                                     (4E04)
212 vn=254:na=70:st=-12:f=294:w1=-
3:GOSUB 106:f=256:GOTO 256              (450C)
213 w=10:u=1:l1=19:l2=8:GOSUB 322       (2905)
214 ON FNr GOTO 215,216,217,218,21
9,220,215,217,215,217                   (3396)
215 vn=254:na=200:st=-9:f=296:w1=-
17:GOSUB 106:GOTO 222                   (3C50)
216 vn=254:na=136:st=-10:f=296:w1=
-13:GOSUB 106:GOTO 475                  (3D55)
217 vn=254:na=130:st=-11:f=296:w1=
-11:GOSUB 106:GOTO 233                  (3DB6)
218 vn=254:na=96:st=-15:f=296:w1=-
8:GOSUB 106:GOTO 239                    (3C02)
219 vn=254:na=106:st=-11:f=296:w1=
-9:GOSUB 106:lo=1:GOTO 484              (449B)
220 vn=254:na=106:st=-11:f=296:w1=
-9:GOSUB 106:lo=2:GOTO 484              (441E)
221 :                                   (06BF)
222 l1=9:l2=13:GOSUB 322:w=10:u=1:
ORIGIN 1,1                              (2E8E)
223 ON FNr GOTO 224,225,226,230,22
8,229,231,227,231,227                   (3319)
224 vn=186:na=90:st=-15:f=154:w1=3
0:GOSUB 106:GOTO 249                    (3B33)
225 vn=186:na=106:st=-6:f=154:w1=1
8:GOSUB 106:lo=3:GOTO 484               (42BE)
226 vn=186:na=110:st=-6:f=154:w1=1
9:GOSUB 106:lo=4:GOTO 484               (4240)
227 vn=190:na=120:st=-7:f=154:w1=2
8:GOSUB 106:GOTO 191                    (3AD0)
228 vn=190:na=220:st=4:f=154:w1=28
:GOSUB 106:ks=3:GOTO 445                (415D)
229 vn=196:na=230:st=6:f=154:w1=31
:GOSUB 106:ks=2:GOTO 445                (41D9)
230 vn=196:na=244:st=5:f=154:w1=20
:GOSUB 106:ks=1:GOTO 445                (41ED)
231 vn=210:na=270:st=13:f=154:w1=1
9:GOSUB 106:w1=-13:GOTO 213             (4553)
232 :                                   (06D5)
233 l1=9:l2=18:GOSUB 322:w=8:u=1:O
RIGIN 1,1                               (2D86)
234 ON FNr GOTO 235,236,237,238,23
5,236,235,236                           (2B51)
235 vn=150:na=270:st=14:f=154:w1=1
2:GOSUB 106:w1=-13:GOTO 213             (4576)
236 vn=150:na=260:st=14:f=154:w1=1
4:GOSUB 106:w1=-11:GOTO 207             (45AB)
237 vn=134:na=280:st=14:f=158:w1=1
6:GOSUB 106:GOTO 355                    (3BC1)
238 vn=134:na=270:st=14:f=154:w1=2
4:GOSUB 106:GOTO 114                    (3B4E)
239 :                                   (06E3)
240 x=13:y=20:s=4:z=13:GOSUB 248:w
=10:u=1                                 (37B0)
241 IF FNr>5 THEN w=5:ON FNr GOTO
243,244,245,246,247                     (2F70)
242 w=10:v=0:GOSUB 111:GOTO 249         (1DE0)
243 v=1:GOSUB 111:GOTO 191              (1586)
244 v=2:GOSUB 111:lo=3:GOTO 484         (1DCA)
245 v=3:GOSUB 111:lo=4:GOTO 484         (1D35)
246 v=4:GOSUB 111:f=360:GOTO 286        (1E4C)
247 v=5:GOSUB 111:f=340:GOTO 286        (1E72)
248 PEN 3:LOCATE x,y:PRINT a(z):BO
RDER FNr:sc=1:GOSUB 458:BORDER 0:S
OUND s,2000,60,7,3:PEN 1:LOCATE x,
y:PRINT a(z):RETURN                     (6219)
249 x=26:y=20:s=1:z=14:GOSUB 248:w
=10:u=1                                 (37B3)
250 IF FNr>5 THEN w=4:ON FNr GOTO
252,253,254,255                         (2B23)
251 w=10:v=6:GOSUB 111:GOTO 240         (1D2C)
252 v=7:GOSUB 111:lo=1:GOTO 484         (1D0A)
253 v=8:GOSUB 111:lo=2:GOTO 484         (1D9F)
254 v=9:GOSUB 111:f=240:GOTO 257        (1D1B)
255 v=10:GOSUB 111:GOTO 233             (1638)
256 :                                   (0625)
257 ORIGIN 1,1:w=3:u=2:st=FNr:w1=7
2:IF f>250 THEN w1=62                   (3EED)
258 TAGOFF:LOCATE 1,1:PRINT a(0):T
AG:FOR i=f TO 286 STEP st:w1=w1-1:
MOVE i,w1:PRINT"j";:MOVE i,w1:PRIN
T"j";:IF INKEY(71)=0 THEN 260           (6B41)
259 SOUND 4,w1,1,4:NEXT:TAGOFF:GOT
O 323                                   (1AB2)
260 TAGOFF:LOCATE 1,1:PRINT a(1):P
EN 1:LOCATE 15,22:PRINT alt:GOSUB
 162:LOCATE 15,22:PRINT alt             (3B11)
261 w=6:u=1                             (1274)
262 IF i<260 THEN ON FNr GOTO 263,
264,265,266,267,264 ELSE 268            (32AC)
263 RESTORE 263:GOSUB 284:GOTO 348
:DATA 80,280,14,230,0                   (253B)
264 RESTORE 264:GOSUB 284:GOTO 361
:DATA 80,270,14,240,1                   (2580)
265 RESTORE 265:GOSUB 284:w1=-11:G
OTO 207:DATA 80,270,14,240,4            (2F5E)
266 RESTORE 266:GOSUB 284:GOTO 355
:DATA 80,270,14,240,6                   (25D7)
267 RESTORE 267:GOSUB 284:GOTO 347
:DATA 80,270,14,240,8                   (255B)
268 IF i<270 THEN w=9:ON FNr GOTO
269,270,272,271,272,273,274,274,26
9 ELSE 275                              (4578)
269 RESTORE 269:GOSUB 284:GOTO 361
:DATA 80,270,14,254,1                   (2594)
270 RESTORE 270:GOSUB 284:w1=-13:G
OTO 213:DATA 80,270,14,254,2            (2FB2)
271 RESTORE 271:GOSUB 284:w1=-10:G
OTO 201:DATA 80,270,14,254,4            (2FEE)
272 RESTORE 272:GOSUB 284:GOTO 355
:DATA 80,270,14,254,5                   (25F9)
273 RESTORE 273:GOSUB 284:GOTO 347
:DATA 80,270,14,254,7                   (25A3)
274 RESTORE 274:GOSUB 284:GOTO 114
:DATA 80,270,14,254,10                  (2637)
275 w=10:ON FNr GOTO 277,282,276,2
78,278,276,279,280,281,282              (3BB6)
276 RESTORE 276:GOSUB 284:w1=-13:G
OTO 213:DATA 70,260,14,270,1            (2F69)
277 RESTORE 277:GOSUB 284:w1=-11:G
OTO 207:DATA 70,260,14,270,2            (2F85)
278 RESTORE 278:GOSUB 284:GOTO 347
:DATA 70,270,14,270,6                   (252B)
279 RESTORE 279:GOSUB 284:ks=1:GOT
O 445:DATA 70,250,15,270,12             (2EA1)
280 RESTORE 280:GOSUB 284:ks=2:GOT
O 445:DATA 70,234,14,270,14             (2E8C)
281 RESTORE 281:GOSUB 284:ks=3:GOT
O 445:DATA 70,224,14,270,16             (2E85)
282 RESTORE 282:GOSUB 284:GOTO 380
:DATA 70,180,14,270,22                  (26A5)
283 :                                   (063B)
284 READ vn,na,st,f,w1:GOSUB 105:R
ETURN                                   (2941)
285 :                                   (063F)
286 ORIGIN 1,1:w=3:u=2:st=FNr:w1=6
4:IF f>345 THEN w1=74                   (3F7A)
287 TAGOFF:LOCATE 1,1:PRINT a(0):T
AG:FOR i=f TO 306 STEP-st:w1=w1-1:
MOVE i,w1:PRINT"j";:MOVE i,w1:PRIN
T"j";:IF INKEY(22)=0 THEN 289           (6CEB)
288 SOUND 1,w1,1,4:NEXT:TAGOFF:GOT
O 323                                   (1A84)
289 TAGOFF:LOCATE 1,1:PRINT a(1):P
EN 1:LOCATE 21,23:PRINT attr:GOSUB
 162:LOCATE 15,22:PRINT alt             (3BB0)
290 w=8:u=1                             (12B2)
291 IF i>340 THEN ON FNr GOTO 292,
293,295,294,295,296,297,295 ELSE 2
98                                      (3ABD)
292 RESTORE 292:GOSUB 320:GOTO 347
:DATA 80,280,18,360,0                   (25D4)
293 RESTORE 293:GOSUB 320:GOTO 355
:DATA 80,260,14,346,-1                  (2687)
294 RESTORE 294:GOSUB 320:w1=-13:G
OTO 213:DATA 80,270,14,346,-5           (30D2)
295 RESTORE 295:GOSUB 320:GOTO 361
:DATA 80,280,14,346,-6                  (26A4)
296 RESTORE 296:GOSUB 320:GOTO 348
:DATA 80,280,14,346,-8                  (266F)
297 RESTORE 297:GOSUB 320:GOTO 416
:DATA 80,270,14,346,-11                 (2777)
298 w=11:u=1                            (13C7)
299 IF i>326 THEN ON FNr GOTO 300,
301,302,303,304,305,307,307,300,30
6,306 ELSE 308                          (4626)
300 RESTORE 300:GOSUB 320:GOTO 355
:DATA 70,260,14,332,0                   (253B)
301 RESTORE 301:GOSUB 320:w1=-10:G
OTO 201:DATA 70,260,14,332,-2           (3040)
302 RESTORE 302:GOSUB 320:w1=-11:G
OTO 207:DATA 70,260,14,332,-3           (3064)
303 RESTORE 303:GOSUB 320:w1=-13:G
OTO 213:DATA 70,260,14,332,-4           (301A)
304 RESTORE 304:GOSUB 320:GOTO 361
:DATA 70,280,14,332,-5                  (2631)
305 RESTORE 305:GOSUB 320:GOTO 348
:DATA 70,280,14,332,-7                  (2689)
306 RESTORE 306:GOSUB 320:GOTO 416
:DATA 70,270,14,332,-9                  (2695)
307 RESTORE 307:GOSUB 320:GOTO 437
:DATA 70,270,14,332,-11                 (2714)
```

```
308 w=15:u=1                                    (13E3)
309 ON FNr GOTO 310,318,311,312,31
3,314,315,316,317,312,318,314,315,
314,316                                         (47B2)
310 RESTORE 310:GOSUB 320:w1=-10:G
OTO 201:DATA 70,260,14,320,0                    (2F32)
311 RESTORE 311:GOSUB 320:w1=-13:G
OTO 213:DATA 70,260,14,320,-3                   (30E5)
312 RESTORE 312:GOSUB 320:GOTO 361
:DATA 70,280,14,320,-4                          (26C5)
313 RESTORE 313:GOSUB 320:GOTO 348
:DATA 70,280,14,320,-6                          (260B)
314 RESTORE 314:GOSUB 320:GOTO 416
:DATA 70,270,14,320,-8                          (266B)
315 RESTORE 315:GOSUB 320:GOTO 437
:DATA 70,262,14,320,-10                         (27CF)
316 RESTORE 316:GOSUB 320:GOTO 377
:DATA 70,232,14,320,-14                         (275C)
317 RESTORE 317:GOSUB 320:GOTO 221
:DATA 70,196,14,320,-16                         (2756)
318 RESTORE 318:GOSUB 320:GOTO 469
:DATA 70,190,14,320,-20                         (27FB)
319 :                                           (0683)
320 READ vn,na,st,t,w1:GOSUB 106:R
ETURN                                           (29A9)
321 :                                           (0687)
322 PRINT a(5):PEN 3:LOCATE 11,12:
PRINT at;:PEN 2:PRINT a(7):SOUND 4
,t+100,7,15,4,3:SOUND 1,t+10,3,12,
4,1,3:sc=10:GOSUB 465:PEN 2:LOCATE
 11,12:PRINT at;:PEN 1:PRINT a(7):
PRINT a(6):RETURN                               (98CA)
323 :                                           (068B)
324 LOCATE 1,1:PRINT a(1):TAG:FOR
i=50 TO 22 STEP-2:SOUND 7,(1000-(i
*10)),10,15,4,3:MOVE 296,i:CALL q:
PRINT"j";:NEXT:MOVE 296,i:PRINT" "
;:TAGOFF:lo=0:GOSUB 158                          (771A)
325 GOSUB 159:IF sa=1 THEN neu=2:l
o=0:GOSUB 413:sa=0:GOTO 135 ELSE k
g(pl)=kg(pl)-1:xt1=10:xt2=5:xt3=5:
GOSUB 179:mt1=0:mt2=0:GOSUB 329:GO
SUB 168:GOSUB 83:GOSUB 563:GOTO 33
5                                               (9E9B)
326 mult=1:mll=20:ps1=0:ps2=0:mlr=
20:RESTORE 22:FOR i=0 TO 8:READ kt
(i):NEXT                                        (528E)
327 PAPER 0:PEN 1:RESTORE 545:READ
 11,12,ad:WHILE 11<>-1:GOSUB 154:L
OCATE 11,12:PRINT ad:GOSUB 151:REA
D 11,12,ad:WEND:LOCATE 13,4:PRINT
ak(1):LOCATE 11,6:PRINT ak(2):LOCA
TE 28,10:PRINT ak(3)                            (9090)
328 PEN 0:PAPER 2:FOR s=20 TO 12 S
TEP-1:LOCATE 35,s:PRINT"wxwx":NEXT
:LOCATE 35,11:PRINT"w01x":wt=rt(pl
):PAPER 0:bon=0:da=0:eg=0:GOTO 135
                                                (6F67)
329 IF da=0 THEN 333 ELSE IF pl=1
THEN zb=2 ELSE zb=1                             (2CE7)
330 GOSUB 159:PAPER#zb,3:PEN#zb,2:
CLS#zb:anz=add:x=2:y=19:da=da*100:
FOR loop=1 TO mult:SOUND 4,2*loop,
20,15,4,3:LOCATE#zb,1,1:PRINT#zb,l
oop;av+"X";USING"######";da*loop;:
GOSUB 187:GOSUB 151:NEXT:GOSUB 159
:GOSUB 163:CLS#zb                               (D888)
331 z=da*mult:FOR p=1 TO mult:SOUN
D 7,p*20,25,15,4,3:z=z-da:pt(pl)=p
t(pl)+da:GOSUB 466:PRINT#zb,z:FOR
s=0 TO 1000:NEXT s,p:PAPER#zb,2:PE
N#zb,3:CLS#zb:GOSUB 159:da=0:GOSUB
 181                                            (C3B6)
332 IF pt(zb)>0 THEN LOCATE#zb,2,1
:PRINT#zb,USING"########";pt(zb);               (42C8)
333 PAPER 3:PEN 0:LOCATE 2,19:PRIN
T add:PAPER 0:IF bon=0 THEN RETURN
                                                (2705)
334 GOSUB 151:sc=1:GOSUB 159:GOSUB
 427:PEN 1:FOR n=bon TO 1 STEP-1:L
OCATE 18,14:PRINT USING"####";n:GO
SUB 461:SOUND 7,n+10,1,7:FOR s=0 T
O INT(n/2)STEP 2:NEXT s,n:LOCATE 1
8,14:PRINT"0000":RETURN                         (8A21)
335 IF pt(pl)>=pb(pl)THEN pb(pl)=p
b(pl)+1000000:kg(pl)=kg(pl)+1:INK
0,0,20:BORDER 20,0:GOSUB 156:GOSUB
 155:GOSUB 167:GOSUB 408:GOSUB 159
:GOSUB 179:INK 1,24:INK 3,6:INK 0,
0:BORDER 0:GOTO 335                             (A0EC)
336 neu=1:IF sp=2 THEN 338                      (1AAD)
337 IF kg(1)<1 THEN GOSUB 407:GOSU
B 396:GOTO 345 ELSE 326                         (24CF)
338 IF pl=1 AND kg(1)<1 THEN GOSUB
 407:GOSUB 396                                  (2208)
339 IF pl=2 AND kg(2)<1 THEN GOSUB
 407:GOSUB 396                                  (22FE)
340 IF kg(1)<1 AND kg(2)<1 THEN 34
5                                               (1F16)
341 IF pl=1 AND kg(2)>0 THEN pl=2:
GOTO 326                                        (2555)
342 IF pl=1 AND kg(2)<1 THEN 326                (1C20)
343 IF pl=2 AND kg(1)>0 THEN pl=1:
GOTO 326                                        (25BA)
344 IF pl=2 AND kg(1)<1 THEN 326                (1C14)
345 PAPER 3:CLS:PEN 2:LOCATE 16,13
:PRINT"GAME OVER":GOSUB 158:GOSUB
152:GOSUB 160:PRINT STRING$(25,10)
:GOSUB 161:RUN                                  (4140)
346 :                                           (06B9)
347 v=24:GOTO 349                               (1137)
348 v=25                                        (0C9F)
349 GOSUB 549:LOCATE 1,1:PRINT a(0
):TAG:ORIGIN o1(v),o2(v):t=0:u=1:w
=7                                              (4750)
350 WHILE ce1(v,t)<>-1:MOVE ce1(v,
t),ce2(v,t):CALL q:PRINT"j";:SOUND
 FNr,ce1(v,t)+100,1,15,4,3:MOVE ce
1(v,t),ce2(v,t):PRINT"j";:t=t+1:WE
ND:TAGOFF:PRINT a(1)                            (B4B2)
351 ORIGIN 1,1:BORDER 0:sc=xt1:xt1
=xt1+2:TAGOFF:PRINT a(1):GOSUB 168
:GOSUB 465                                      (3E32)
352 IF v=24 THEN vn=260:na=80:st=-
14:t=226:w1=2:GOSUB 105:GOTO 256                (44ED)
353 vn=260:na=80:st=-14:t=360:w1=-
2:GOSUB 105:GOTO 285                            (3DBC)
354 :                                           (06C9)
355 IF kt(4)=1 THEN kt(4)=0:LOCATE
 22,8:GOSUB 359:GOTO 358                        (2B2D)
356 v=20:GOSUB 111:l1=20:l2=6:GOSU
B 374:d=2:GOSUB 559:GOSUB 185:GOSU
B 165:GOSUB 369:GOSUB 360                       (47F1)
357 GOSUB 154:kt(3)=1:PEN 1:LOCATE
 17,8:PRINT"e":l1=20:l2=6:GOSUB 37
5:v=22:GOSUB 110                                (4573)
358 RESTORE 358:GOSUB 284:GOTO 256
:DATA 270,80,-14,330,-5                         (27A3)
359 PRINT" ":GOSUB 162:sc=1:GOSUB
461:RETURN                                      (1DAD)
360 GOSUB 186:sc=xt1:GOSUB 459:mg=
mg+5:m=10:GOSUB 189:RETURN                      (3803)
361 IF kt(3)=1 THEN kt(3)=0:LOCATE
 17,8:GOSUB 359:GOTO 364                        (2B36)
362 v=21:GOSUB 111:l1=19:l2=6:GOSU
B 374:d=1:GOSUB 559:GOSUB 185:GOSU
B 165:GOSUB 366:GOSUB 360                       (478B)
363 GOSUB 154:kt(4)=1:PEN 1:LOCATE
 22,8:PRINT"e":l1=19:l2=6:GOSUB 37
5:v=23:GOSUB 110                                (4546)
364 RESTORE 364:GOSUB 320:GOTO 285
:DATA 270,70,-14,254,5                          (261D)
365 :                                           (06DF)
366 PAPER 2:PEN 3:IF ps1=0 THEN ps
1=1:LOCATE 35,mll:PRINT"w":PAPER 0
:PEN 2:RETURN                                   (3597)
367 mt1=mt1+1:mult=mult+1:IF mt1=1
0 THEN mll=20:GOTO 368 ELSE IF mt1
>10 THEN ps1=0:mll=20:mult=mult-1:
RETURN                                          (738F)
368 GOSUB 372:ps1=0:LOCATE 36,mll:
PRINT"x":PAPER 0:PEN 2:mll=mll-1:R
ETURN                                           (3AEF)
369 PAPER 2:PEN 3:IF ps2=0 THEN ps
2=1:LOCATE 38,mlr:PRINT"x":RETURN               (2FEC)
370 mt2=mt2+1:mult=mult+1:IF mt2=1
0 THEN mlr=20:GOTO 371 ELSE IF mt2
>10 THEN ps2=0:mlr=20:mult=mult-1:
RETURN                                          (7372)
371 GOSUB 372:ps2=0:LOCATE 37,mlr:
PRINT"w":PAPER 0:PEN 2:mlr=mlr-1:R
ETURN                                           (3A9D)
372 PAPER 2:PEN 3:INK 3,6,26:LOCAT
E 36,11:PRINT USING"##";mult:SOUND
 7,50,100,7,1,3:GOSUB 159:INK 3,6:
RETURN                                          (4309)
373 :                                           (06EF)
374 PRINT a(5):PEN 1:LOCATE 11,12:
PRINT"i";:PEN 2:PRINT av+"j":PRINT
 a(6):GOSUB 161:GOSUB 153:GOSUB 15
9:RETURN                                        (4B86)
375 SOUND 7,0,5,7,,,22:PEN 1:LOCAT
E 11,12:PRINT"i":GOSUB 154:PEN 2:R
ETURN                                           (317F)
376 :                                           (06F5)
377 IF kt(0)=1 THEN kt(0)=0:l1=10:
l2=11:GOSUB 502:SOUND 7,0,5,7,,,2:
sc=200:GOSUB 163:GOSUB 465:l1=10:l
2=12:GOSUB 502:GOTO 222                         (6E39)
378 l1=10:l2=11:GOSUB 502:l1=9:l2=
9:GOSUB 381:INK 0,6,0:GOSUB 156:GO
SUB 152:INK 0,0:GOSUB 161:mg=mg+15
:m=50:GOSUB 397:IF sc=0 AND kt(2)<
2 THEN sa=1:GOSUB 413 ELSE kg(pl)=
kg(pl)+1:GOSUB 401:GOSUB 179                    (AFCE)
379 sc=200:GOSUB 459:GOSUB 427:sc=
35:GOSUB 404:GOSUB 467:GOSUB 189:G
OSUB 550:sc=10:GOSUB 182:mult=mult
+2:GOSUB 372:neu=2:lo=0:GOTO 135                (6F14)
380 IF kt(8)=1 THEN kt(8)=0:SOUND
7,0,5,7,,,2:sc=100:l1=31:l2=15:GOS
UB 502:GOSUB 465:GOSUB 564:GOTO 19
1 ELSE l1=31:l2=15:GOSUB 502:l1=32
:FOR l2=15 TO 17:GOSUB 502:NEXT:ne
u=0:lo=0:GOTO 135                               (A280)
381 SOUND 7,0,5,7,,,10:TAGOFF:FOR
s=0 TO 100:PEN 1:LOCATE 11,12:PRIN
T"(":SOUND 7,s+1,2,15,4,3:PEN 2:LO
CATE 11,12:PRINT"j":NEXT:PEN 1:LOC
ATE 11,12:PRINT"(":GOSUB 167:RETUR
N                                               (7A9C)
382 :                                           (B602)
383 w=3:u=1:s=311:x=372:y=x:EVERY
10,0 GOSUB 389:FOR i=311 TO 414:OR
IGIN 208,364,s,i,x,y:CLG FNr:s=s-1
:NEXT:z=REMAIN(0):ORIGIN 208,364,2
08,414,380,364:PLOT 300,0,2                     (A254)
384 IF flag=0 THEN 390                          (1311)
385 CLG 3:FOR s=1 TO flag                       (179C)
386 TAG:FOR i=160 TO-LEN(anz)*13 S
TEP-10:SOUND 7,ABS(i+200),2,15,4,3
:MOVE i,14:PRINT anz;" ";:NEXT i                (56D4)
387 TAG:FOR i=-LEN(anz)*13 TO 160
STEP 10:SOUND 7,ABS(i+200),2,15,4,
3:MOVE i,14:PRINT" ";anz;:NEXT i                (55D0)
388 NEXT s:TAGOFF:GOSUB 392:ORIGIN
 1,1,1,639,1,399:flag=0:GOSUB 414:
RETURN                                          (33C7)
389 x=x+1:y=y-1:RETURN                          (1ED2)
390 CLG 3:TAG:FOR i=200 TO-LEN(anz
)*16 STEP-20:MOVE i,14:PRINT anz;"
  ";:NEXT                                       (3E00)
391 TAGOFF:GOSUB 392:ORIGIN 1,1,1,
639,1,399:flag=0:GOSUB 414:RETURN               (2D49)
392 s=372:FOR i=372 TO 388:ORIGIN
208,364,208,414,s,i:CLG 0:s=s-1:SO
UND 7,s,5,10,4,3:NEXT:RETURN                    (58D4)
393 anz="NOW YOU ARE IN YEAR"+STR$
(rt(pl)):GOSUB 151:INK 3,6:flag=0:
GOTO 382                                        (4B0A)
394 anz="TIME-BONUS":GOSUB 163:GOS
UB 151:GOSUB 151:flag=1:GOTO 382                (36DB)
395 anz="BUT THIS IS NOT THE PRESE
NT TIME!":flag=0:GOTO 382                       (3E3F)
396 anz="YOU ARE LOST IN THE TIME!
":GOSUB 164:flag=0:GOTO 382                     (3B6D)
397 anz="VERY GOOD!":flag=1:GOTO 3
82                                              (2735)
398 GOSUB 176:GOSUB 176:anz="SORRY
,YOU HAVE NOT ENOUGH ENERGY FOR TH
E TIME-WARP!":GOSUB 158:flag=0:GOT
O 382                                           (5F3E)
399 anz="!CONGRATULATION!":flag=2:
GOSUB 382                                       (2D6B)
400 anz="NOW YOU ARE IN THE PRESEN
T TIME!":flag=0:GOTO 382                        (3D4A)
401 anz="!EXTRABALL!":GOSUB 163:fl
ag=2:GOTO 382                                   (2DD2)
```

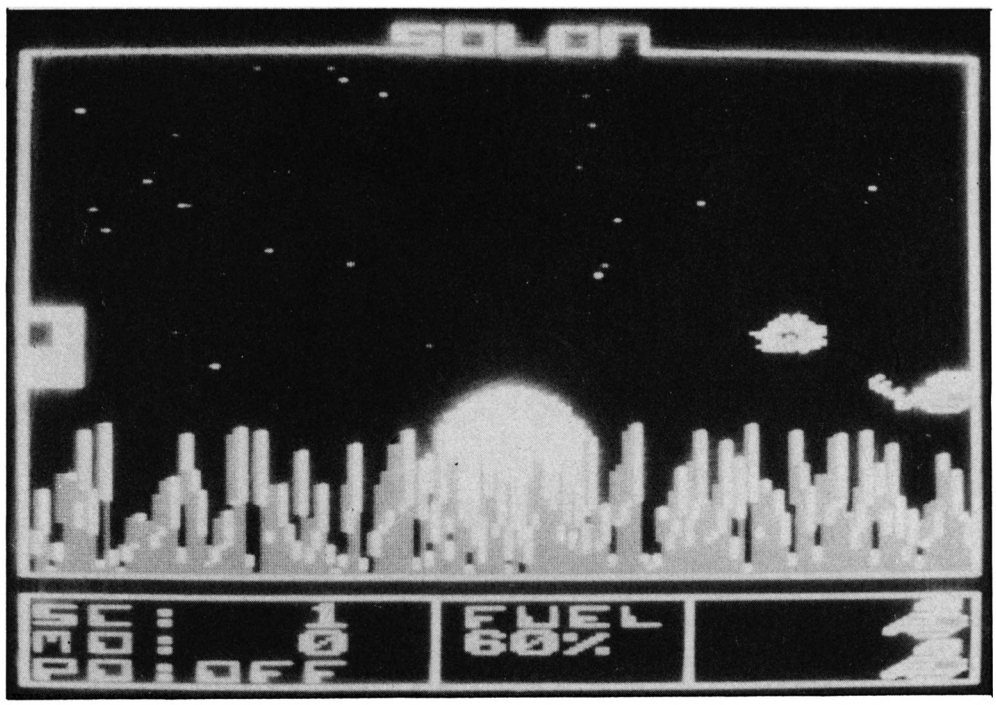
```
402 anz="!TIME-WARP!":flag=2:GOTO
382                                    (28AD)
403 anz="PLAYER NO."+STR$(pl):flag
=0:GOTO 382                            (3173)
404 anz="+"+STR$(sc*1000)+" BONUS"
:GOSUB 163:flag=1:GOTO 382             (3AE4)
405 anz="DON'T FORGET THE MAGNETFI
ELDS!":flag=0:lo=4:GOTO 382            (432E)
406 anz="PRESS (J) IN THE RIGHT MO
MENT TO CATCH THE BALL!":flag=0:GO
TO 382                                 (4DCE)
407 anz="PLAYER NO."+STR$(pl)+"
      ) GAME OVER (":FLAG=0:GOTO
 382                                   (4C5A)
408 anz="!!BALL!!":INK 3,4:INK 1,2
0,15:GOSUB 159:flag=3:GOTO 382         (3823)
409 anz="LEAVE THE TIME-CYLINDER!"
:flag=0:GOTO 382                       (3553)
410 anz="GOOD BYE":flag=2:GOTO 382
                                       (25A4)
411 anz="TIMEWARP-GATE OPEN":GOSUB
 163:flag=0:GOTO 382                   (34D6)
412 anz="SUPERBONUS-GATE OPEN":GOS
UB 163:flag=0:GOTO 382                 (361D)
413 anz="SHOOT AGAIN!":flag=2:GOTO
 382                                   (298E)
414 RESTORE 414:w=3:u=1:TAG:FOR i=
218 TO 408 STEP 16:READ z:PLOT 640
,0,FNr:MOVE i,380:PRINT CHR$(z);:N
EXT:TAGOFF:PLOT 640,0,2:RETURN:DAT
A 80,73,78,66,65,76,76,77,65,78,73
,65                                    (864B)
415 :                                  (0644)
416 vn=270:st=10:f=192:t=2:flag=0:
IF kt(1)=0 THEN 417 ELSE ON kt(t)G
OTO 428,429,430,431,432                (658A)
417 na=340:GOSUB 104:v=11:GOSUB 11
1:l1=9:l2=5:GOSUB 502:LOCATE 9,6:P
RINT"j":PEN 1:LOCATE 9,7:PRINT"c":
GOSUB 152                              (5196)
418 GOSUB 157:sc=xt2:GOSUB 460:mg=
mg+40:eg=eg+45:wt=wt+30:GOSUB 188:
xt2=xt2+5:sc=10:GOSUB 182:sc=20:GO
SUB 404:GOSUB 467:GOSUB 151            (8303)
419 IF eg<wt-rt(pl)THEN GOSUB 398:
GOTO 424                               (2899)
420 INK 1,10:INK 2,1:INK 3,11:GOSU
B 161:GOSUB 402                        (1F56)
421 w=26:u=1:INK 0,0,26:BORDER 0,2
6:FOR i=310 TO 2 STEP-2:SOUND 1,i,
3,13,4,3:SOUND 4,i,3,7,,,INT(i/10)
:INK 1,FNr:INK 2,FNr:INK 3,FNr:NEX
T:INK 1,26,0:GOSUB 155                 (85B7)
422 BORDER 0:INK 0,0:GOSUB 158:INK
 1,24:eg=eg-(wt-rt(pl)):rt(pl)=wt:
GOSUB 176:INK 2,20:GOSUB 169:GOSUB
 393:INK 3,6:IF rt(pl)<>2111 THEN
GOSUB 395                              (7C94)
423 flag=1:IF rt(pl)=2111 THEN INK
 3,4:INK 2,20:GOSUB 156:GOSUB 399:
GOSUB 409:GOSUB 410:GOSUB 161:FOR
i=1 TO 25:SOUND 7,3000-(i*25),100,
7,3:LOCATE#3,1,25:PRINT#3,CHR$(10)
:NEXT:GOSUB 161:LOCATE 17,12:PRINT
"THE END":GOSUB 160:GOSUB 160:RUN      (A002)
424 GOSUB 160:GOSUB 550:LOCATE 9,6
:PRINT ak(5):l1=9:l2=5:GOSUB 502:v
=12:GOSUB 111:na=270:vn=340:st=-10
:f=192:GOSUB 104:vn=na:na=80:w1=2:
GOSUB 105                              (8CD6)
425 IF flag=1 THEN kt(2)=5:SOUND 7
,0,5,7,,,20:PEN 1:LOCATE 13,4:PRIN
T ak(1):PEN 2                          (3E8F)
426 GOTO 256                           (094E)
427 PEN 3:LOCATE 17,16:PRINT ax:RE
TURN                                   (17D3)
428 na=340:GOSUB 104:SOUND 7,0,5,7
,,,2:GOSUB 163:GOTO 434                (2AE6)
429 na=330:GOTO 433                    (1329)
430 na=310:GOTO 433                    (130F)
431 na=300:GOTO 433                    (130F)
432 na=280                             (0E48)
433 GOSUB 104:SOUND 7,0,8,7,,,2        (16FE)
434 sc=30-kt(t):GOSUB 465:kt(t)=kt
(t)-1:IF kt(t)<0 THEN kt(t)=0          (551A)
435 st=-10:vn=na:na=270:GOSUB 104:
IF t=1 THEN 443                        (35C3)
436 vn=na:na=270:st=-10:f=192:GOSU
B 104:vn=na:na=80:w1=2:GOSUB 105:G
OTO 256                                (58A1)
437 vn=270:st=10:f=160:t=1:IF kt(t
)=0 THEN 438 ELSE ON kt(t)GOTO 440
,441,442                               (53A4)
438 na=310:GOSUB 104:SOUND 7,100,5
,15,4,3,2:TAGOFF:PEN 3:LOCATE 11,5
:PRINT"b";:PEN 2:LOCATE 11,6:PRINT
"j":sc=xt3:GOSUB 462:xt3=xt3+1         (6597)
439 d=0:GOSUB 559:INK 1,20,0:SOUND
 7,10,40,7,1:FOR s=500 TO 20 STEP-
20:SOUND 7,s,2,15,4,3:NEXT:sc=2:GO
SUB 467:PEN 3:LOCATE 11,5:PRINT"a"
+am+" ":INK 1,24:vn=na:na=270:st=-
10:GOSUB 104:GOTO 443                  (9E7A)
440 na=310:GOSUB 104:SOUND 7,0,5,7
,,,2:GOSUB 163:GOTO 434                (2AEF)
441 na=300:GOTO 433                    (1323)
442 na=280:GOTO 433                    (1311)
443 vn=270:na=80:w1=4:GOSUB 105:GO
TO 256                                 (2981)
444 PEN 2:LOCATE x,y:PRINT"j":GOSU
B 150:LOCATE x,y:PRINT" ":RETURN       (2E2C)
445 TAGOFF:SOUND 4,100,8,15,4,3,1      (178B)
446 IF ks<>1 THEN 449                  (1151)
447 x=28:y=10:GOSUB 444:IF kt(5)=1
 THEN kt(5)=0:d=3:GOSUB 559:GOSUB
454                                    (415E)
448 vn=244:na=80:st=-8:w1=-3:f=430
:GOSUB 105:GOTO 286                    (3BBA)
449 IF ks<>2 THEN 452                  (11AF)
450 x=29:y=11:GOSUB 444:IF kt(6)=1
 THEN kt(6)=0:d=4:GOSUB 559:GOSUB
454                                    (41AA)
451 vn=224:na=80:st=-8:w1=-4:f=440
:GOSUB 105:GOTO 286                    (3BB3)
452 x=30:y=12:GOSUB 444:IF kt(7)=1
 THEN kt(7)=0:d=5:GOSUB 559:GOSUB
454                                    (411A)
453 vn=210:na=80:st=-8:w1=-5:f=450
:GOSUB 105:GOTO 286                    (3BAC)
454 SOUND 7,100,15,15,2,3:sc=2:GOS
UB 461:FOR i=5 TO 7:IF kt(i)=1 THE
N RETURN ELSE NEXT                     (3F4A)
455 INK 2,0:INK 3,3,6:INK 1,20,0:G
OSUB 549:GOSUB 167:GOSUB 161:INK 2
,14:GOSUB 394:wt=wt+40:GOSUB 173:s
c=5:GOSUB 467                          (53A5)
456 BORDER 0:sc=INT(wt/10):GOSUB 4
59:eg=eg+45:GOSUB 176:mg=mg+30:GOS
UB 170:INK 3,6:INK 1,24:INK 2,20:k
t(5)=1:kt(6)=1:kt(7)=1:PEN 1:PAPER
 0:LOCATE 28,10:PRINT ak(3):SOUND
7,0,8,15,4,3,20:RETURN                 (A356)
457 :                                  (0698)
458 sum=10:GOTO 463                    (1335)
459 sum=100:GOTO 463                   (1364)
460 sum=500:GOTO 463                   (14B5)
461 sum=1000:GOTO 463                  (14CE)
462 sum=5000                           (0FED)
463 TAGOFF:FOR i=1 TO sc:pt(pl)=pt
(pl)+sum:SOUND 1,100+(i*2),3,12,4,
3:GOSUB 466:NEXT:RETURN                (5638)
464 :                                  (06A6)
465 TAGOFF:pt(pl)=pt(pl)+(100*sc)      (2B76)
466 LOCATE#pl,2,1:PRINT#pl,USING"#
#######";pt(pl):RETURN                 (333D)
467 FOR i=1 TO sc:bon=bon+1:PEN 3:
LOCATE 17,14:PRINT STRING$(6,207):
PEN 2:SOUND 7,bon+2,3,15,4,3:LOCAT
E 17,14:PRINT CHR$(246);:PEN 1:PRI
NT USING"####";bon;:PEN 2:PRINT CH
R$(247):NEXT:PEN 1:RETURN              (84EA)
468 :                                  (06AE)
469 w=10:u=1:ORIGIN 1,1:l1=9:l2=13
:GOSUB 322:IF FNr>2 THEN 473           (3B8A)
470 w=2:u=1:ON FNr GOTO 471,472        (21CE)
471 vn=174:na=80:st=-12:w1=14:f=13
6:GOSUB 106:GOTO 256                   (3B95)
472 TAGOFF:PEN 2:l1=9:l2=15:GOSUB
502:GOSUB 481:IF FNr=1 THEN 482 EL
SE PEN 2:LOCATE 9,15:PRINT"j":GOSU
B 151:LOCATE 9,15:PRINT" ":GOTO 46
9                                      (55F0)
473 TAG:FOR i=174 TO 134 STEP-30:M
OVE 136,i:CALL q:PRINT"j";:MOVE 13
6,i:CALL q:PRINT" ";:NEXT:TAGOFF       (4473)
474 :                                  (06BA)
475 w=10:u=1:ORIGIN 1,1:l1=9:l2=18
:GOSUB 322:IF FNr>2 THEN 479           (3B17)
476 w=2:u=1:ON FNr GOTO 477,478        (2104)
477 v=13:GOSUB 111:f=350:GOTO 286      (1F9E)
478 TAGOFF:PEN 2:l1=9:l2=17:GOSUB
502:GOSUB 481:IF FNr=1 THEN 482 EL
SE PEN 2:LOCATE 9,17:PRINT"j":GOSU
B 151:LOCATE 9,17:PRINT" ":GOTO 47
5                                      (552B)
479 TAG:FOR i=144 TO 184 STEP 30:M
OVE 136,i:CALL q:PRINT"j";:MOVE 13
6,i:CALL q:PRINT" ";:NEXT:TAGOFF:G
OTO 469                                (48FD)
480 :                                  (06C6)
481 l1=8:l2=16:GOSUB 374:GOSUB 153
:sc=10:GOSUB 460:mg=mg+5:m=15:GOSU
B 189:sc=5:GOSUB 182:GOSUB 550:sc=
1:GOSUB 467:l1=8:l2=16:GOSUB 375:R
ETURN                                  (7FD7)
482 vn=150:na=100:st=-5:w1=20:f=13
6:GOSUB 106:GOTO 249                   (3A21)
483 :                                  (06CC)
484 ORIGIN 1,1:LOCATE 1,1:PRINT a(
0):IF lo<3 THEN f=170:w1=190:flag=
1:GOTO 486                             (40CC)
485 f=410:w1=420:flag=0                (216B)
486 WHILE INKEY(19)=0 AND mg>0         (16E3)
487 TAG:MOVE f,110:PRINT"j";:MOVE
f,110:CALL q:PRINT"j";                 (2A17)
488 IF flag=1 THEN f=f-10:IF f<w1-
30 THEN flag=0                         (35D3)
489 IF flag=0 THEN f=f+10:IF f>w1
THEN flag=1                            (321F)
490 TAGOFF:LOCATE 1,1:PRINT a(1):m
g=mg-1:PEN 1:LOCATE 3,5:PRINT USIN
G"###";mg:SOUND 7,mg+f-100,5,5,3,1
:PRINT a(0)                            (5A18)
491 WEND:TAGOFF:PRINT a(1):IF mg=0
 THEN SOUND 7,100,70,7,3               (27E3)
492 IF lo<3 THEN IF f>170 THEN l1=
14:l2=20:GOSUB 502:GOTO 240 ELSE 4
94                                     (378B)
493 IF f<410 THEN l1=25:l2=20:GOSU
B 502:GOTO 249                         (2AF4)
494 IF lo<3 THEN l1=11:l2=21:GOSUB
 502:GOTO 495 ELSE 498                 (2EA7)
495 :                                  (06E4)
496 IF lo=1 THEN l1=10:l2=21:GOSUB
 502:l2=22:GOSUB 374:sc=5:GOSUB 46
0:GOSUB 182:l1=10:l2=22:GOSUB 375:
l2=23:GOSUB 502:l2=24:GOSUB 381:mg
=mg+5:m=10:GOSUB 189:lo=0:neu=2:GO
TO 135                                 (A8A8)
497 IF lo=2 THEN l1=12:l2=21:GOSUB
 502:l2=22:GOSUB 374:sc=10:GOSUB 4
60:GOSUB 182:l1=12:l2=22:GOSUB 375
:l1=13:l2=23:GOSUB 502:vn=232:na=3
00:st=2:GOTO 503                       (93DD)
498 LOCATE 28,20:PRINT"j":GOSUB 15
1:LOCATE 28,20:PRINT" "                (21FC)
499 IF lo=3 THEN l1=27:l2=21:GOSUB
 502:l2=22:GOSUB 374:sc=10:GOSUB 4
60:GOSUB 182:l1=27:l2=22:GOSUB 375
:l1=26:l2=23:GOSUB 502:vn=378:na=2
92:st=-2:GOTO 503                      (9583)
500 l1=29:l2=21:GOSUB 502:l2=22:GO
SUB 374:sc=5:GOSUB 460:GOSUB 182:l
1=29:l2=22:GOSUB 375:l1=30:l2=21:G
OSUB 502:l1=31:GOSUB 502:l1=32:GOS
UB 502:neu=2:GOTO 136                  (9362)
501 :                                  (06F0)
502 PEN 2:LOCATE l1,l2:PRINT"j":GO
SUB 151:LOCATE l1,l2:PRINT" ":RETU
RN                                     (320F)
503 TAG:FOR i=vn TO na STEP st:SOU
ND 7,INT(i/3),1,12,4,3:MOVE i,30:P
RINT"j";:NEXT:MOVE i,30:PRINT" ";:
TAGOFF:GOSUB 164:GOTO 325              (6188)
504 PLOT 640,0,2:l1=30:TAG             (1809)
505 READ a$:x=VAL("&"+a$):IF x=&FF
 THEN TAGOFF:RETURN                    (2A13)
506 MOVE l1,280:PRINT CHR$(x);:l1=
```

```
:1+14:GOTO 505                                  (2EB3)
507 :                                           (06FC)
508 v=14:GOSUB 111:GOSUB 427                    (166C)
509 l1=29:l2=3:GOSUB 374:PEN 2:LOC
ATE 26,8:PRINT"o":LOCATE 26,7:PRIN
T"q":PEN 1:kt(8)=1:LOCATE 31,15:PR
INT"ɡ":GOSUB 154:sc=3:GOSUB 467:w=
6:u=1:sc=10:GOSUB 460:IF FNr)3 AND
 wt()2111 THEN 516 ELSE w=3:ON FNr
 GOTO 510,513,513                               (ADD4)
510 PEN 2:LOCATE 28,7:PRINT"y":GOS
UB 166:GOSUB 375:l1=28:GOSUB 522                (2A19)
511 PAPER 0:IF kt(1)>0 THEN FOR i=
8 TO 6 STEP-1:GOSUB 151:LOCATE 11,
i:PRINT" ":GOSUB 154:NEXT:kt(1)=0:
GOSUB 412                                       (4A8B)
512 l1=28:l2=6:GOSUB 375:l2=7:GOSU
B 502:l1=29:l2=8:GOSUB 502:GOTO 52
3                                               (4278)
513 PEN 2:LOCATE 30,7:PRINT"y":GOS
UB 166:GOSUB 375:l1=30:GOSUB 522                (2AFC)
514 PAPER 0:IF kt(2)>0 THEN FOR i=
8 TO 4 STEP-1:GOSUB 151:LOCATE 13,
i:PRINT" ":GOSUB 154:NEXT:kt(2)=0:
GOSUB 411                                       (4A35)
515 l1=30:l2=6:GOSUB 375:l2=7:GOSU
B 502:l2=8:GOSUB 502:GOTO 523                   (39F4)
516 GOSUB 375:v=15:GOSUB 111:l1=32
:PAPER 3:RESTORE 94:FOR l2=6 TO 9:
READ x:LOCATE l1,l2:PRINT"j":GOSUB
 150:PEN 1:LOCATE l1,l2:PRINT CHR$
(x):PEN 2:NEXT:PAPER 0                          (71E8)
517 LOCATE l1,l2:PRINT"j";:PEN 1:P
RINT am+"c":GOSUB 152:GOSUB 165                 (2F83)
518 sc=xt2:GOSUB 460:eɡ=eɡ+40:GOSU
B 176:PAPER 3:LOCATE 14,2:PRINT CH
R$(24)+" PRESS SPACE "+CHR$(24):GO
SUB 167:w=wt+50:u=wt+20                         (733A)
519 WHILE INKEY(47)<>0:RESTORE 94:
FOR s=6 TO 9:READ x:LOCATE 32,s:PR
INT CHR$(x):SOUND 7,500,2,5,,2:PEN
 0:LOCATE 32,s:PRINT CHR$(x):PEN 1
:NEXT s:wt=FNr:GOSUB 175:WEND:GOSU
B 166:GOSUB 173:PAPER 0:LOCATE 14,
2:PRINT SPC(13):GOSUB 414                       (956F)
520 GOSUB 154:PEN 1:LOCATE 32,10:P
RINT ak(5):l1=32:RESTORE 520:PAPER
 3:FOR l2=9 TO 6 STEP-1:PEN 2:LOCA
TE l1,l2:PRINT"j":PEN 0:READ x:LOC
ATE l1,l2:PRINT CHR$(x):NEXT:PEN 1
:DATA 69,77,73,84                               (84F1)
521 v=16:GOSUB 111:PAPER 0:GOTO 50
9                                               (19DA)
522 l2=4:GOSUB 502:l2=5:GOSUB 502:
l2=6:GOSUB 374:GOSUB 185:mult=mult
+1:GOSUB 372:sc=20:GOSUB 460:GOSUB
 186:RETURN                                     (5CD0)
523 :                                           (061C)
524 l1=30:l2=9:GOSUB 381:GOSUB 167
:GOSUB 159:neu=2:lo=0:GOTO 135                  (3A4C)
525 :                                           (0620)
526 a="":LOCATE 13,18:PRINT"(1 OR
2) PLAYERS ":GOSUB 528:sp=VAL(a):I
F sp<1 OR sp>2 THEN 526                         (514B)
527 CLS:a="":LOCATE 7,18:PRINT"LEV
EL 1-3 (1=EASY/3=STRONG)":GOSUB 52
8:lv=VAL(a):IF lv<1 OR lv>3 THEN 5
27 ELSE CLS:RETURN                              (617C)
528 LOCATE 20,20:PRINT CHR$(7)+"_"
:WHILE INKEY$<>"":WEND:a="":WHILE
a="":a=INKEY$:WEND:LOCATE 20,20:PR
INT UPPER$(a):GOSUB 159:RETURN                  (52DA)
529 :                                           (0628)
530 DATA 0,0,0,18,18,7E,FF,7E,0,0,
0,7E,FF,7E,18,18,99,FF,3C,18,18,18
,18,18,18,18,18,18,18,18,18,18,0,0
,0,0,0,FF,81,FF,0,3,5,B,15,29,51,A
1,20,70,D8,6C,36,1B,E,4,4,E,1B,36,
6C,D8,70,20,3C,5A,C3,99,81,E7,66,3
C,0,0,3C,6E,76,7E,3C,0,0,0,3C,66,5
A,66,3C,0                                       (F521)
531 DATA 0,0,18,24,24,24,24,24,24,
24,24,24,24,18,0,0,0,0,0,FF,0,0,0,
0,0,0,7E,C3,7E,0,0,0,0,7E,56,C3,6A
,7E,0,0,0,7E,3C,18,0,0,0,0,0,0,0,0
,0,0,0,0,F,30,40,C0,87,88,88,88,88
,88,88,87,C0,60,30,F,11,11,11,E1,3
,6,1C,F0,F0,1C,6                                (DA56)
532 DATA 3,E1,11,11,11,1E,3E,7E,7E
,7E,7E,3E,1E,78,7C,7E,7E,7E,7E,7C,
78,0,18,3C,7E,0,0,0,0,0,10,30,70,7
0,30,10,0,3C,66,C3,99,99,C3,66,3C,
80,F0,48,24,12,A,6,3,7E,81,66,DB,D
B,66,81,7E,F,39,63,CE,18,70,C0,0,0
,0,0,3,E,18,13,1E,F0,9C                         (E177)
533 DATA C6,73,18,E,3,0,0,0,0,C0,7
0,18,C8,78,FE,3,3,FE,0,0,0,0,F,10,
10,F,0,0,0,0,7F,C0,C0,7F,0,0,0,0,F
0,8,8,F0,0,0,0,0,0,0,18,3C,3C,18,0
,0,0,C0,A0,D0,A8,94,8A,85,84,82,85
,88,90,A0,C0,FF,80,40,A0,50,28,14,
A,FE,21,41,A1,11,9,5,3,FF                       (E3B5)
534 DATA 1,2,5,A,14,28,50,7F,BD,3D
,DD,ED,F5,F9,1,FF,BD,BD,BD,A5,A5,B
D,BD,BD,FE,1,F9,F5,ED,DD,3D,BD,FF,
0,FF,E7,E7,FF,0,FF,7F,80,9F,AF,B7,
BB,BC,BD,BD,BC,BB,B7,AF,9F,80,7F                (A6A7)
535 DATA 47,41,4D,45,20,44,45,53,4
9,47,4E,20,26,20,50,52,4F,47,52,41
,4D,4D,49,4E,47,20,42,59,20,45,2E,
4A,2E,4B,4C,41,53,41,52,45,4B,FF                (84C9)
536 DATA 19,6,ii,17,8,e,22,8,e,18,
4,k,21,4,k,26,5,"!",9,9,£,32,13,i,
8,16,i,10,11,h,27,22,i i,26,6,i,12
,22,i,31,15,ɡ,30,9,£                            (789B)
537 DATA 30,6,i,29,3,i,15,7,k,24,7
,k,35,3,TIME,35,7,TIME,9,4,k,-1,-1
,""                                             (452D)
538 DATA 1,1,144,5,143,142,33,1,14
4,6,143,142,1,25,145,5,143,140,33,
25,145,6,143,140,2,2,144,3,143,142
,2,3,145,3,143,140,2,23,144,3,143,
142,2,24,145,3,143,140,2,4,144,3,1
43,142,2,6,145,3,143,140,2,8,144,3
,143,142                                        (D262)
539 DATA 2,10,145,3,143,140,3,12,1
44,2,143,142,3,14,145,2,143,140,3,
16,144,2,143,142,3,18,145,2,143,14
0,3,20,144,2,143,142,3,22,145,2,14
3,140,34,23,144,4,143,142,34,24,14
5,4,143,140                                     (B30E)
540 DATA 0,-20,90,56,52,2,0,-16,90
,86,78,1,0,-16,90,54,44,2,1,-30,94
,22,20,3,456,322,1,-6,90,58,48,2,1
68,338,0,-10,95,18,12,4,1,-4,92,42
,40,2,0,0                                       (8FC0)
541 DATA ,,220,260,210,286,210,338
,,,186,338,186,286,190,258,,,126,3
38,126,286,144,286,144,328,150,336
,,,190,258,180,272,180,336,154,336
,154,268,168,232,164,232,144,278,1
20,278,142,206,,,398,320,398,278,3
82,258,,,416,320,424,312,424,292,4
90,226                                          (F2AC)
542 DATA 490,312,504,316,,,490,214
,430,274,410,248,416,278,416,318,,
,490,214,470,178,474,174,494,288,5
14,208,514,30,494,30,494,154,474,1
24,,,128,178,110,160,110,136,126,1
20,126,80,142,60,142,10,160,10,160
,50,168,46,180,20,278,10,278,4                  (E809)
543 DATA 324,4,324,10,464,20,464,9
4,,,104,396,104,1,518,1,518,396,10
2,396,102,2,520,2,520,398,104,398,
,,108,394,108,382,206,382,206,362,
416,362,416,382,512,382,512,394,,,
280,392,280,386,340,386,340,392,28
0,392,,,514,208,514,30,,,512,288,5
12,30                                           (F1AA)
544 DATA ,,250,194,352,194,352,174
,250,174,250,194,,,246,198,356,198
,356,170,246,170,246,198,,,366,352
,366,334,400,334,400,352,366,352,-
1,-1                                            (8A58)
545 DATA 10,11,h,17,8,e,22,8,e,31,
15,ɡ,-1,-1                                      (2A53)
546 DATA 19,8,9,13,29,18,9,18                   (1BFA)
547 DATA 2,5,3,2,9,3,3,13,2,3,17,2
,3,21,2                                         (27B0)
548 DATA 11,11,11,22,16,23,23,23,2
7,5,29,5,31,5,8,17,19,7,20,7,29,23
                                                (42FB)
549 mk=w:w=25:BORDER FNr,FNr:w=mk:
RETURN                                          (31D6)
550 kt(0)=1:GOSUB 154:PAPER 0:PEN
1:LOCATE 10,11:PRINT"h":RETURN                  (28F4)
551 SOUND 7,80,3,5:RETURN                       (10F4)
552 RESUME NEXT                                 (0757)
553 DATA 296,60,160,20,-10,90,90,3
34,66,160,30,-8,120,120,314,66,160
,20,-8,110,120,314,66,160,20,-10,1
10,120,290,70,160,10,-8,70,100,280
,70,150,10,-10,64,70,296,60,20,160
,10,90,90,280,40,40,150,8,130,110,
280,40,40,150,8,130,110                         (E153)
554 DATA 300,40,30,160,8,80,120,25
0,60,30,130,10,130,110,150,340,40,
150,12,32,30,160,340,150,40,-12,32
,30,240,110,150,-20,-10,120,80,450
,322,140,100,-10,50,40,450,322,60,
20,-10,50,40,450,322,20,60,10,50,4
2                                               (CB2E)
555 DATA 304,110,40,160,8,160,60,3
24,110,20,190,10,110,100,330,110,3
0,190,10,90,90,294,286,-10,60,12,4
0,34,294,286,190,120,-12,40,34,294
,286,50,-20,-16,40,34,294,286,130,
200,16,40,34,294,286,7,190,13,70,6
8,294,286,180,-10,-13,70,68                     (E54D)
556 DEG:RESTORE 553:t=0:FOR s=0 TO
 25:READ o1(s),o2(s),vn,na,st,w1,w
2:ORIGIN o1,o2:FOR i=vn TO na STEP
 st:ce1(s,t)=FNk1:ce2(s,t)=FNk2:t=
t+1:NEXT i:ce1(s,t)=-1:ce2(s,t)=-1
:t=0:NEXT s:RETURN                              (F6C7)
557 DATA 34,22,yMULTy,2,7,yMGTy,34
,5,yREALy,34,9,yWARPy,2,11,yERGy,3
,15," 1y ",3,19," 2y "                          (5854)
558 DATA 12,10,P,17,9,I,22,9,N,27,
11,B,28,12,A,29,13,L,30,16,L                    (3C25)
559 IF bs(d)=1 THEN RETURN                      (1527)
560 PAPER 3:PEN 1:LOCATE x(d),y(d)
:PRINT ab(d):PAPER 0:bs(d)=1:FOR i
=0 TO 6:IF bs(i)=0 THEN RETURN ELS
E NEXT                                          (5D66)
561 FOR i=1 TO 25:GOSUB 163:NEXT:s
c=3:FOR i=40 TO 4 STEP-1:GOSUB 549
:FOR s=0 TO 6:PAPER 2:PEN 3:LOCATE
 x(s),y(s):PRINT ab(s):SOUND 7,s*i
,3,15,4,3:GOSUB 465:PAPER 3:PEN 2:
LOCATE x(s),y(s):PRINT ab(s):NEXT
s,i                                             (B532)
562 m=50:mɡ=mɡ+30:GOSUB 189:sc=25:
mult=mult+2:GOSUB 404:GOSUB 467:GO
SUB 372:BORDER 0:GOSUB 83:PAPER 0               (5530)
563 FOR i=0 TO 6:bs(i)=0:NEXT:RETU
RN                                              (2079)
564 IF bs(6)=1 THEN t=1 ELSE t=0                (1F75)
565 f=31:FOR i=15 TO 17:LOCATE f,i
:CALL q:PRINT"j":GOSUB 150:PAPER 3
:PEN t:LOCATE 30,16:PRINT ab(6):PA
PER 0:PEN 2:LOCATE f,i:PRINT" ":f=
f-1:NEXT:RETURN                                 (7380)
```




SOLON

Solon ist ein hochkarätiges Action-Spiel für die CPCs. Geschrieben wurde es von unserem Autor Michael Dudenhöffer aus Taunusstein. Um in den vollen Genuß der Vielzahl von Farben zu kommen, empfiehlt sich ein Farbmonitor.



Aus den Laboren von Ters wurde eine Superwaffe von Spionen der fliegenden Stadt Solon geraubt. Ihr Auftrag lautet, die Waffe, die von einem computergesteuerten Raumschiff, Flügler genannt, bewacht wird, zurückzuholen. Dazu müssen vier verschiedene Screens durchflogen werden.
In den drei ersten Screens werden Sie von Ufos der fliegenden Stadt angegriffen, die Sie aber mit Hilfe Ihres eingebauten Lasers vernichten können. Hüten Sie sich auch vor den Asteroiden, die Ihr Raumschiff zerstören könnten. Aber ganz so hilfslos sind Sie nicht. Nach jedem zweiten abgeschossenen Raumschiff wird für 20 Sekunden ein Schutz aktiviert, der die Asteroiden abwehrt.
Im vierten Screen wartet der stärkste Flügler auf Sie. Er bewacht die Waffe, und Zusammenstöße mit ihm sind äußerst ungesund. Ihr Laser ist gegen ihn wirkungslos. Sie können ihn nur zerstören, indem Sie ihn auf einen Felsen locken, der irgendwo im Raum schwebt. Ist Ihnen das gelungen, müssen Sie nur noch dicht über die Waffe fliegen, die dann automatisch nach Ters gebeamt wird.
Geschafft? Denkste! Sie müssen nämlich wieder in den ersten Screen zurückfliegen. Diese Reise ist kein Vergnügen, denn die Angreifer werden Ihnen das Leben so schwer wie möglich machen. Aber es ist höchste Eile geboten, denn der Tank leert sich zusehends.
Solon enthält einen Tasteneditor, mit dem Sie sich Ihre individuelle Tastaturbelegung einstellen können. Selbstverständlich kann auch der Joystick benutzt werden.

TB/Michael Dudenhöffer □

```
10 '******************************* <24C3>
20 '*            SOLON            * <24CC>
30 '*             VON             * <2429>
40 '*    MICHAEL DUDENHOEFFER     * <249C>
50 '*            FUER             * <245F>
60 '*     SCHNEIDER CPC-WELT      * <24E2>
70 '*      CPC 464/664/6128   tb* <2474>
71 '******************************* <243C>
80 CLS:MODE 0:SPEED INK 5,5       <0F69>
90 CLEAR:KEY DEF 68,0,169,1,0     <14E9>
100 DIM ac(9),te$(9),b(5)         <1F20>
110 REM ENV's und ENT's *****     <1C6A>
120 ENV 1,3,-2,85                 <0F0D>
130 ENV 2,0,-1,1,1,0,1,1,0,1,12,-1
    ,8,2,-1,20                    <2AD9>
140 ENV 3,40,127,3,10,2,3,1,2,3   <1C20>
150 ENV 5,11,-1,25                <1080>
160 ENV 6,21,-15,4                <1092>
170 ENV 7,15,1,5                  <0E80>
180 ENV 8,15,-1,2                 <0FBB>
190 ENV 9,10,-1,10,15,-1,2        <18E0>
200 ENV 10,40,-1,10               <11B6>
210 ENT 1,5,-1,51                 <0F25>
220 ENT-2,100,1,3,1,-1,3,1,0,1,1,1
    ,3,1,-1,3                     <29BB>
230 ENT 3,39,-1,255,50,120,144    <19EA>
240 ENT 4,50,-1,1                 <0FB7>
250 ENT 5,9,49,5,9,-10,15         <172F>
260 ENT 6,1,120,255               <0F70>
270 ENT 7,120,-10,2               <1073>
280 ENT 8,10,-12,2                <103D>
290 ENT 9,5,5,2,2,1,2             <13CE>
300 ENT 10,40,-1,5                <102F>
310 REM********* SYMBOLE *********
                                  <21CC>
320 SYMBOL AFTER 256              <0A09>
```

```
330 MEMORY &A500                                    <093D>
340 SYMBOL AFTER 238                                <091C>
350 SYMBOL 238,&0,&7E,&42,&42,&7E,
&60,&60,&0:'P                                       <2C03>
360 SYMBOL 239,&0,&7E,&42,&42,&62,
&62,&7E,&0:'O                                       <2C36>
370 SYMBOL 244,&0,&7E,&60,&60,&7E,
&6,&7E,&0:'S                                        <2CF4>
380 SYMBOL 245,&0,&7E,&40,&40,&60,
&60,&7E,&0:'C                                       <2CBB>
390 SYMBOL 246,&0,&7E,&40,&40,&78,
&60,&60,&0:'F                                       <2C59>
400 SYMBOL 247,&0,&42,&42,&42,&62,
&62,&7E,&0:'U                                       <2C0D>
410 SYMBOL 248,&0,&7E,&40,&40,&78,
&60,&7E,&0:'E                                       <2CFF>
420 SYMBOL 249,&0,&40,&40,&40,&60,
&60,&7E,&0:'L                                       <2CD0>
430 SYMBOL 250,&0,&7C,&54,&54,&44,
&44,&44,&0:'M                                       <2C99>
440 SYMBOL 251,&0,&7E,&42,&42,&62,
&62,&62,&0:'N                                       <2C8B>
450 SYMBOL 252,&7F,&60,&60,&7F,&3,
&3,&FF:'S                                           <2833>
460 SYMBOL 253,&7F,&61,&61,&61,&61
,&61,&7F:'O                                         <2853>
470 SYMBOL 254,&60,&60,&60,&60,&60
,&60,&7F:'L                                         <28B8>
480 SYMBOL 255,&7F,&61,&61,&61,&61
,&61,&61:'N                                         <2888>
490 b(1)=8:b(2)=9:b(3)=11:b(4)=10:
b(5)=88                                             <392F>
500 GOSUB 3800                                      <09B5>
510 REM********** SPRITE-DATEN ****
*****                                               <26D6>
520 REM ------- Daten fuer stadt                     <1F56>
530 DATA &01,&08,&19,&00,&00,&00,&
00,&00,&40,&C0,&80,&00,&C0,&80,&00
,&00,&40                                            <4A91>
540 DATA &00,&80,&00,&80,&81,&03,&
00,&40,&C0,&80,&00,&C0,&81,&01,&33
,&62,&00                                            <4A99>
550 DATA &80,&00,&80,&81,&03,&33,&
62,&00,&80,&00,&C0,&81,&01,&23,&42
,&C0,&80                                            <4A10>
560 DATA &00,&80,&81,&03,&23,&42,&
00,&80,&00,&C0,&81,&01,&23,&42,&00
,&80,&00                                            <4AD4>
570 DATA &80,&81,&03,&23,&42,&C0,&
80,&00,&C0,&81,&01,&23,&42,&00,&80
,&00,&80                                            <4ABF>
580 DATA &81,&C0,&81,&42,&00,&80,&
00,&C0,&81,&80,&01,&42,&C0,&80,&00
,&80,&81                                            <4A49>
590 DATA &C0,&81,&42,&00,&80,&C0,&
C0,&C0,&C0,&C0,&C0,&C0,&C0,&C5,&CF
,&FC,&FC                                            <4A7A>
600 DATA &FC,&FC,&FC,&FC,&40,&CF,&
FC,&FC,&FC,&FC,&A8,&40,&CF,&DE
,&FC,&FC                                            <4ABB>
610 DATA &FC,&FC,&A8,&00,&C5,&DE,&
FC,&FC,&FC,&FC,&00,&00,&C5,&CF,&FC
,&FC,&FC                                            <4A74>
620 DATA &FC,&00,&00,&40,&CF,&FC,&
FC,&FC,&A8,&00,&00,&40,&C5,&FC,&FC
,&FC,&A8                                            <4AB7>
630 DATA &00,&00,&00,&C5,&DE,&FC,&
FC,&00,&00,&00,&00,&40,&CF,&FC,&A8
,&00,&00                                            <4AB6>
640 DATA &00,&00,&40,&C5,&FC,&A8,&
00,&00,&00,&00,&00,&40,&A8,&00,&00
,&00,&00                                            <4A5B>
650 DATA 9999                                       <0BCC>
660 stadt$=STRING$(212,0)                           <16DE>
670 RESTORE 530:a=@stadt$:a=PEEK(a
+2)*256+PEEK(a+1):w=0:WHILE w<>999
9:READ w:IF w<>9999 THEN POKE a,w                   <63D3>
680 a=a+1:WEND                                      <1274>
690 REM Daten fuer------vog                         <1A3E>
700 DATA &01,&08,&0A,&00,&00,&00,&
00,&00,&00,&00,&00,&00,&00,&00,&00
,&00,&00                                            <4A8D>
710 DATA &00,&00,&00,&00,&CF,&00,&
00,&CF,&00,&00,&00,&45,&0F,&8A,&45
,&0F,&8A                                            <4A0D>
720 DATA &00,&00,&8F,&C0,&4F,&8F,&
C0,&4F,&00,&45,&4A,&00,&80,&40,&00
,&85,&8A                                            <4A9D>
730 DATA &8F,&80,&00,&40,&80,&00,&
40,&4F,&4A,&00,&00,&00,&00,&00,&00
,&85,&80                                            <4AC3>
740 DATA &00,&00,&00,&00,&00,&00,&
40,&80,&00,&00,&00,&00,&00,&00,&40
,&00,&00                                            <4AB7>
750 DATA 9999                                       <0B95>
760 vog$=STRING$(92,0)                              <14D7>
770 RESTORE 700:a=@vog$:a=PEEK(a+2
)*256+PEEK(a+1):w=0:WHILE w<>9999:
READ w:IF w<>9999 THEN POKE a,w                     <610E>
780 a=a+1:WEND                                      <123E>
790 REM ------- Daten fuer vog2                      <1EE1>
800 DATA &01,&08,&0A,&00,&00,&00,&
00,&00,&00,&00,&00,&00,&00,&00,&00
,&00,&00                                            <4A55>
810 DATA &00,&00,&00,&CF,&8A,&00,&
00,&45,&CF,&00,&45,&0F,&4F,&8A,&45
,&8F,&0F                                            <4AE5>
820 DATA &8A,&0F,&C0,&85,&4F,&8F,&
4A,&C0,&0F,&C0,&00,&40,&80,&40,&80
,&00,&C0                                            <4A97>
830 DATA &00,&00,&00,&40,&80,&00,&
00,&00,&00,&00,&00,&00,&00,&00,&00
,&00,&00                                            <4A0F>
840 DATA &00,&00,&00,&00,&00,&00,&
00,&00,&00,&00,&00,&00,&00,&00,&00
,&00,&00                                            <4A5D>
850 DATA 9999                                       <0B5D>
860 vog2$=STRING$(92,0)                             <1548>
870 RESTORE 800:a=@vog2$:a=PEEK(a+
2)*256+PEEK(a+1):w=0:WHILE w<>9999
:READ w:IF w<>9999 THEN POKE a,w                    <6212>
880 a=a+1:WEND                                      <1205>
890 REM ------- Daten fuer rakete                    <202A>
900 DATA &01,&08,&05,&00,&E9,&30,&
20,&00,&00,&00,&00,&FC,&43,&1A,&30
,&30,&00                                            <4A4C>
910 DATA &00,&00,&A9,&03,&87,&25,&
30,&30,&10,&10,&FC,&43,&1A,&30,&30
,&00,&00                                            <4A71>
920 DATA &00,&00,&E9,&30,&20,&00,&
00,&00,&00,&00,&00,&00,&00,&00,&00
,&00,&00                                            <4A60>
930 DATA 9999                                       <0BFD>
940 rakete$=STRING$(52,0)                           <17F7>
950 RESTORE 900:a=@rakete$:a=PEEK(
a+2)*256+PEEK(a+1):w=0:WHILE w<>99
99:READ w:IF w<>9999 THEN POKE a,w
                                                    <6454>
960 a=a+1:WEND                                      <12A5>
970 REM ------- Daten fuer la                        <1CBB>
980 DATA &01,&08,&03,&74,&FC,&FC,&
03,&03,&43,&C3,&C3,&00,&00,&00,&00
,&00,&00                                            <4AB7>
990 DATA &00,&00,&74,&FC,&FC,&03,&
03,&43,&C3,&C3,&00,&00,&00,&00,&00
,&00,&00                                            <4AF8>
1000 DATA 9999                                      <0B8A>
1010 la$=STRING$(36,0)                              <132A>
1020 RESTORE 980:a=@la$:a=PEEK(a+2
)*256+PEEK(a+1):w=0:WHILE w<>9999:
READ w:IF w<>9999 THEN POKE a,w                     <60CF>
1030 a=a+1:WEND                                     <1233>
1040 REM ------- Daten fuer waffe                     <1F2F>
1050 DATA &01,&08,&19,&00,&00,&00,
&51,&80,&00,&00,&00,&00,&00,&45,&7
3,&C0,&80                                           <4AEF>
1060 DATA &00,&00,&00,&00,&87,&9B,
&E2,&C0,&00,&00,&00,&00,&87,&9B,&E
2,&C0,&00                                           <4A27>
1070 DATA &00,&00,&54,&43,&4F,&73,
&C0,&80,&00,&00,&54,&43,&4F,&73,&C
0,&80,&00                                           <4A27>
1080 DATA &00,&54,&43,&4F,&73,&C0,
&80,&00,&00,&FC,&43,&4F,&73,&C0,&C
0,&00,&00                                           <4AF4>
1090 DATA &FC,&A9,&87,&9B,&E2,&C0,
&00,&00,&FC,&A9,&87,&9B,&E2,&C0,&0
0,&00,&FC                                           <4AC7>
1100 DATA &A9,&87,&9B,&E2,&C0,&00,
&00,&FC,&A9,&87,&9B,&E2,&C0,&00,&0
0,&FC,&43                                           <4A4B>
1110 DATA &4F,&73,&C0,&C0,&00,&00,
&54,&43,&4F,&73,&C0,&80,&00,&00,&5
4,&43,&4F                                           <4A6F>
1120 DATA &73,&C0,&80,&00,&00,&54,
&43,&4F,&73,&C0,&80,&00,&00,&00,&8
7,&9B,&E2                                           <4A46>
1130 DATA &C0,&00,&00,&00,&00,&87,
&9B,&E2,&C0,&00,&00,&00,&00,&E5,&7
3,&C0,&D0                                           <4A4B>
1140 DATA &00,&00,&00,&95,&7A,&F1,
&D0,&B5,&6A,&00,&CF,&C0,&3F,&7A,&B
5,&3F,&C0                                           <4A76>
1150 DATA &CF,&8E,&1D,&95,&3F,&3F,
&6A,&2E,&4D,&8E,&2E,&C0,&95,&6A,&C
0,&1D,&4D                                           <4A27>
1160 DATA &9F,&0C,&48,&6A,&95,&84,
&0C,&6F,&CF,&CF,&CF,&C0,&C0,&CF,&C
F,&CF,&00                                           <4A91>
1170 DATA 9999                                      <0BDE>
1180 waffe$=STRING$(212,0)                          <1685>
1190 RESTORE 1050:a=@waffe$:a=PEEK
(a+2)*256+PEEK(a+1):w=0:WHILE w<>9
999:READ w:IF w<>9999 THEN POKE a,
w                                                   <63BD>
1200 a=a+1:WEND                                     <1286>
1210 REM ------- Daten fuer ufo                       <1D34>
1220 DATA &01,&06,&0A,&00,&00,&00,
&00,&00,&00,&10,&00,&00,&00,&00,&2
0,&20,&00                                           <4A8D>
1230 DATA &00,&00,&00,&10,&20,&20,
&00,&00,&10,&10,&1A,&00,&00,&00,&0
0,&25,&05                                           <4AAB>
1240 DATA &20,&80,&40,&10,&0A,&00,
&1A,&00,&00,&25,&00,&00,&05,&20,&1
0,&0A,&00                                           <4AE1>
1250 DATA &00,&00,&1A,&25,&00,&00,
&00,&00,&05,&0A,&00,&00,&00,&00,&0
0,&00,&00                                           <4AC8>
1260 DATA 9999                                      <0B93>
1270 ufo$=STRING$(70,0)                             <14A7>
1280 RESTORE 1220:a=@ufo$:a=PEEK(a
+2)*256+PEEK(a+1):w=0:WHILE w<>999
9:READ w:IF w<>9999 THEN POKE a,w                   <6169>
1290 a=a+1:WEND                                     <123C>
1300 REM ------- Daten fuer uff                       <1DC5>
1310 DATA &01,&06,&0A,&00,&00,&00,
&00,&00,&00,&00,&00,&00,&00,&00,&0
0,&00,&00                                           <4AB3>
1320 DATA &00,&00,&00,&00,&10,&00,
&00,&00,&00,&00,&20,&20,&00,&00,&0
0,&10,&20                                           <4A18>
1330 DATA &20,&00,&00,&10,&10,&1A,
&20,&80,&40,&10,&25,&05,&1A,&20,&1
0,&25,&0A                                           <4AB6>
1340 DATA &00,&05,&1A,&25,&0A,&00,
&00,&00,&05,&0A,&00,&00,&00,&00,&0
0,&00,&00                                           <4A20>
1350 DATA 9999                                      <0B47>
1360 uff$=STRING$(70,0)                             <14B4>
1370 RESTORE 1310:a=@uff$:a=PEEK(a
+2)*256+PEEK(a+1):w=0:WHILE w<>999
9:READ w:IF w<>9999 THEN POKE a,w                   <6128>
1380 a=a+1:WEND                                     <12F0>
1390 REM ------- Daten fuer ufo2                      <1E24>
1400 DATA &01,&07,&0C,&00,&00,&00,
&40,&C0,&C0,&00,&00,&00,&00,&91,&3
3,&62,&00                                           <4A8A>
1410 DATA &00,&00,&40,&33,&33,&62,
&00,&00,&00,&40,&C0,&C0,&C0,&00,&0
0,&00,&00                                           <4A63>
1420 DATA &00,&C0,&80,&00,&00,&00,
&0C,&0C,&0C,&0C,&00,&00,&04,&F0,&E
0,&03,&03                                           <4AA7>
1430 DATA &80,&00,&48,&C0,&C0,&03,
&03,&2A,&04,&03,&03,&03,&03,&03,&8
0,&09,&0C                                           <4A85>
1440 DATA &0C,&0C,&0C,&0C,&00,&0C,
&00,&80,&00,&00,&80,&00,&00,&00,&4
0,&C0,&C0                                           <4A22>
1450 DATA 9999                                      <0B10>
1460 ufo2$=STRING$(95,0)                            <1524>
1470 RESTORE 1400:a=@ufo2$:a=PEEK(
a+2)*256+PEEK(a+1):w=0:WHILE w<>99
99:READ w:IF w<>9999 THEN POKE a,w
                                                    <6213>
1480 a=a+1:WEND                                     <12B7>
1490 REM ------- Daten fuer ufor$                     <1FCA>
```

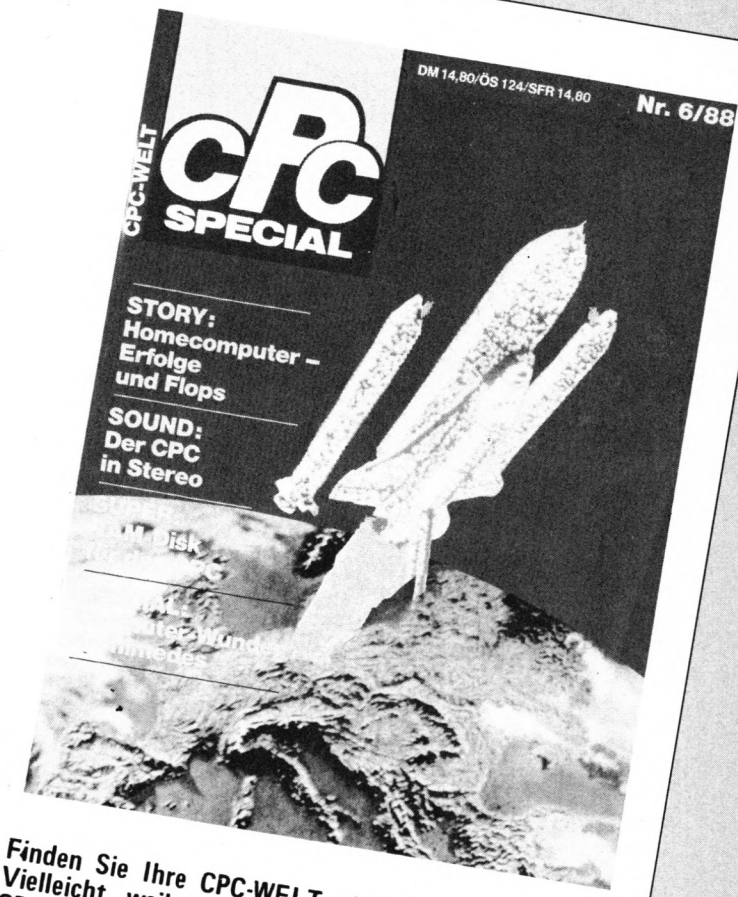
```
1500 DATA &01,&07,&0C,&00,&C0,&C0,
&80,&00,&00,&00,&00,&91,&33,&62,&0
0,&00,&00                          <4A1B>
1510 DATA &00,&91,&33,&33,&00,&00,
&00,&00,&C0,&C0,&C0,&00,&00,&00,&0
0,&40,&C0                          <4A3C>
1520 DATA &00,&00,&00,&00,&00,&0C,
&0C,&0C,&0C,&00,&00,&40,&03,&03,&D
0,&F0,&00                          <4A29>
1530 DATA &00,&15,&03,&03,&C0,&C0,
&84,&00,&40,&03,&03,&03,&03,&03,&0
0,&00,&0C                          <4A37>
1540 DATA &0C,&0C,&0C,&0C,&06,&00,
&40,&00,&00,&42,&00,&0C,&00,&C0,&C
0,&C0,&00                          <4A8F>
1550 DATA 9999                     <0BD9>
1560 utorg$=STRING$(95,0)          <1624>
1570 RESTORE 1500:a=@utorg$:a=PEEK
(a+2)*256+PEEK(a+1):w=0:WHILE w<>9
999:READ w:IF w<>9999 THEN POKE a,
w                                  <634A>
1580 a=a+1:WEND                    <1281>
1590 REM ------- Daten fuer kap     <1D6C>
1600 DATA &01,&07,&0C,&00,&00,&8A,
&00,&8A,&00,&00,&01,&02,&00,&00,&0
0,&00,&00                          <4AD3>
1610 DATA &00,&02,&08,&08,&08,&08,
&02,&00,&01,&40,&00,&40,&01,&00,&0
0,&01,&A8                          <4AB7>
1620 DATA &FC,&80,&A8,&00,&00,&00,
&00,&C0,&08,&A8,&00,&00,&00,&00,&C
0,&5C,&56                          <4AC1>
1630 DATA &A8,&03,&01,&A8,&00,&D4,
&09,&00,&01,&02,&40,&00,&00,&45,&0
0,&00,&00                          <4A34>
1640 DATA &08,&FC,&04,&45,&00,&00,
&00,&54,&00,&00,&45,&8A,&00,&54,&A
8,&45,&8A                          <4AE3>
1650 DATA 9999                     <0BA0>
1660 kap$=STRING$(95,0)            <1433>
1670 RESTORE 1600:a=@kap$:a=PEEK(a
+2)*256+PEEK(a+1):w=0:WHILE w<>999
9:READ w:IF w<>9999 THEN POKE a,w  <61B5>
1680 a=a+1:WEND                    <1248>
1690 REM Daten fuer------fs        <1968>
1700 DATA &01,&04,&08,&FC,&00,&E8,
&C0,&F8,&FC,&D0,&C0,&F8,&F4,&F8,&C
0,&FC,&F8                          <4A7A>
1710 DATA &E0,&C0,&54,&F0,&D0,&80,
&54,&F8,&F0,&80,&00,&FC,&C0,&00,&0
0,&54,&A8                          <4AEF>
1720 DATA 9999                     <0B2D>
1730 fs$=STRING$(40,0)             <1375>
1740 RESTORE 1700:a=@fs$:a=PEEK(a+
2)*256+PEEK(a+1):w=0:WHILE w<>9999
:READ w:IF w<>9999 THEN POKE a,w   <60BD>
1750 a=a+1:WEND                    <12D4>
1760 REM ------- Daten fuer ast     <1DC1>
1770 DATA &01,&07,&0D,&00,&00,&00,
&00,&00,&00,&00,&00,&00,&00,&80,&C
5,&00,&00                          <4AEC>
1780 DATA &00,&00,&45,&CF,&CF,&00,
&00,&00,&00,&CF,&DE,&C5,&8A,&00,&0
0,&00,&CF                          <4A96>
1790 DATA &C7,&E7,&E7,&8A,&00,&45,
&E7,&E9,&CB,&DB,&8A,&45,&CF,&CB,&F
0,&D6,&DB                          <4A4C>
1800 DATA &8A,&00,&CF,&CB,&F4,&F4,
&CA,&00,&00,&45,&E7,&E9,&C3,&CA,&8
A,&00,&45                          <4A5C>
1810 DATA &E7,&C7,&E7,&DB,&8A,&00,
&CF,&E7,&DB,&D1,&E7,&00,&00,&45,&C
F,&CF,&40                          <4A8E>
1820 DATA 9999                     <0BF6>
1830 ast$=STRING$(102,0)           <143C>
1840 RESTORE 1770:a=@ast$:a=PEEK(a
+2)*256+PEEK(a+1):w=0:WHILE w<>999
9:READ w:IF w<>9999 THEN POKE a,w  <61B7>
1850 a=a+1:WEND                    <129E>
1860 REM ------- Daten fuer kr      <1CF0>
1870 DATA &01,&07,&0C,&F0,&F0,&F0,
&C2,&F0,&F0,&F0,&F0,&F0,&F0,&C2,&F
0,&F0,&F0                          <4A74>
1880 DATA &F0,&F0,&C3,&C2,&C0,&F0,
&F0,&F0,&F0,&C0,&C2,&C0,&F0,&F0,&F
0,&F0,&F0                          <4AEC>
1890 DATA &C2,&F0,&F0,&F0,&F0,&F0,
&F0,&C2,&F0,&F0,&F0,&F0,&F0,&F0,&C
2,&F0,&F0                          <4A65>
1900 DATA &F0,&F0,&F0,&F0,&C2,&F0,
&F0,&F0,&F0,&F0,&F0,&C2,&F0,&F0,&F
0,&F0,&F0                          <4A41>
1910 DATA &F0,&C2,&F0,&F0,&F0,&F0,
&F0,&F0,&C2,&F0,&F0,&F0,&F0,&F0,&F
0,&C2,&F0                          <4AD7>
1920 DATA 9999                     <0BBD>
1930 kr$=STRING$(95,0)             <131E>
1940 RESTORE 1870:a=@kr$:a=PEEK(a+
2)*256+PEEK(a+1):w=0:WHILE w<>9999
:READ w:IF w<>9999 THEN POKE a,w   <604E>
1950 a=a+1:WEND                    <1265>
1960 DATA 010ba52120a5cdd1bcc916a5
c3c7a5c3cca5c3d1a54558c34745d45055
d4000000000000000000000000000fe03
2813fe04c287a5dd6e00dd23dd6600dd23
7d322ca5dd6e00dd6601228a5dd6e02dd
                                   <A762>
1970 DATA 6603226a5dd6e04dd660522
24a5dd2a28a5dd6e01dd6602222 8a5dd2a
28a5dd7e01322aa5dd7e02322ba5dd23dd
23dd23dd2228a5c9e1c900000000c5e5
dde5e106004f7ccbbffe4038077e4779cd
                                   <A7E5>
1980 DATA 38a67778e1c1c93a24a56f26
00328ca529eb3a26a56f3ec89d328da526
00cd1dbc228aa5c900f53e011808f53e02
1803f53e0332c6a5f1cd2da5cda8a52a28
a5dd2a8aa53a2ba547c53a8ca54fdd2a8a
                                   <A759>
1990 DATA a53a2aa547c53ac6a5fe01cc
24a63ac6a5fe02cc2aa63ac6a5fe03cc33
a623dd23c110e1e52a8aa5cd26bc228aa5
e1c110c6c97ecd8ea577c9afcd8ea577cd
8ea5c97ecd8ea5c9f53a2ca5fe01280cfe
                                   <A77E>
2000 DATA 03280bfe02280bf1ae77c9f1
77c9f1b677c9f1c54fe6aab728065f78e6
55b34779e655b728065f78e6aab34778c1
c900                               <6759>
2010 RESTORE 1960:ad=&A501:FOR x=1
 TO 5:READ d$(x):FOR y=1 TO LEN(d$
(x))STEP 2:w=VAL("&"+MID$(d$(x),y,
2)):POKE ad,w:ad=ad+1:NEXT y,x:CAL
L &A501                            <8E55>
2020 REM********* INITIALISIERUNG
*********                          <29C8>
2030 CLS:PEN 2:LOCATE 4,9:PRINT"LE
VEL [1-4]?"                        <1E4B>
2040 LOCATE 2,11:PRINT"LEICHT....S
CHWER"                             <1EDD>
2050 level$=INKEY$:IF level$=""THE
N 2050                             <21E4>
2060 level=VAL(level$):IF level<1
OR level>4 THEN 2050               <3536>
2070 IF level>2 THEN race=3 ELSE r
ace=2                              <259E>
2080 lif=level+1:flu=ROUND(level*1
.2):IF flu=5 THEN flu=6            <44E9>
2090 MODE 0:CLS:INK 0,0:BORDER 0:I
NK 1,26                            <172F>
2100 x=70:y=100:xu=4:yu=150:take=0
:go=1:sc=0:zuf=0:po=0:zz=0:po$=CHR
$(239)+STRING$(2,246):tu=100:zeit=
0                                  <81B5>
2110 REM********* AMATUREN *******
**                                 <2287>
2120 PLOT 28,384,4:DRAW 608,384:DR
AW 608,78:DRAW 28,78:DRAW 28,384   <2E35>
2130 PLOT 28,66:DRAW 610,66:DRAW 6
10,14:DRAW 28,14:DRAW 28,66        <295B>
2140 PLOT 275,16:DRAW 275,64:PLOT
430,16:DRAW 430,64                 <2452>
2150 PEN 7:LOCATE 2,22:PRINT CHR$(
244)+CHR$(245)+":":LOCATE 2,23:PRI
NT CHR$(250)+CHR$(239)+": ";sc:LOC
ATE 2,24:PRINT CHR$(238)+CHR$(239)
+":";po$:LOCATE 10,22:PRINT CHR$(2
46)+CHR$(247)+CHR$(248)+CHR$(249):
LOCATE 10,23:PRINT"100%"           <9230>
2160 RESTORE 2170                  <09B7>
2170 DATA 69,32,69,19,62,32,62,19,
55,32                              <241C>
2180 FOR lf=1 TO lif               <1418>
2190 READ oux,ouy                  <13BA>
2200 ship$=ufo2$:/EXC,oux*1,ouy*1,
@ship$,2                           <37CD>
2210 NEXT lf:lf=0                  <138C>
2220 TAG:PLOT-20,-20,4             <11E4>
2230 MOVE 245,397:PRINT CHR$(252)+
CHR$(253)+CHR$(254)+CHR$(253)+CHR$
(255);                             <31A4>
2240 TAGOFF                        <06CE>
2250 WINDOW 2,19,2,20:CLS          <1137>
2260 REM********* SCREEN 1 *******
**                                 <22AC>
2270 f=0:screen=1                  <17B9>
2280 PEN#1,7:LOCATE#1,5,22:PRINT#1
," 1":PEN#1,1                      <2301>
2290 hm=140:GOSUB 5670             <12E2>
2300 FOR c=37 TO 605 STEP 4        <135C>
2310 q=RND*90                      <0FDF>
2320 PLOT c,80,4:DRAWR 0,q         <1758>
2330 PLOT c,80,8:DRAWR-4,q         <18B9>
2340 NEXT c                        <0AAA>
2350 GOSUB 4800                    <097D>
2360 PEN 15:LOCATE 1,10:PRINT CHR$
(139):LOCATE 1,11:PRINT CHR$(142):
PEN 1                              <273B>
2370 IF take=1 THEN GOTO 2380 ELSE
 GOTO 2970                         <1AFD>
2380 PEN 7:LOCATE 14,5:PRINT CHR$(
214)+CHR$(143)+CHR$(215):PEN 8:LOC
ATE 14,6:PRINT STRING$(3,127):PEN
9:LOCATE 14,7:PRINT CHR$(213)+CHR$
(143)+CHR$(212):PEN 10:LOCATE 14,8
:PRINT CHR$(214)+CHR$(143)+CHR$(21
5):PEN 1:LOCATE 14,9:PRINT STRING$
(3,143)                            <88DF>
2390 LOCATE 14,10:PRINT STRING$(3,
143):LOCATE 14,11:PRINT STRING$(3,
143):LOCATE 14,12:PRINT CHR$(213)+
CHR$(143)+CHR$(212)                <43C6>
2400 FOR c=257 TO 285 STEP-4       <1424>
2410 PLOT 450,c,3:DRAW 540,c       <1A72>
2420 NEXT c                        <0A49>
2430 GOTO 2850                     <0937>
2440 REM********* SCREEN 2 *******
**                                 <2245>
2450 screen=2:PEN#1,7:LOCATE#1,5,2
2:PRINT#1," 2":PEN#1,1             <2F90>
2460 mg=18:yb=20:za=1:xb=0         <2691>
2470 PEN 15:LOCATE 1,10:PRINT CHR$
(139):LOCATE 1,11:PRINT CHR$(142)  <2487>
2480 LOCATE 18,1:PRINT CHR$(135):L
OCATE 18,2:PRINT CHR$(141)         <20A8>
2490 FOR c=1 TO 7                  <0E4C>
2500 PEN c+6                       <0CF2>
2510 xb=xb+za:yb=yb-1:mg=mg-za     <3683>
2520 LOCATE xb+1,yb:PRINT STRING$(
mg-xb,127)                         <278C>
2530 IF mg<14 THEN za=0            <169D>
2540 NEXT c                        <0A3A>
2550 PEN 8:LOCATE xb+1,yb-1:PRINT
STRING$(mg-xb,219):PEN 1           <2FD1>
2560 LOCATE 8,14:PRINT CHR$(32)+ST
RING$(2,9)+CHR$(32)                <218C>
2570 GOSUB 4800                    <0936>
2580 GOTO 2850                     <0965>
2590 REM********* SCREEN 3 *******
**                                 <2263>
2600 screen=3:PEN#1,7:LOCATE#1,5,2
2:PRINT#1," 3":PEN#1,1             <2FE9>
2610 PEN 15:LOCATE 18,1:PRINT CHR$
(135):LOCATE 18,2:PRINT CHR$(141)  <2410>
2620 LOCATE 1,18:PRINT CHR$(139):L
OCATE 1,19:PRINT CHR$(142):PEN 1   <2387>
2630 GOSUB 4800:hm=270:GOSUB 5670  <18F5>
2640 yw=280:FOR c=1 TO 40          <19A8>
2650 yw=yw-0.5:xw=INT(RND*100):PLO
T 250,yw,0:DRAWR xw,0              <3CF2>
2660 NEXT c                        <0A2A>
2670 GOTO 2850                     <0918>
2680 REM********* SCREEN 4 *******
**                                 <2246>
2690 screen=4:f=4                  <17B3>
2700 PEN#1,7:LOCATE#1,5,22:PRINT#1
," 4"                              <1D62>
2710 PEN 15:LOCATE 18,17:PRINT CHR
```

```
$(135):LOCATE 18,18:PRINT CHR$(141
):PEN 1:GOSUB 4800.                    (2E78)
2720 FOR c=50 TO 150 STEP 4            (120D)
2730 f=f+1:IF f=13 THEN f=4            (20B9)
2740 PLOT c,80,f:DRAWR 0,70            (18FD)
2750 NEXT c                            (0ADE)
2760 hi1=INT(RND(1)*(-49)+65):hi2=
INT(RND(1)*(91)+70)                    (377D)
2770 sn$=fs$:/EXC,hi1*1,hi2*1,@sn$
,3                                     (31F7)
2780 st$=stadt$:/EXC,35,180,@st$,3
                                       (2809)
2790 IF take=0 THEN 2800 ELSE 2850
                                       (18BA)
2800 w1$=waffe$:/EXC,9,100,@w1$,2      (27ED)
2810 GOTO 2850                         (0931)
2820 REM                               (062A)
2830 REM********** HAUPTPROGRAMM **
*******                                (2780)
2840 REM                               (0652)
2850 ON screen GOTO 2980,3010,3040
,3070                                  (1F41)
2860 ON go GOTO 3130,3880              (13D0)
2870 ship$=ufo2$:ke$=INKEY$            (1DFA)
2880 IF ke$=CHR$(b(3))THEN y=y+3:I
F y>192 THEN y=192                     (340F)
2890 IF ke$=CHR$(b(4))THEN y=y-3:I
F y<50 THEN y=50                       (34F7)
2900 IF ke$=CHR$(b(1))THEN x=x-3:I
F x<4 THEN x=4                         (3207)
2910 IF ke$=CHR$(b(2))THEN x=x+3:s
hip$=ufor$:IF x>70 THEN x=70           (45E4)
2920 IF ke$=CHR$(b(5))THEN ON go G
OTO 3310,3420                          (26C6)
2930 /EXC,x*1,y*1,@ship$,2             (23A6)
2940 CALL &BD19                        (09EF)
2950 /PUT,x*1,y*1,@ship$,1             (2309)
2960 zeit=zeit+1:IF zeit=60 THEN z
eit=0:GOTO 5520                        (31EF)
2970 GOTO 2850                         (0971)
2980 IF x=4 THEN IF MIN(y,122)=MAX
(y,111)THEN GOSUB 3100:x=67:y=185:
GOTO 2450                              (3FB1)
2990 IF take=1 THEN IF MIN(x,63)=M
AX(x,55)THEN IF y<101 THEN GOTO 40
40                                     (36EC)
3000 GOTO 2860                         (09FD)
3010 IF x=70 THEN IF y>183 THEN GO
SUB 3100:x=7:y=113:GOTO 2270           (2FC4)
3020 IF x=4 THEN IF MIN(y,122)=MAX
(y,112)THEN GOSUB 3100:x=67:y=190:
GOTO 2600                              (3FFF)
3030 GOTO 2860                         (093A)
3040 IF x=70 THEN IF y>183 THEN GO
SUB 3100:x=7:y=118:GOTO 2450           (2F23)
3050 IF x=4 THEN IF y<59 THEN GOSU
B 3100:x=67:y=59:go=2:GOTO 2690        (3703)
3060 GOTO 2860                         (0976)
3070 IF x=70 THEN IF MIN(y,68)=MAX
(y,56)THEN GOSUB 3100:x=7:y=56:go=
1:zuf=0:GOTO 2600                      (5006)
3080 IF take=0 THEN IF x=10 THEN I
F MIN(y,118)=MAX(y,109)THEN GOSUB
3250                                   (360D)
3090 GOTO 2860                         (09B3)
3100 zu=4:yu=INT(RND(1)*(60-160+1)
+160):SOUND 1,1062,100,15,7,7,0:FO
R papier=8 TO 13:PAPER papier:CLS:
NEXT papier:PAPER 0:CLS                (69D0)
3110 RETURN                            (067F)
3120 REM********** ANGREIFER/UFOS *
********                               (288F)
3130 ship2$=ufo$                       (1483)
3140 ras=ras XOR 2:IF ras=2 THEN s
hip2$=uff$                             (2E2D)
3150 IF zu<x THEN zu=zu+race           (2442)
3160 IF zu>=x THEN zu=zu-race:IF y
u>y THEN yu=yu-race ELSE IF yu<y T
HEN yu=yu+race                         (654B)
3170 IF zu<4 THEN zu=4                 (155E)
3180 /EXC,zu*1,yu*1,@ship2$,2          (262B)
3190 CALL &BD19                        (09E4)
3200 /PUT,zu*1,yu*1,@ship2$,1          (26CD)
3210 sh=INT(RND*10):IF sh>7 THEN G
OTO 3540                               (22F1)
3220 hagel=INT(RND*50):IF hagel>47
 THEN GOTO 4180                        (29F9)
3230 GOTO 2870                         (091B)
3240 REM********** WAFFE AUFNEHMEN
*********                              (29AB)
3250 SOUND 2,100,30,15,3,3,0           (1690)
3260 /PUT,9,100,@w1$,1                 (18C8)
3270 take=1:geld=150:GOSUB 5470        (1E49)
3280 SOUND 1,70,90,15,3,4,0            (1615)
3290 RETURN                            (06E6)
3300 REM********** FEUERN *********
                                       (2001)
3310 ship$=ufo2$:/EXC,x*1,y*1,@shi
p$,2                                   (33DA)
3320 IF screen<4 THEN ship2$=ufo$:
/EXC,zu*1,yu*1,@ship2$,2               (4391)
3330 xra=x-8:yra=y-3:IF x<=4 OR xr
a<5 THEN/PUT,zu*1,yu*1,@ship2$,1:G
OTO 2940                               (5831)
3340 SOUND 1,150,30,11,3,5,20          (1725)
3350 ra$=rakete$:/EXC,xra*1,yra*1,
@ra$,2                                 (3567)
3360 CALL &BD19:CALL &BD19             (0E0C)
3370 /PUT,xra*1,yra*1,@ra$,1           (253E)
3380 IF x>=zu THEN IF ABS(x-zu)<=1
2 THEN IF ABS(yra-yu)<=6 THEN GOTO
 3440                                  (3C1C)
3390 /PUT,zu*1,yu*1,@ship2$,1          (264A)
3400 GOTO 2940                         (0999)
3410 REM********** ZURUECK BEI FEUE
R *********                            (2B21)
3420 GOTO 2930                         (0971)
3430 REM********** EXPLOSION ******
***                                    (23DD)
3440 utt$=kap$                         (12FD)
3450 SOUND 2,145,255,0,5,5,31          (1668)
3460 /EXC,zu*1,yu*1,@utt$,2            (2447)
3470 FOR p=1 TO 1000:NEXT p            (15F3)
3480 /PUT,zu*1,yu*1,@utt$,1            (24FC)
3490 /PUT,zu*1,yu*1,@ship2$,1          (2611)
3500 geld=25:GOSUB 5470                (1458)
3510 po=po+1:IF po=2 THEN po=0:GOS
UB 4740                                (2893)
3520 zu=4:yu=INT(RND*180):IF yu<55
 THEN GOTO 3520 ELSE GOTO 2940         (3170)
3530 REM********** GEGNERISCHES FEU
ER *********                           (2C2C)
3540 IF zu<=x THEN IF ABS(yu-y)<=4
 THEN GOTO 3560                        (27BA)
3550 GOTO 2870                         (099C)
3560 rak=zu+8:IF rak>69 THEN 2870      (2214)
3570 SOUND 1,162,82,15,8,8,30          (178D)
3580 ship$=ufo2$:/EXC,x*1,y*1,@shi
p$,2                                   (33F7)
3590 ship2$=ufo$:/EXC,zu*1,yu*1,@s
hip2$,2                                (3685)
3600 IF zu>70 THEN 2870                (123D)
3610 fe$=la$:/EXC,rak*1,yu*1,@fe$,
2                                      (3048)
3620 CALL &BD19:CALL &BD19             (0E14)
3630 /PUT,rak*1,yu*1,@fe$,1            (240F)
3640 IF ABS(zu-x)<=12 THEN GOTO 36
80                                     (1C0D)
3650 /PUT,x*1,y*1,@ship$,1             (2384)
3660 GOTO 3190                         (0976)
3670 REM********** SPIEL VERLOREN O
DER WEITERES LEBEN ? *********         (3E87)
3680 SOUND 2,145,255,0,5,5,12          (16E8)
3690 /PUT,x*1,y*1,@ship$,1             (23D4)
3700 utt$=kap$:/EXC,x*1,y*1,@utt$,
2                                      (30C2)
3710 OUT &BC00,8:OUT &BD00,1           (1289)
3720 FOR p=1 TO 1500:NEXT p            (16CE)
3730 OUT &BC00,8:OUT &BD00,0           (12AD)
3740 IF lf=0 THEN RESTORE 2170         (122C)
3750 lf=lf+1:READ oux,ouy              (21DB)
3760 kr1$=kr$:/EXC,oux*1,ouy*1,@kr
1$,1                                   (33DB)
3770 IF lf=lif THEN GOTO 3790 ELSE
 IF screen<4 THEN/PUT,x*1,y*1,@utt
$,1:/PUT,zu*1,yu*1,@ship2$,1:zu=4:
yu=INT(RND(1)*(-139)+190):GOTO 294
0                                      (8B21)
3780 /PUT,x*1,y*1,@utt$,1:/GET,vx*
1,vy*1,@vo$,1:vx=INT(RND(1)*(-60)+
65):vy=INT(RND(1)*(-139)+190):GOTO
 2940                                  (788A)
3790 o=REMAIN(1):SOUND 1,100,0,0,2
,2:PEN 14:LOCATE 6,12:PRINT"GAME O
VER":k=0:WHILE k<201:PLOT 0,k,0:DR
AWR 640,0:PLOT 0,400-k:DRAWR 640,0
:PLOT k,0:DRAWR 0,400:PLOT 640-k,0
:DRAWR 0,400:k=k+2:WEND                (9831)
3800 GOSUB 4390                        (0908)
3810 PEN 10:LOCATE 2,7:PRINT"[1] D
EFINE KEYS"                            (20FE)
3820 LOCATE 2,9:PRINT"[2] START GA
ME."                                   (1C28)
3830 LOCATE 2,11:PRINT"[3] MONEY..
...."                                  (1DE9)
3840 GOSUB 4550                        (0957)
3850 GOSUB 4830                        (0930)
3860 GOSUB 4980:IF fr$=""THEN GOTO
 3860                                  (1855)
3870 REM********** ANGREIFER IM 4.S
CREEN *********                        (2F32)
3880 IF zuf=2 THEN 2870                (12BA)
3890 sp=sp+1:IF sp>1 THEN vo$=vog$
:sp=0 ELSE vo$=vog2$                   (3F6C)
3900 IF zuf=0 THEN vx=INT(RND(1)*(
-49)+55):vy=INT(RND(1)*(131)+50):G
OTO 3920                               (4492)
3910 fl=fl+1:IF fl=5 THEN SOUND 4,
162,82,12,6,6,30:fl=0                  (3682)
3920 IF vx<x THEN vx=vx+flu            (23F8)
3930 IF vx>x THEN vx=vx-flu            (230B)
3940 IF vy<y THEN vy=vy+flu            (23F3)
3950 IF vy>y THEN vy=vy-flu            (23E2)
3960 IF vx<=x THEN IF vx+4)=x THEN
 IF vy<=y THEN IF vy+4)=y THEN 368
0                                      (3C06)
3970 /EXC,vx*1,vy*1,@vo$,2             (2305)
3980 CALL &BD19                        (0914)
3990 /PUT,vx*1,vy*1,@vo$,1             (2303)
4000 zuf=1                             (0DA7)
4010 IF ABS(vx-hi1)<=5 THEN IF ABS
(vy-hi2)<=5 THEN GOSUB 4330            (3132)
4020 GOTO 2870                         (094A)
4030 REM********** ZERSTOERUNG SOLO
N'S *********                          (2D65)
4040 o=REMAIN(1):SOUND 1,100,220,1
5,1,4,0:FOR p=1 TO 2000:NEXT p         (33A7)
4050 CLS:FOR w=1 TO 30                 (1156)
4060 PLOT INT(RND(1)*(-569)+605),I
NT(RND(1)*(221)+150),INT(RND*13):N
EXT w                                  (3C7E)
4070 st$=stadt$:/EXC,35,115,@st$,2
                                       (2833)
4080 FOR p=1 TO 2500:NEXT p            (1674)
4090 GOSUB 4660:/PUT,35,115,@st$,1
                                       (1EAB)
4100 utt$=kap$:/EXC,37,110,@utt$,3
:FOR p=1 TO 1000:NEXT p:/PUT,37,11
0,@utt$,1:FOR p=1 TO 1500:NEXT p       (62B1)
4110 BORDER 0:OUT &BC00,6:dr=25:WH
ILE dr>0:dr=dr-1:OUT &BD00,dr:CALL
 &BD19:FOR p=1 TO 40:NEXT p:WEND:P
EN 8:CLS                               (567F)
4120 IF sc>=0 AND sc<=180 THEN bg$
="ELITE D"ELSE IF sc>180 AND sc<=2
80 THEN bg$="ELITE C"ELSE IF sc>28
0 AND sc<=380 THEN bg$="ELITE B"EL
SE IF sc>380 THEN bg$="ELITE A"        (8D27)
4130 LOCATE 7,2:PRINT"URKUNDE":LOC
ATE 7,3:PRINT STRING$(7,45):LOCATE
 5,6:PRINT"SIE WERDEN":LOCATE 3,8:
PRINT"BEFOERDERT ZUM":LOCATE 7,10:
PRINT bg$:LOCATE 6,12:PRINT"COMMAN
DER"                                   (6D72)
4140 OUT &BC00,6:WHILE dr<25:dr=dr
+1:OUT &BD00,dr:CALL &BD19:FOR p=1
 TO 40:NEXT p:WEND:GOSUB 4580:FOR
p=1 TO 5000:NEXT p:OUT &BC00,6:OUT
 &BD00,25:GOTO 3800                    (71C1)
4150 REM                               (0693)
4160 REM********** ASTEROIDEN *****
****                                   (2414)
4170 REM                               (0633)
4180 hax=INT(RND(1)*(-34)+65)          (1EC8)
4190 IF ABS(hax-x)<=4 THEN IF zz=0
 THEN ko=1:dis=y:GOTO 4210             (3987)
4200 dis=55:ko=0                       (169F)
4210 ship$=ufo2$:/EXC,x*1,y*1,@shi
```

```
p$,2                                    (33E7)
4220 ship2$=ufo$:/EXC,xu*1,yu*1,@s
hip2$,2                                 (3672)
4230 SOUND 1,100,50,15,1,5,0:hay=1
90+stp                                  (27B9)
4240 WHILE hay>dis:hay=hay-9            (23ED)
4250 met$=ast$:/EXC,hax*1,hay*1,@m
et$,3                                   (342F)
4260 CALL &BD19:/PUT,hax*1,hay*1,@
met$,1                                  (2567)
4270 WEND                               (06C2)
4280 IF ko=1 THEN 3680                  (11F9)
4290 /PUT,x*1,y*1,@ship$,1              (23E7)
4300 /PUT,xu*1,yu*1,@ship2$,1           (2668)
4310 GOTO 2940                          (09B9)
4320 REM********** FLUEGLER ZERSTOE
RT *********                            (2C9D)
4330 /PUT,ux*1,uy*1,@uo$,1              (23AD)
4340 SOUND 1,10,100,10,9,9,1:SOUND
 2,100,250,15,9,8,10                    (2943)
4350 OUT &BC00,3:FOR e=1 TO 30:b=I
NT(RND*10):SPEED INK b+1,b+1:BORDE
R b+16,0:INK 0,b+16,0:OUT &BD00,b:
NEXT e:OUT &BC00,3:OUT &BD00,7:BOR
DER 0:INK 0,0:SPEED INK 10,10           (7B45)
4360 geld=100:GOSUB 5470                (143A)
4370 zuf=2:RETURN                       (0FD5)
4380 REM********** UP TITELBILD ***
******                                  (26D4)
4390 OUT &BC00,6:OUT &BD00,0            (12E3)
4400 MODE 0:CALL &BC02:INK 0,1:BOR
DER 1:PEN 1:PAPER 0:INK 15,14,26        (24C6)
4410 f=5:za=0.11                        (18EE)
4420 LOCATE 1,25:PRINT"SOLON"           (1353)
4430 FOR x0=4 TO 155 STEP 4             (121D)
4440 FOR y0=2 TO 14 STEP 2              (1233)
4450 IF TEST(x0,y0)=1 THEN GOSUB 4
480                                     (1C27)
4460 NEXT y0,x0                         (114D)
4470 GOTO 4520                          (0973)
4480 f=f+za:IF f>13 THEN za=-0.13       (2B56)
4490 MOVE x0+160*1.5,y0*5+320           (21F7)
4500 DRAWR 0,-7,C1NT(f)                 (135D)
4510 RETURN                             (0673)
4520 LOCATE 1,25:PRINT SPACE$(5):I
NK 1,26                                 (172B)
4530 WINDOW 2,19,7,24                   (0F7F)
4540 RETURN                             (06AF)
4550 WINDOW#1,1,20,25,25:PEN#1,1:L
OCATE#1,4,25:PRINT#1,"- COBRA-SOFT
 -"                                     (3797)
4560 OUT &BC00,6:OUT &BD00,25:OUT
&BC00,1:FOR o=1 TO 40:OUT &BD00,o:
CALL &BD19:FOR p=1 TO 20:NEXT p,o:
RETURN                                  (4CE5)
4570 REM********** MELODIE ********
*                                       (2125)
4580 RESTORE 4620:FOR n=1 TO 30         (14BA)
4590 READ s                             (0A4B)
4600 SOUND 1,s,30,5,2,1,0:SOUND 2,
1000+s,30,4,2,1,0                       (2E42)
4610 NEXT n                             (0A9B)
4620 DATA 100,89,100,113,119,113,1
00,0,127,119,113,0,119,113,100,0        (3FEC)
4630 DATA 100,89,100,113,119,113,1
00,0,127,0,100,0,119,150                (37B7)
4640 RETURN                             (0679)
4650 REM********** UP - ZERSTOERUNG
 *********                              (2AEB)
4660 SOUND 1,29,250,15,10,10,31         (1936)
4670 FOR f=3 TO 0 STEP-3                (117D)
4680 r=11:a=80:WHILE a<203              (1D45)
4690 r=r-0.3                            (152C)
4700 PLOT a*1.5,a,f:DRAWR-r,r:PLOT
 640-a*1.5,a:DRAWR r,r                  (4872)
4710 a=a+3:WEND:SOUND 1,145,50,0,5
,5,31:OUT &BC00,8:OUT &BD00,1:NEXT
 f                                      (38A7)
4720 SOUND 2,100,270,15,1,1,20:OUT
 &BC00,8:OUT &BD00,0:RETURN             (2873)
4730 REM********** POWER-ANZEIGE **
*******                                 (2724)
4740 zz=1:po$=" "+CHR$(239)+CHR$(2
51):SOUND 2,29,100,15,10,9,20:GOSU
B 4780                                  (3DE0)
4750 AFTER 1000,1 GOSUB 4770            (0F4F)
4760 RETURN                             (0668)
4770 zz=0:po$=CHR$(239)+STRING$(2,
246):GOSUB 4780:RETURN                  (292C)
4780 PEN#1,9:LOCATE#1,5,24:PRINT#1
,po$:PEN#1,1:RETURN                     (25E6)
4790 REM********** STERNE *********
                                        (2084)
4800 FOR st=1 TO 30:PLOT INT(RND(1
)*(40-600+1)+600),INT(RND(1)*(200-
376+1)+376),INT(RND*13)                 (4D47)
4810 NEXT st:RETURN                     (0DF0)
4820 REM********** TITELMELODIE ))
OH,SUSANNA (( *********                 (378C)
4830 RESTORE 4890:musik=0               (146E)
4840 musik=musik+1:GOSUB 4980           (1DEC)
4850 READ ton,dauer                     (1569)
4860 IF ton=-1 THEN RETURN              (1123)
4870 SOUND 1,ton,dauer,15,2,1,0:SO
UND 2,ton*2,dauer,14,5,1,0              (3E68)
4880 GOTO 4840                          (09B0)
4890 DATA 506,30,478,30,426,30,358
,30,0,1,358,30,319,30,358,30,426,3
0                                       (42F9)
4900 DATA 506,30,478,30,426,30,0,1
,426,30,478,30,506,30,478,60,506,3
0                                       (4283)
4910 DATA 478,30,426,30,358,30,0,1
,358,30,319,30,358,30,426,30,506,3
0                                       (428D)
4920 DATA 478,30,426,30,0,1,426,30
,478,30,0,1,478,30,506,60,358,30,0
,1                                      (4361)
4930 DATA 358,30,284,25,0,1,284,50
,284,30,319,30,0,1,319,30,379,30,4
78                                      (4398)
4940 DATA 30,426,60,478,30,426,30,
379,30,319,30,0,1,319,30,284,30,31
9                                       (427B)
4950 DATA 30,379,30,478,30,426,30,
379,30,0,1,379,30,426,30,0,1,426,3
0,478,150                               (4A1F)
4960 DATA -1,-1                         (0CC2)
4970 REM********** MENUE ********* (1F5F)
4980 fr$=INKEY$:IF fr$=""THEN RETU
RN                                      (1949)
4990 IF fr$="1"THEN CLS:CLEAR:GOTO
 5170                                   (183C)
5000 IF fr$="2"THEN CLS:LOCATE 4,1
0:PRINT"PLEASE WAIT":GOTO 530           (2BA3)
5010 IF fr$="3"THEN GOTO 5040           (14ED)
5020 fr$="":RETURN                      (0F10)
5030 REM********** UP MONEY *******
**                                      (2298)
5040 INK 0,0:BORDER 0                   (0C3D)
5050 MODE 1:PEN 1                       (0AF5)
5060 OUT &BC00,1:OUT &BD00,1            (120C)
5070 OUT &BC00,6:OUT &BD00,1            (1239)
5080 PLOT 120,280,1                     (0EAD)
5090 DRAWR 0,100:DRAWR 20,-10:DRAW
R 20,-60:DRAWR 120,25:DRAWR 40,-50
:DRAWR-4,-2:DRAWR-15,20:DRAWR-4,-2
:DRAWR-15,15:DRAWR-60,-35:DRAWR 20
,-70:DRAWR-50,-35:DRAWR-60,50:DRAW
R-120,-60:DRAWR 20,-20:DRAWR-4,-2:
DRAWR 20,-20:DRAWR-4,-2:DRAWR-60,6
0                                       (987A)
5100 DRAWR 117,100:DRAWR 60,-30:DR
AWR-40,-25:DRAWR 50,-30:DRAWR 40,2
2:DRAWR-10,50:DRAWR-40,-15:DRAWR-1
8,55:PLOTR 60,-30:DRAWR 2,-6:DRAWR
 60,42:PLOTR-42,-105:DRAWR 0,-4:DR
AWR-50,-34:DRAWR-60,40:DRAWR 0,12:
PLOTR 0,-12:DRAWR-96,-48                (8F2C)
5110 PEN 2:LOCATE 14,18:PRINT"1 ZE
RST. FLUEGLER  = 100":LOCATE 14,20
:PRINT"1 ZERST. UFO       = 25":L
OCATE 14,22:PRINT"AUFGENOMMENE WAF
FE = 150"                               (70E1)
5120 PEN 3:LOCATE 14,19:PRINT"1 ZE
RST. FLUEGLER  = 100":LOCATE 14,21
:PRINT"1 ZERST. UFO       = 25":L
OCATE 14,23:PRINT"AUFGENOMMENE WAF
FE = 150"                               (7091)
5130 OUT &BC00,1:OUT &BD00,40           (13F9)
5140 OUT &BC00,6:OUT &BD00,25           (1305)
5150 FOR i=2 TO 3:INK i,11:CALL &B
D19:FOR p=1 TO 500:NEXT p:INK i,0:
NEXT i:a$=INKEY$:IF a$=""THEN 5150
                                        (519D)
5160 GOTO 3800                          (0951)
5170 REM********** DEFINE KEYS ****
*****                                   (2536)
5180 ug=0:zif=0:PEN 1:LOCATE 4,3:P
RINT"DEFINE KEYS":PEN 15                (30DE)
5190 PLOT 0,0,1:DRAW 639,0:DRAW 63
9,399:DRAW 0,399:DRAW 0,0               (2735)
5200 DATA 32,13,127,9,8,11,10,16,2
24                                      (21F9)
5210 RESTORE 5200                       (0974)
5220 te$(1)="SPACE":te$(2)="ENTER"
:te$(3)="DEL":te$(4)="JOY-RIGHT"        (4A99)
5230 te$(5)="JOY-LEFT":te$(6)="JOY
-UP":te$(7)="JOY-DOWN":te$(8)="CLR
":te$(9)="COPY"                         (5D1B)
5240 FOR dat1=1 TO 9:READ co:ac(da
t1)=co:NEXT dat1                        (3687)
5250 me=1:wort$="LEFT RIGHTUP   DO
WN FIRE"                                (2F24)
5260 FOR z=1 TO 5                       (0EB8)
5270 b$=MID$(wort$,me,5)                (1CD2)
5280 LOCATE(20-LEN(b$))/2,7:PRINT
b$+SPACE$(1)                            (2376)
5290 tas$=INKEY$:IF tas$=""THEN 52
90                                      (1D84)
5300 b(z)=ASC(tas$)                     (1A2B)
5310 ug=ug+1                            (1207)
5320 READ dat                           (0CFE)
5330 IF b(z)=b(dat)THEN CLEAR:SOUN
D 1,100,100,15,7,7,0:GOTO 5210          (3639)
5340 IF ug<>4 THEN 5310 ELSE ug=0       (1A9E)
5350 SOUND 1,250,30,15,3,3,0            (16E1)
5360 zif=zif+1                          (14A1)
5370 IF b(z)=ac(zif)THEN LOCATE(20
-LEN(te$(zif)))/2,9:PRINT te$(zif)
:zif=0:GOTO 5400                        (57CF)
5380 IF zif<>9 THEN 5362 ELSE zif=
0                                       (1C39)
5390 LOCATE(20-LEN(CHR$(b(z))))/2,
9:PRINT CHR$(b(z))                      (31A8)
5400 FOR p=1 TO 1000:NEXT p             (1610)
5410 LOCATE 2,9:PRINT SPACE$(17)        (11C6)
5420 me=me+5                            (1282)
5430 NEXT z:CLS                         (0C86)
5440 RESTORE:GOTO 3800                  (0BA6)
5450 DATA 2,3,4,5,1,3,4,5,1,2,4,5,
1,2,3,5,1,2,3,4                         (2E4B)
5460 REM********* PUNKTE ZAEHLEN U.
 AUSGEBEN *********                     (33A7)
5470 sc=sc+geld                         (186B)
5480 IF LEN(STR$(sc))>3 THEN h=1 E
LSE h=0                                 (24DF)
5490 PEN#1,8:LOCATE#1,5,23:PRINT#1
,RIGHT$(STR$(sc),LEN(STR$(sc))-h)       (38C1)
5500 RETURN                             (0633)
5510 REM********** FUEL *********      (1EDC)
5520 fu=fu-10                           (13D8)
5530 SOUND 3,300,150,15,2,3,0:SOUN
D 4,100,150,15,2,3,0                    (2943)
5540 fu$=RIGHT$(STR$(fu)+"% ",4)        (1FA9)
5550 IF fu<=30 THEN PEN#1,14 ELSE
PEN#1,7                                 (1C64)
5560 LOCATE#1,10,23:PRINT#1,fu$         (18BC)
5570 IF fu=0 THEN GOTO 5600             (12BF)
5580 GOTO 2850                          (09DF)
5590 REM********** TANK LEER ******
***                                     (23B5)
5600 IF screen<4 THEN ship2$=ufo$:
/EXC,xu*1,yu*1,@ship2$,2 ELSE IF z
uf<>2 THEN uo$=uog$:/EXC,ux*1,uy*1
,@uo$,2                                 (7AFF)
5610 SOUND 2,145,255,0,5,5,12:SOUN
D 1,180,255,0,5,5,12                    (28CA)
5620 utt$=kap$:/EXC,x*1,y*1,@utt$,
3                                       (30CE)
5630 FOR p=1 TO 2000:NEXT p             (16F0)
5640 /PUT,x*1,y*1,@utt$,1               (2218)
5650 GOTO 3790                          (09D6)
5660 REM********** MOND *********      (1E3B)
5670 DEG                                (06A8)
5680 FOR c=1 TO 180 STEP 4              (1126)
5690 PLOT 320+50*COS(c),hm+50*SIN(
c),4                                    (29A2)
5700 DRAWR 0,-100*SIN(c)                (148F)
5710 NEXT c                             (0A8B)
5720 RETURN                             (06ED)
```

Jetzt schlägt's dreizehn!

CPC-WELT kommt direkt zu Ihnen ins Haus! 7x pro Jahr! Für nur 70 DM! Sie sparen über 30 DM!





Finden Sie Ihre CPC-WELT nicht immer am Kiosk? Vielleicht, weil schon ausverkauft? Möchten Sie die CPC-WELT schon vor der Kioskbelieferung in Händen haben? Dann gibt es jetzt die Möglichkeit! Wir beliefern Sie im Abonnement mit sechs plus einer Ausgabe für ganze 70 DM (Inland) oder 80 DM (Ausland). Sie erhalten dann das jeweils druckfrische Heft, in der Regel sogar früher, als es am Kiosk hängt (so die Bundespost will). Sechsmal. Und außerdem gehört zum Abo noch unser jährlicher CPC-SPECIAL Sammelband im Wert von DM 14,80. Einzige Bedingung: Das Abo muß zum Zeitpunkt des Erscheinens dieses Bandes noch bestehen und bezahlt sein. Ist das ein Angebot?

WICHTIGE RECHTLICHE GARANTIE!

Sie können diesen Abo-Auftrag binnen einer Woche nach Zugang der Abo-Bestätigung durch den Verlag an Sie widerrufen. Postkarte genügt! Zur Wahrung der Frist genügt die rechtzeitige Absendung. Bitte, bestätigen Sie durch Ihre zweite Unterschrift, daß Sie von diesem Widerspruchsrecht Kenntnis genommen haben. Sie können dieses Abonnement jeweils mit einer Frist von einem Monat zum Ende des Bezugszeitraumes (sechs Hefte) kündigen. Unterlassen Sie diese Kündigung, wird die Belieferung mit weiteren sechs Heften zu den gleichen Bedingungen fortgesetzt! Die Lieferung beginnt nach Eingang der Abo-Gebühr.

ABO-SERVICE-KARTE

COUPON

Ich nehme zur Kenntnis, daß die Belieferung erst beginnt, wenn die Abo-Gebühr dem Verlag zugegangen ist.

Ja, ich möchte von Ihrem Angebot Gebrauch machen.
Bitte senden Sie mir bis auf Widerruf ab sofort jeweils die nächsten sechs (+ 1) Ausgaben an untenstehende Anschrift. Wenn ich nicht einen Monat vor Ablauf kündige, läuft diese Abmachung automatisch weiter.

Name _____
Vorname _____
Straße/Hausnr. _____
PLZ/Ort _____
Ich bezahle: _____
☐ per beiligendem Verrechnungsscheck
☐ gegen Rechnung
☐ bargeldlos per Bankeinzug von meinem Konto
bei (Bank) und Ort _____
Kontonummer _____
Bankleitzahl _____
(steht auf jedem Kontoauszug)

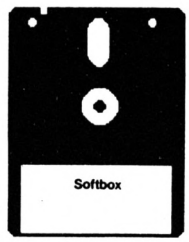
Unterschrift _____
Von meinem Widerspruchsrecht habe ich Kenntnis genommen.

Unterschrift _____ SPI/10

CPC-WELT
ABO-SERVICE SPI/10
POSTFACH 1161
D-8044
UNETSCHLEISSHEIM

PROGRAMMSERVICE
Achtung! Preis-Senkung!!!

Hiermit bestelle ich in Kenntnis Ihrer Verkaufsbedingungen die Listings dieses Heftes auf



☐ Diskette (30 DM)
Name _____
Vorname _____
Straße/Hausnr. _____
PLZ/Ort _____
Ich bezahle:
☐ per beiliegendem Verrechnungsscheck / Bargeld
☐ bargeldlos per Bankeinzug von meinem Konto (nur möglich in der Bundesrepublik!)
bei (Bank) und Ort _____
Kontonummer _____
Bankleitzahl _____
(steht auf jedem Kontoauszug)
Verkaufsbedingungen: Lieferung nur gegen Vorkasse oder Bankabbuchung. Keine Nachnahme.
Umtausch bei Nichtfunktionieren.

Unterschrift _____ SPI/10
Bitte ausschneiden und einsenden an

CPC-WELT
SOFTWARE-SERVICE SPI/10
POSTFACH 1161
D-8044 UNTERSCHLEISSHEIM

Jede Diskette nur noch 30 DM! Sie sparen 10 DM

LOAD&RUN



Das Spiele-Magazin

Hallo, liebe Leser,

Die Redaktion von Load & Run wurde mal wieder vom Fieber befallen. Kaum waren die Epidemien Shanghai und Tetris überstanden, machte sich eine neue Krankheit breit, der sich keiner der Redakteure erwehren konnte. Ich spreche vom neuesten Amiga-Game: ‚Ports of Call'. Dieser Handelsimulation ist für kurze Zeit sogar unser Verleger zum Opfer gefallen. Leider hat diese Krankheit aber auch einen Haken: Ist ein Spieler lange genug infiziert, so verflüchtigt sie sich wieder mit einem Schlag. Nach langem Spiel deckte unser Verleger einige sehr bedeutende Schwächen und Schlampereien in der Programmierung auf, die den Spielwitz erheblich senken oder gar völlig verschwinden lassen. So wurde durch schlampige Arbeit einmal mehr ein hervorragendes Spiel kaputtgemacht. Doch mehr dazu im ausführlichen Testbericht in dieser Ausgabe.

Ansonsten finden Sie in dieser Ausgabe wieder alles wie gewohnt vor. Ausführliche Spieletests berichten über die neuesten Games, Kurzberichte behandeln die Umsetzungen und Neuerscheinungen auf dem Spielemarkt, und die Player's Pages geben Hilfestellung für jeden verzweifelten Joystick-Akrobaten. Ich hoffe, es ist für jeden etwas dabei.

Viel Spaß beim Lesen wünscht Ihnen die Redaktion

Impressum

LOAD & RUN erscheint in der München Aktuell Verlags GmbH, Heßstraße 90, 8000 München 40.

Verantwortlich für den Inhalt:
Gert Seidel

Redaktion:
Thomas Bosch, Michael Nebauer.

Layout:
Sonja Anderle

Anzeigenverwaltung:
ADV-Mediendienste, Aindlingerstr. 8900 Augsburg, Tel. 0821/790 42 43

Inhalt

Amiga-Darts
Ein Spiel, bei dem Sie ins Schwarze treffen sollen.
Seite III

Rockford is still alive
Die neue Boulder-Dash-Variante, jetzt auch für den Amiga.
Seite III

Alte Idee – gutes Spiel
Beyond the Ice Palace, ein Hüpf- und Ballerspiel der Extraklasse.
Seite IV

Feuer frei
Bei Gunshoot ist schnelle Reaktion angesagt.
Seite IV

Vorsicht: Suchtgefahr
Powerstyx, eine neue Version des Klassikers Styx, sorgt per Amiga für lange Nächte.
Seite V

Ports of Call
Ein Wirtschafts-Spiel mit versteckten Mängeln.
Seite VI

Freistoß für Amiga
Euro Soccer '88 – eine Fußball-Simulation für den Amiga.
Seite VII

Die Wargames-Edition
Strategische Spiele – kritisch unter die Lupe genommen.
Seite VIII

Horrortrip als Comicstrip
Ooze – ein Grafik-Adventure aus der Gruselkiste.
Seite X

Mörderischer Auftrag
Lösen Sie das Rätsel des Bermuda-Dreiecks.
Seite XI

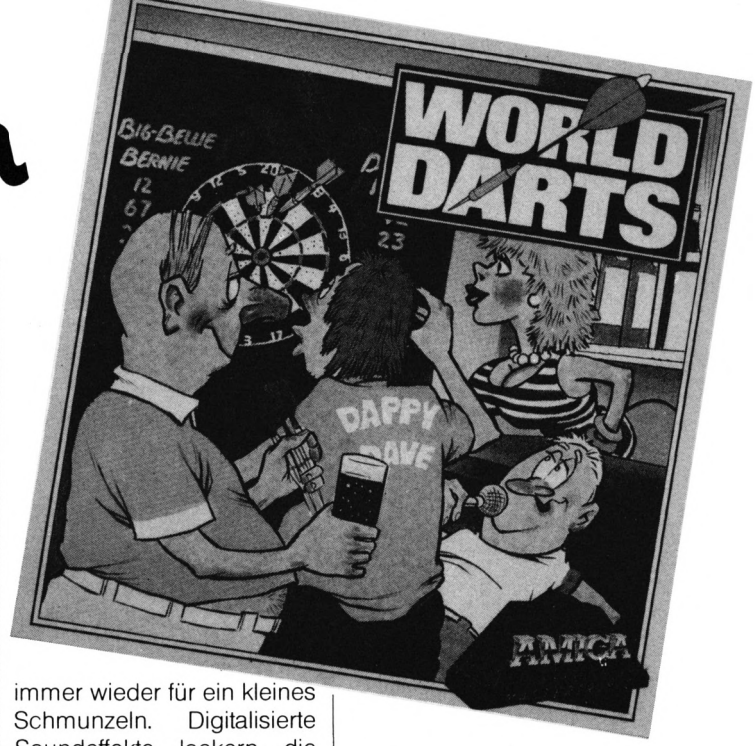
Games für kleine Geldbeutel
Die Welle der Billig-Spiele reißt nicht ab: acht neue Kassetten im Test.
Seite XII

Kurzberichte
Aktuelle Infos über Umsetzungen und Neuerscheinungen für Amiga, Atari und den C16.
Seite XIV

Player's Pages
Tips & Tricks zum besseren Spielen.
Seite XVI

Darts auf dem Amiga

Jetzt brauchen auch die Amiga-Besitzer nicht länger zu trauern. Mastertronic bringt in diesen Tagen eine Darts-Simulation auf den Markt.



Beim Darts-Spiel müssen Sie durch geschicktes Pfeilwerfen, im Volksmund auch „Spikkern" genannt, Ihren Punkteabstand so schnell wie möglich auf Null bringen.
Bis zu zwei Spieler können an der Darts-Simulation von Mastertronic teilnehmen. Wenn man mit Werfen an der Reihe ist, erscheint auf dem Bildschirm die Darts-Scheibe und eine Hand, welche einen Pfeil hält und leider ständig hin und her wackelt (fast wie in der Realität). Mit dem Joystick müssen Sie die Hand in die richtige Position bringen und anschließend mit dem Feuerknopf den Pfeil werfen. Danach schreibt eine weitere Hand Ihren aktuellen Punktestand auf die nebenstehende Tafel. Sieger ist derjenige Spieler, der zuerst bei Null angelangt ist.
Grafisch ist World Darts sehr gut gemacht. Viele kleine Details wie zum Beispiel wechselnde Gesichtszüge einer passiven Spielfigur sorgen immer wieder für ein kleines Schmunzeln. Digitalisierte Soundeffekte lockern die Atmosphäre zusätzlich auf. Wer sich für den Darts-Sport begeistert und zudem noch einen Amiga mit mindestens 512 Kbyte RAM und einen Farbmonitor besitzt, kann sich World Darts ja mal ansehen. TB ■

Titel:	**Darts**
Getestet:	**Amiga**
Umsetzungen:	---

Im Test: ☒ Diskette ☐ Kassette

Preis (DM): k.A.

☒ Joystick ☐ Tastatur ☐ Maus

Wertung	0	25	50	75	100
Grafik					
Sound					
Bedienung					
Motivation					

Rockford is still alive

Die Boulder-Dash-Reihe reißt nicht ab. Auch für den Amiga gibt es jetzt ein solches Game.

Eine hervorragende Adaption der beliebten Boulder-Dash-Reihe liegt jetzt auch für den Amiga vor. Das Spielprinzip ist dasselbe geblieben. Rockford besitzt aber im Gegensatz zu seinen Vorgängern viele neue Features, die das Spielgeschehen auflockern. So können Sie sich zu Beginn des Spieles eine von fünf Rockford-Figuren aussuchen. Je nach Spielfigur ändert sich der Aufbau des Spielfeldes. Haben Sie sich beispielsweise für Rockford, den Meisterkoch entschieden, finden sie sich in einem Chaos aus Bestecken, Tellern und Tischen wieder. Glauben Sie aber nicht, daß sich am Spiel etwas ändert, denn auch hier müssen Sie Objekte sammeln und dabei aufpassen, daß Sie nicht von herunterfallenden Suppenschüsseln zerschmettert werden.
Jedes Spielfeld ist in vier Level unterteilt, die mit steigender Nummer schwieriger werden. Unter anderem nimmt die zur Verfügung stehende Zeit immer schneller ab.
Rockford ist grafisch sehr gut gelungen. Auch über mangelnde Sounduntermalung kann sich der Spieler nicht beklagen. Das Spiel kann ich jedem empfehlen, der einen Amiga mit Farbmonitor besitzt und noch nicht genug von der Boulder-Dash-Welle hat. TB ■

Anzeige

Der clevere Kontakt:

Immer die neueste Software zu absolut coolen Preisen! Testen Sie uns noch heute, wir sind jederzeit für Sie da!



SOFTWARE VERSAND
Andreas Bachler
Postfach 429
D-4290 Bocholt
Tel. (0 28 71) 18 30 88

Alte Idee – gutes Spiel

Elite, der Spezialist in Sachen Spielautomaten-Umsetzungen, hat wieder zugeschlagen. Ein gut aufgemachtes Ballerspiel für den ST ist das Ergebnis.



Lange Zeit war es still um den englischen Softwarehersteller Elite-Systems Limited, doch jetzt endlich gibt es ein neues Spiel, erhältlich für den C 64 und den ST. Die Spielidee kann man nicht mehr als alt bezeichnen, uralt trifft wohl eher zu: Ein unbekanntes Land wird von den Mächten des Bösen unterdrückt. Nach heftigen Diskussionen beschließt man, einen einsamen Helden, der zufällig über neun Leben verfügt, hinab in die Dungeons zu schicken, damit er den Zauberern und ihren Monstergehilfen ein für allemal den Garaus macht: Es darf geballert werden.

Trotz der etwas einfallslosen Spielidee dürfen Sie aber nicht glauben, daß auch das Game selbst verschrottet gehöre – im Gegenteil! Beyond The Ice Palace ist ein gutes bis sehr gutes Jump-and-run-Spiel mit hervorragender Grafik und zahlreichen Bildschirmen. Ein für ST-Verhältnisse sehr ordentliches Scrolling und gelungen animierte Sprites sorgen für einen flüssigen Spielablauf. Auch die passenden Soundeffekte lockern die Atmosphäre stark auf.

Auf dem Weg durch die Dungeons sind neben den Gegnern auch zahlreiche Hindernisse wie tiefe Schluchten oder steile Berghänge zu überwinden. Die neun Leben Ihrer Spielfigur sind wirklich nicht zu viel.

Beyond The Ice Palace ist für Freunde und Liebhaber der Jump-and-run-Spiele ein absolutes Muß. Wer noch nicht allzu viele davon in einer Sammlung hat, sollte sich das Game so schnell wie möglich zulegen. Die Atari-ST-Version benötigt übrigens ROM-TOS und einen Farbmonitor oder TV-Modulator.

TB ■

Titel:	**Beyond The Ice Palace**
Getestet:	**Atari ST**
Umsetzungen:	---

Im Test: ☒ Diskette ☐ Kassette

Preis (DM): 59,-

☒ Joystick ☐ Tastatur ☐ Maus

Wertung	0	25	50	75	100
Grafik					
Sound					
Bedienung					
Motivation					

Feuer frei

Gute Nachrichten für alle Amiga-Besitzer: Wer vor einem Jahr noch mit neidischen Blicken auf Westbank für den CPC geschaut hat, darf dieses Game jetzt auch auf seinem Amiga spielen. Gunshoot ist da.

Axxion hat sich der Umsetzung von Westbank angenommen. Und weil die Amiga-Version noch ein paar Features mehr bot als die des CPC, hat man ihr gleich einen neuen Namen verpaßt: Gunshoot, was übersetzt etwa Revolverschuß heißt. Für Nicht-Kenner von Westbank hier kurz die Handlung:

Sie sind ein kleiner Bankangestellter und möchten gerne ein Held sein. Deswegen kommt Ihnen auch der Umstand, daß in letzter Zeit ziemlich viele Banken überfallen wurden, gerade recht. Mit einem 45er Revolver bewaffnet, warten Sie hinter Ihrem Schalter auf Banditen. Darum herum sind zwölf Türen, die Sie ständig im Auge behalten müssen. Von Zeit zu Zeit öffnen sich eine oder zwei und Personen betreten die Bank. Sind diese aber in der Absicht gekommen, ihr Vermögen anzulegen oder wollen sie gar die Bank um ein paar Tausender erleichtern?

Keine Chance für Bankräuber

Beobachten Sie die Leute genau. Es kann sehr schnell passieren, daß plötzlich einer seinen Colt zieht. Manche kommen sogar schon be-



waffnet herein. Sie müssen augenblicklich reagieren, wenn Sie einen Überfall vereiteln wollen. Dazu bewegen Sie mit dem Joystick ein Fadenkreuz auf die jeweilige Person. Mit dem Feuerknopf wird der Abzug betätigt. Haben Sie einen Räuber erwischt, gibt's Punkte en masse. Wenn Sie aber auf einen ehrlichen Kunden ballern, gibt's jede Menge Ärger mit dem Sheriff. Also genau hinsehen, bevor Sie abdrücken.

Gunshoot ist ein Game, das jedem zu empfehlen ist. Die ohnehin relativ hohe Spielmotivation wird durch hervorragende Grafik und digitalisierte Soundeffekte noch gesteigert. Und wenn's trotzdem mal zu langweilig werden sollte, bietet das Spiel noch einen speziellen Zwei-Player-Mode, in dem Spieler eins das Fadenkreuz steuert und sein Partner den Abzug betätigt. Gunshoot gibt's für alle Amigas ab 512 KByte RAM mit Farbmonitor. TB ■

Titel: Gunshoot
Getestet: Amiga
Umsetzungen: ---

Im Test:

Preis (DM): k.A.

- [X] Joystick
- [ ] Tastatur
- [ ] Maus

Wertung	0	25	50	75	100
Grafik					
Sound					
Bedienung					
Motivation					

Vorsicht: Suchtgefahr!

Ein einfaches Puzzle-Spiel – das suggeriert die Aufmachung von Powerstyx. Doch hinter diesem Namen verbirgt sich sehr viel mehr.

In Powerstyx müssen Sie verschiedene Bilder freilegen, die zu Beginn noch unsichtbar sind. Mit einem Cursor, der nur mit dem Joystick gesteuert werden kann, schneiden Sie aus dem Bildschirm beliebig große Rechtecke heraus. Dadurch wird ein Teil des verdeckten Bildes sichtbar, der jeweils so groß ist wie das ausgeschnittene Rechteck. Sind mehr als 75 Prozent des Bildes freigelegt, gelangen Sie in den nächsten Level. Es gibt 15 Bilder zu entdecken, jedes repräsentiert einen Level.

Mit dem Cursor Rechtecke ausschneiden – das klingt doch wirklich nicht schwierig. Und tatsächlich weckt das Game zunächst keine große Begeisterung. Mißmutig bewege ich meinen Cursor über den Bildschirm. Hier ein Rechteck, da ein Rechteck. Doch was ist das? Aus der linken oberen Ecke tauchen plötzlich zwei Totenköpfe auf, die meinen Linien folgen. Da die beiden nicht gerade vertrauenerweckend aussehen, entschließe ich mich zur Flucht nach hinten. Doch diverse fliegende Buchstaben versperren mir den Weg. Nach wenigen Sekunden gebe ich den Kampf auf – und bin eines meiner fünf Leben los.

Der zweite Anlauf gelingt endlich. Nach knapp vier Minuten, die zu 80 Prozent aus Flucht vor diversen bösartigen Sprites bestehen, ist das erste Bild freigelegt oder besser gesagt: 75 Prozent davon. Das Bild wird komplett aufgebaut und ich kann mich knappe 30 Sekunden lang auf die wirklich einmalig schöne Grafik konzentrieren. Noch bevor ich mich richtig sattgesehen habe, ertönt ein akustisches Warnsignal und der Bildschirm wird wieder schwarz, was den nächsten Level ankündigt. Nun gut, ich habe den ersten in nur vier Minuten überstanden, da wirft mich der zweite auch nicht um – denkste! Die fliegenden Totenschädel haben Verstärkung bekommen und auch die restlichen Sprites scheinen wesentlich angriffslustiger. Nach drei Minuten muß ich kapitulieren, da meine Leben aufgebraucht sind. Ein neuer Start, ein neues Spiel.

Untermalt wird das Spektakel von diversen Hintergrundgeräuschen, die nicht sonderlich gut gelungen sind. Das fällt aber nicht allzu stark ins Gewicht, da sich der Spieler ständig auf das Spielgeschehen konzentrieren muß. Bis in den 15. Level habe ich es zwar nicht geschafft, aber die hervorragende Qualität der Grafiken dürfte sich wohl kaum verschlechtern. Übrigens sind die Bilder bei der C64-Version fast bessser als die der Amiga-Version. Auch läuft auf dem C64 alles ein klein wenig schneller ab. Aber keine Angst, liebe Amiga-Freaks, auch die Geschwindigkeit auf dem 16-Bit-Computer reicht doppelt und dreifach aus.

Powerstyx sollte in keiner Sammlung fehlen. Ich kann es nur wärmstens weiterempfehlen. Doch lassen Sie sich warnen: Wer es einmal gespielt hat, kommt so schnell nicht mehr davon los.

Thomas Bosch ■



Ports of Call

Stellen Sie sich vor, Sie bekommen fünf Millionen Dollar und wollen daraus 500 Millionen Dollar machen. Bei Ports of Call kann das leicht gelingen. Wer sich allerdings keine Mühe gibt, bei dem wird bald der Pleitegeier über der Reederei schweben.



Ports of Call ist ein Spiel, bei dem der Spieler die Rolle eines Reeders übernimmt. Mit fünf Millionen Dollar müssen nach dem Spielstart erst einmal Schiffe gekauft werden. Am billigsten sind Low-Coast-Ships, diese sehen allerdings auch so aus. Die verbeulten Kähne sollten erst einmal repariert werden. Der Zustand des Schiffes wird in jedem Hafen angegeben, die Tauglichkeit sollte niemals 60% unterschreiten, da sonst das Schiff von einer Rattenplage befallen werden und später sogar untergehen kann. Ist das Schiff also gerade noch schwimmfähig, kann die Ladung ausgewählt werden. Manche Güter können sehr viel Frachterträge bringen, doch bei der Auswahl ist auf sehr viele Dinge zu achten: Wie weit ist es vom Starthafen zum Zielhafen, wieviel Treibstoff wird verbraucht, lohnen die Frachten am Zielhafen, wie hoch ist die Gefahr, Eisberge oder Riffe umfahren zu müssen, sind in diesem Gebiet viele Stürme oder drohen Freibeuter den Tanker zu plündern ? Ist die richtige Entscheidung gefallen, kann es schon losgehen. Zum Ablegen können Schlepper zu Hilfe genommen werden, diese kosten allerdings auch Geld. Außerdem ist auf die Schlepperkapitäne auch kein Verlaß, hin und wieder streiken diese und da ein Reeder keine Zeit zu verlieren hat, heißt es, selbst im Hafen ab- bzw. anzulegen. Dies hört sich zwar sehr einfach an, muß aber erst einmal ausdauernd geübt werden. Ein kleiner Crash gegen die Hafenmauer kostet nämlich viel Zeit und Geld. Sobald das beladene Schiff sicher den Hafen verlassen hat, kann der Spieler auf der Weltkarte sein kleines Schiff entlangtuckern sehen. Unterwegs kann auch so manches Ereignis geschehen, so ist es manchmal nötig, einem Sturm auszuweichen oder Schiffbrüchige zu retten. Im Zielhafen heißt es, nach neuen ertragreichen Ladungen umsehen. Ist keine gewinnbringende Fracht in Sicht, kann ja mal das Schiff auf Vordermann gebracht werden. Nach ein paar Tagen, das Schiff glänzt nun wieder in der Abendsonne, muß dann Geld verdient werden. Deshalb sollten Häfen, in denen es keine ertragreichen Güter gibt, gemieden werden.

Dunkle Geschäfte können natürlich auch getätigt werden. Für einen Schrankkoffer, der irgendwo im Schiff versteckt wird, erhält der Reeder in den meisten Fällen 100.000 Dollar. An 50 Kisten Waffen hingegen kann schon einmal eine Millon Dollar verdient werden. Nur: Der Zoll findet diese Schmugglerei nicht so toll. In den meisten Fällen genügt aber ein saftiges Schmiergeld, um das betroffene Schiff wieder freizukaufen. Sind die ersten 20 Millionen Dollar erwirtschaftet, kann der Reeder ein High-Tech-Schiff mit Querstrahlruder anzahlen. Mit 40% Anzahlung kann sich fast jede Reederei dieses Superschiff leisten. Aber aufgepaßt, mehr als 40 Millionen Dollar sollte für so einen großen Pott nicht bezahlt werden. Natürlich hat jeder Reeder auch ein Büro, von hier aus können die Gewinne und die Verluste abgefragt werden, Hypotheken getilgt oder Kredite aufgenommen werden. Falls mal irgendwann die Geschäfte nicht mehr so gut laufen, kann mit dem Menüpunkt Registerwechsel der Heimathafen des Reeders gewechselt werden. Zu den ertragreichsten Frachtrouten gehören: Rotterdam – Kalkutta und Vancouver – Karachi.

Mittlerweile wird dieses Spiel so gepriesen, als ob es noch kein besseres Wirtschaftsspiel gegeben hätte. Anfangs macht es sicherlich unwahrscheinlich viel Spaß. Doch nach einigen Tagen, wenn der Spieler genug Geld erwirtschaftet hat, verschwindet die Spieldiskette wieder in der Versenkung. Denn die Fehler, die dieses Spiel hat, machen es nach einer gewissen Zeit total lächerlich.

Das fängt schon damit an, daß ein auf Reede liegendes Schiff in Rio de Janeiro von einem Eisberg gerammt werden kann. Es kann weiter mal passieren, daß einem Spieler unterwegs der Treibstoff ausgeht. Das betroffene Schiff muß dann natürlich abgeschleppt werden.

Doch wenn im Hafen wieder aufgetankt werden soll, ist der Dieselbunker noch fast voll. Dem Milliardär macht dieses lächerliche Milliönchen Abschleppkosten überhaupt nichts aus. Doch der kleine Reeder sieht mal wieder den Pleitegeier.

Oder folgender Fall tritt ein: Wegen mangelnder Aufträge müssen die Schiffe im Hafen festgemacht werden. Diese Schiffe rühren sich dann, für die angegebene Zeit, nicht mehr vom Fleck. Doch wenn dann ein anderes Schiff den „Kurs“ des eigenen Pottes kreuzt, welches man mittels einem Radar umfahren sollte, muß man sich schon an den Kopf greifen. Das gleiche Desaster haben die armen Freibeuter, diese greifen meist erst an, wenn das Schiff leer im Hafen liegt oder sie es bereits ausgeraubt hatten. Wenn beim Einlaufen mal wieder die Schlepperkapitäne streiken sollten, ist es immer billiger, das Schiff rückwärts aus dem Hafen zu steuern, denn der Strafzettel ist allemal niedriger als die folgenden Reparaturkosten-

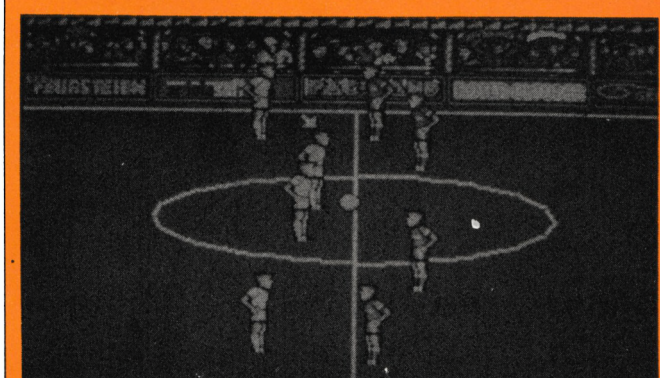
Rechnung. Daß ein Schiff mal auf Kollisionskurs fahren kann, ist die natürlichste Sache der Welt, aber dann dem geisterfahrenden Schiff im Rückwärtsgang zu entfliehen, ist sicher keine Lösung. Irgendwann mal kommt der Tag, an dem sich der Spieler entscheidet, 100 High-Tech-Schiffe zu kaufen. Das benötigt dann schon eine Ewigkeit, aber dann den Spieler hundertfach zu zwingen, alle seine noch im Hafen liegenden Schiffe aus dem Kollisionsbereich zu fahren, ist schon eine Zumutung. Irgendwann pfeift jeder auf Frachten, er kauft die hochmodernen Schiffe für unter ca. 40 Millionen und verkauft diese wieder für ca. 70 Millionen, ohne daß das Schiff jemals den Hafen verläßt.
Im Grunde kann man dieses Spiel niemandem empfehlen. ah ■

Freistoß für Amiga

Die Fußball-EM ist zwar vorbei, aber das bedeutet noch lange nicht, daß jetzt eine Fußball-tote Zeit folgt. Besitzer eines Atari ST oder Amiga können die EM auf ihrem Computer nachvollziehen.

Von Grandslam, bekannt durch Spiele wie Terramex oder Fred Feuerstein, kommt ein neues Sportspiel zu uns: Euro Soccer '88, eine Fußball-Simulation. Da bislang kein übermäßig großes und qualitativ hochwertiges Soccerspiel auf dem Markt ist, war ich entsprechend skeptisch. Allerdings sollte ich eine Überraschung erleben, denn Euro Soccer '88 dürfte nur schwer zu schlagen sein. Sowohl grafisch als auch vom Sound her gehört dieses Spiel zum Besten, was für die bekannten 16-Biter zu haben ist. Große detaillierte Sprites und zahlreiche grafische Effekte motivieren den Spieler.
Zu Beginn wählen Sie sich Ihre Mannschaft aus. Wenn Ihnen die Trikot-Farben nicht zusagen, können diese selbstverständlich geändert werden. Nachdem die wichtigsten Parameter eingestellt worden sind, geht's los.
Wie von den bisherigen Fußball-Simulationen gewohnt, können Sie nur die Steuerung für jeweils einen Spieler übernehmen; welchen, wählt der Computer aus. In den meisten Fällen ist es derjenige, der dem Ball am nächsten ist. Damit Ihnen keine Verwechslung unterlaufen kann, schwebt über der aktuellen Spielfigur ein blinkender Pfeil. Auch die Figur des (Computer)-Gegners ist mit einem Pfeil gekennzeichnet.
Der oder die Spieler steuern die Fußballteams im Turnier, der Computer simuliert die Resultate der übrigen Matches. Damit ist schon in den Entscheidungsspielen Spannung und Nervenkitzel garantiert. Dann geht's ins Halbfinale und in die Endrunden.
Euro Soccer '88 wird in einer stabilen Kunststoffverpakkung auf Diskette geliefert. Als kleine Zugaben finden Sie in der Packung noch ein DIN-A2-Poster und einen Ansteck-Button. Die Anleitung ist komplett in Deutsch verfaßt, so daß auch Spieler, die der englischen Sprache nicht mächtig sind, keine Probleme haben werden.
Das Spiel kann jedem, der sich für Fußball begeistert und ein hervorragend aufgemachtes Spiel schätzt, wärmstens empfohlen werden. Den Preis von knapp 60 Mark dürfen Sie ohne Bedenken auf den Tisch legen. TB ■





Titel:	**Euro Soccer '88**
Getestet:	**Amiga**
Umsetzungen:	**Atari ST**

Im Test: ☒ Diskette ☐ Kassette

Preis (DM): k.A.

☒ Joystick ☒ Tastatur ☒ Maus

Wertung	0	25	50	75	100
Grafik					
Sound					
Bedienung					
Motivation					

Die Wargames-Edition

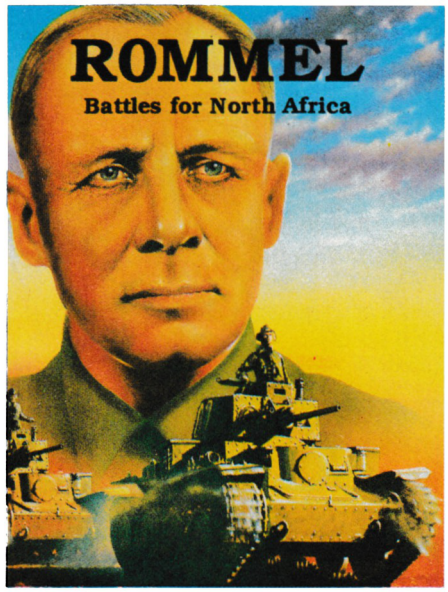
Vor kurzer Zeit erreichte uns eine wahre Flut Kriegsstrategie-Spiele von Electronic Arts Deutschland. Die Spielefreaks und auch die Programmierer scheinen neuerdings Gefallen an dieser zweifelhaften Art von Games zu finden.

Ich mußte lange überlegen, ehe ich mnich entschloß, die Kriegsspiel-Serie von Electronic Arts nehmen. Mit Vorliebe wird der Zweite Weltkrieg behandelt. Aber auch Kollektionen, mit dem jedes nennenswerte Gefecht durchgeführt werden kann, sind dabei. Die Palette reicht vom Flugzeugträgerkrieg im Pazifik über die Schlacht Rommel gegen Patton bis hin zum Bombardement Englands und Deutschlands.



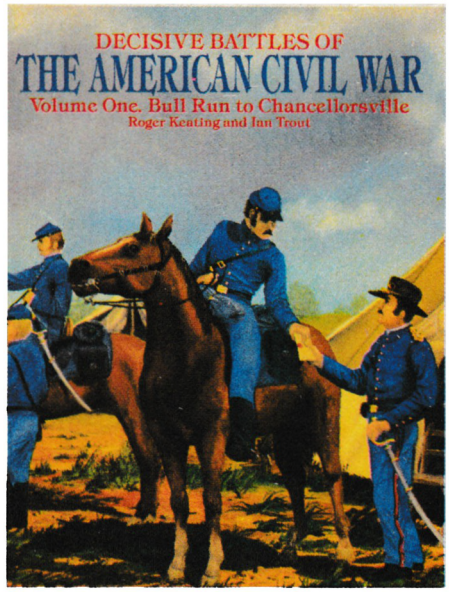
Patton versus Rommel

nennt sich das erste und zugleich beste Spiel dieser Wargames. Es simuliert das Gefecht in der Normandie, als sich der britische General Patton dem deutschen Feldmarschall Rommel gegenübersah.
Das Kriegsspektakel spielt sich im Computer nur auf einer grafisch ganz gut gelungenen Landkarte ab. Alle Einheiten erscheinen auf der Karte als gut erkennbare Symbole, die mit dem Joystick anzuwählen und zu steuern sind. Meldungen und Bestätigungen über eventuelle Aktivitäten erscheinen als gut lesbare Fenster innerhalb der Kartengrafik.
Über den Witz dieser Simulation läßt sich streiten. Ich konnte jedenfalls nur kurze Zeit vor dem Bildschirm ausharren, ehe mir die ganze Sache zu dumm wurde. Aber vielleicht gibt es doch so ausgefuchste Strategen, die dieses Game als eine Herausforderung betrachten.



Rommel – Battles for North Africa

Game zwei beschäftigt sich ebenfalls mit dem Wüstenfuchs Rommel. Bei diesem Spiel können Sie gegen den Computer oder einen zweiten Kommandeur spielen, der entweder die Rolle der Alliierten oder die von Rommel übernimmt. Hier haben Sie jedoch auch die Möglichkeit, unter mehreren Schlachten in Nordafrika zu wählen.
Auch bei diesem Game läuft die Handlung auf der Landkarte ab. Vergleiche mit dem oben genannten bei der Grafik unterlasse ich lieber. Die Darstellung der Karte ist der reinste Horror.
Durch die vielen verschiedenen Farben werden Sie so stark verwirrt, daß Sie sich auf der ohnehin schon Blockgrafik-artigen Karte kaum noch zurechtfinden. Überhaupt ist das ganze Spiel reichlich mit Farbe bestückt, oder besser gesagt: überladen.
Dafür kommt der Sound zu kurz. Wenn überhaupt, dann tönt nur ein langweiliges Gedudel aus dem Lautsprecher, das dem C-64 nicht im geringsten würdig ist.
Gehen wir jetzt etwas in der Zeit zurück, nämlich ins Jahr 1861, als in Amerika der große Ärger begann.



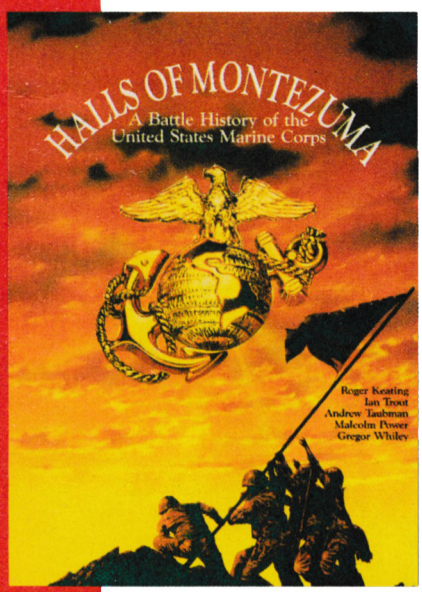
The American Civil War

Der Amerikanische Bürgerkrieg unterscheidet sich im Aufbau, zumindest auf dem Computer, fast nicht von Rommels Schlacht um Nordafrika. Der Bildschirmaufbau ist vollkommen identisch, die Farbgebung auch, und am Sound wurde ebenfalls

nichts verbessert. Einziger Unterschied ist das Szenario. Bedient wird das Spiel über die Cursortasten. Eine Joystick-Steuerung, die wesentlich komfortabler gewesen wäre, wurde nicht eingebaut. Auch hier flaut die Motivation schon nach kurzer Spieldauer ab. Hinzu kommen die langen Wartezeiten beim Laden der Disk, da oft nachgeladen werden muß. Insgesamt also ein sehr mäßiges Spiel.

Halls of Montezuma

Für ganz Kriegslüsterne kommt von denselben Programmierern ein Simulationsspiel, das mehrere



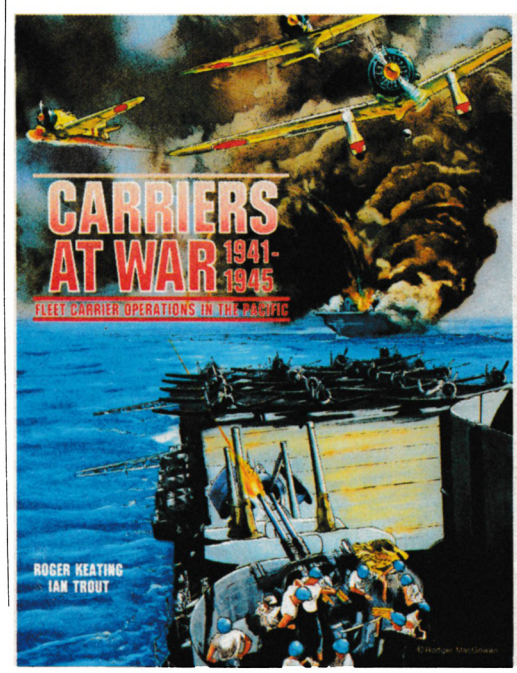
Schlachten, unter anderem auch in den zwei Weltkriegen, zusammenfaßt. Viel mehr kann ich dazu nicht mehr sagen, außer daß sich die Kämpfe vom ersten Weltkrieg über den zweiten und den Koreakrieg bis hin nach Vietnam ziehen. Die ehrenvollen Gemetzel werden natürlich von der amerikanischen Elitetruppe Marines ausgetragen.
Da dieselben Designer am Werk waren, hat sich auch hier gegenüber den anderen Games nicht das geringste geändert. Die Grafik blieb gleich, der Sound auch.
Gehen wir wieder zurück zum Zweiten Weltkrieg und sehen den Flugzeugträgern im Pazifischen Ozean zu.

Carriers AT War 1941–1945

Es ist nicht leicht, einen richtigen Eindruck von diesem Game zu bekommen, ohne die beiden ausführlichen, sechzehn- und vierundzwanzigseitigen Handbücher komplett durchstudiert zu haben.
Ort und Zeit der Handlung ist wieder einmal der Zweite Weltkrieg im Pazifischen Ozean. Zur Auswahl stehen sechs verschiedene Schauplätze, unter anderem Pearl Harbour, Coral Sea, die Midway-Inseln und die Philippinische See. Alles Orte, die erfahrenen Veteranen wohl ein Begriff sind. Hier haben sich zwei Flugzeugträger erbitterte Schlachten geliefert: Die amerikanische „Saratoga" traf auf die japanische „Kaga".
Der Bildschirmaufbau ist wieder einmal der gleiche wie bei den obengenannten Games. Die Steuerung ebenfalls, der schlechte Sound auch.
Trotzdem scheint Carriers at War umfangreicher ausgefallen zu sein, da dem Spiel eine ganze Menge an Handbüchern und Kartenmaterial, das nicht nur bloßer Schnick-Schnack ist, beiliegen.
Kommen wir zum letzten Spiel dieser Episode: der Bombardierung Europas.

Europe Ablaze

In derselben Aufmachung wie Carriers at War geliefert, enthält auch Europe Ablaze jede Menge an Handbüchern und Karten. Diesmal findet die Handlung jedoch nicht auf hoher See statt, sondern hoch in den Lüften. Europe Ablaze simuliert den Luftkrieg über England und Deutschland.
Wie sollte es anders sein, das weitere entspricht den anderen Games. Grafik und Soundeffekte sind völlig identisch, an Komplexität erreicht es mit Leichtigkeit das Pensum von Carriers at War. Gespielt werden kann in drei Szenarien: die Schlacht um England, die Nachtoffensive britischer Bomber und der Angriff auf Berlin.
Ein Wort zu allen hier getesteten Kriegsstrategie-Spielen: Die Aufmachung und die Illustration könnte nicht besser sein. Dem Spieler werden die historischen Hintergründe und Gegebenheiten genau erklärt sowie die verwendeten Waffen, Flugzeuge, Panzer und Schiffe detailgetreu erläutert. Die Handbücher lassen keine Fragen offen und sind sorgfältig und mit viel Liebe zum Detail erarbeitet.
Trotzdem habe ich erhebliche Bedenken gegen das Thema dieser Games. Sie befassen sich leider nur – ohne Ausnahme – mit Krieg. Natürlich, Kriege sind die beste Übung für strategisches Denken, trotzdem sollte man es damit nicht übertreiben. Die Programmierer haben sich große Mühe gegeben, die Schauplätze und Situationen so realitätsgetreu wie möglich zu gestalten. Technisch gesehen sind die Games daher kaum zu übertreffen, obwohl ich das von der grafischen Ausführung nicht gerade behaupten würde.
Kurz gesagt: Es ist Geschmackssache, ob die Spiele gefallen oder nicht. Mich sprechen sie nicht besonders an. Es war außerordentlich schwierig, völlig objektiv zu bleiben, und ich gebe gerne zu, daß mir das nicht ganz gelang. Ausgefuchste Strategen sehen darin aber sicher eine große Herausforderung. Fraglich ist nur, ob sich nicht die Bundesprüfstelle bald für diese Art von Herausforderung interessiert. mn ■





Horrortrip als Comicstrip

Sie haben gerade Lust, sich mal wieder kräftig zu gruseln? Sie haben auch gute Nerven? Und Sie wollen nebenbei auch mal ganz herzhaft lachen? Dann habe ich einen guten Tip für Sie: Besuchen Sie doch mal Rue Morgue, Hausnummer 666. Bitte klingeln bei „Ooze".

Mr. Cheez Burger, wohnhaft in 4233 Denbrough 3575, Rue Morgue No. 666, ist am 28. August 82 um 10 Uhr 25 Minuten in Denbrough, Rue Morgue 666 tot aufgefunden worden.
Der gräßlich verstümmelte Leichnam, die abgetrennten Glieder und die völlige Blutleere des Körpers weisen auf einen möglicherweise rituellen Mord hin. Ebenso die Striemen auf der Haut des Ermordeten, deren Entstehung noch immer unklar bleibt, da sie tiefe Brandmale mit aufweisen. Weitere Ergebnisse der Obduktion können beim zuständigen Gerichtsarzt eingesehen werden."
So steht es in der Sterbeurkunde Ihres so merkwürdig dahingeschiedenen Onkels Cheez Burger. Als Zeugen unterzeichneten unter anderem Mr. Stephen King und Mr. Edgar Allen Poe.
Wenn Sie das gelesen haben, werden Sie sich wohl fragen, was es daran so herzlich zu lachen gibt. Richtig, hier noch gar nichts. Das war nur eine kleine stimmungsmachende Beilage zum deutschen Grafikadventure „OOZE - Als die Geister mürbe wurden."



Einfallsreiche Hintergrundgeschichte

Wenn Sie die mit einem hervorragenden Cover bedeckte Verpackung öffnen, fallen Ihnen zuerst die besagte Sterbeurkunde und ein dreißigseitiges Heft entgegen, das sich als Tagebuch Ihres verstorbenen Onkels entpuppt. Der Roman ist sehr stimmungsvoll geschrieben und eignet sich hervorragend als Einführung in das Abenteuerspiel, zumal die Geschichte dort aufhört, wo das Spiel beginnt.
Als Cheez Burger in die alte Villa einzog, bekam er zuerst große Schwierigkeiten mit den dort wohnenden Geistern aller Art. Diese waren es nämlich nicht gewöhnt, ein fremdes lebendes Wesen zu sehen, das sie möglicherweise in ihrer Ruhe stören könnte. Also bereiteten sie dem alten Cheez kurzerhand eine Menge Terror – bis der Hausgeist Ludus, ehemaliger Hofwitzbold des Königs, herausfand, daß Mr. Burger eigentlich ganz nett ist und den Geistern gar nichts anhaben will. Nach einigen skurrilen Streichen fanden die beiden endlich zusammen und wurden unzertrennliche Freunde – kein Wunder, wenn ein Teil unsterblich war. Nachdem Ludus seine Spukkollegen, T-Bone das Skelett, Vino, den ewig betrunkenen Weingeist und Holunder, dessen liebstes Hobby ist, mit Köpfen zu kegeln, davon überzeugt hatte, daß Cheez Burger zu ihnen hält, hatte dieser eine Gruppe fester Freunde gewonnen.
Leider jedoch hatten auch die Geister ein großes Problem, in das sie Ihren Onkel einweihten: Haustyrann in Carfax Abbey ist ein Monster, das sich Ooze nennt und alle anderen Geister in Schach hält. Dieser Ooze, der laut Ludus nicht mal ein wirklicher Geist ist, weil er noch nie gelebt hat, ist durch und durch böse und hat überall seine Handlanger im Haus versteckt. Ooze hat aber auch eine sehr bedeutende Stärke: Er kann seine Gestalt beliebig verändern.

Roman zum Spiel als Beigabe

Der Plan, Ooze zu vernichten, schlug fehl. Das Ergebnis können Sie in der Sterbeurkunde von Cheez Burger nachlesen.
Hier beginnt das Grafikadventure. Sie müssen als Ham Burger, der Erbe von Cheez, das Haus übernehmen und versuchen, Ooze mit all seinen Spionen zu vertreiben. Daß das gar nicht so einfach ist, hat auch schon Ihr Onkel zu spüren bekommen.
Das wichtigste an einem Abenteuerspiel ist der Eingabeinterpreter. In diesem Fall setzt der deutsche Parser wohl neue Maßstäbe. Er versteht ein großes Stück des Vokabulars und auch die deutsche Grammatik einwandfrei. Lange Sätze, durch Kommas und Bindewörter verknüpft, stellen für ihn kein Problem dar. Wenn er einmal eine Eingabe nicht versteht, dann läßt das Programm Sie das deutlich spüren, indem

es Sie durch einen frechen Kommentar darauf aufmerksam macht.
Eine kleine Schwäche hat der Parser jedoch: Schnelltipper werden sich umgewöhnen und einen langsameren Gang einlegen müssen. Tippen Sie nämlich zu schnell, so kommt der Interpreter nicht mehr mit oder läßt der Bequemlichkeit halber einfach ein paar Buchstaben aus. Nach einer kurzen Eingewöhnungszeit stellt das jedoch auch kein Problem mehr dar.
Der Textteil ist in Deutsch verfaßt und enthält nicht den kleinsten Rechtschreib- oder Grammatikfehler. Vom Umfang her stellt Ooze sogar die Infocom-Produkte in den Schatten. Man kann ohne Übertreibung sagen, daß das neue Dragonware-Game einem illustrierten Horrorroman in der Ich-Perspektive sehr nahe kommt, so detailreich und durchdacht sind die riesigen Textmassen.
Zu einem Grafikadventure gehören natürlich auch Grafiken. Denen von Ooze kann wirklich das Prädikat „Hervorragend" verliehen werden. Zu jedem der siebzig Räume gibt es eine eigene Grafik, an der ich stundenlang die Augen weiden könnte. Teilweise lassen die Bilder sogar die von Jinxter und The Pawn hinter sich.
Mit den Texten und Bildern gelang es den Programmierern ausgezeichnet, eine Horrorstimmung zu erzeugen. Am besten, Sie versuchen nachts, bei Kerzenlicht und geeigneter Musik, Ooze zu schlagen. Aber auch tagsüber läßt die Spannung nicht lange auf sich warten.

Fazit

Mit „Ooze - Als die Geister mürbe wurden" wurde das bislang beste deutsche Adventure programmiert, das sich hinter amerikanischen Produkten wie The Prawn nicht zu verstecken braucht. Durch den gelungenen Parser, die hervorragenden Texte, die sehenswerte Grafik und die witzige Vorgeschichte wird es für deutsche Abenteuerspiele neue Maßstäbe setzen, die zu übertreffen für andere Firmen schwierig sein wird.

mn ■

Titel:	**Ooze**		
Getestet:	**Amiga**		
Umsetzungen:	**Atari ST, MS-DOS, C-64**		
Im Test:	**Preis (DM):**	☐	**Joystick**
☒ 💾	79,90	☒	**Tastatur**
☐ 📼		☐	**Maus**

Wertung	0	25	50	75	100
Grafik					
Sound					
Bedienung					
Motivation					

Ein mörderischer Auftrag



Schon seit über 50 Jahren häufen sich die Rätsel um verschwundene Schiffe und Flugzeuge im Bereich des Bermuda-Dreiecks. Nun soll ein Reporter Licht in die Sache bringen.

Das Rätsel, welches das Bermuda-Dreieck umgibt, beschäftigt schon lange die Wissenschaftler und Theologen. Ganze Schiffskonvois und Flugzeugkolonnen verschwinden plötzlich spurlos vom Radarschirm und tauchen nie mehr auf. Von Mirrorsoft kommt jetzt ein Computerspiel zu uns, das sich mit dem Bermuda-Mysterium beschäftigt.
Sie übernehmen die Rolle eines hochbezahlten Reporters, der von seiner Zeitung den Auftrag erhält, sich auf den Bermuda-Inseln ein wenig umzusehen. Doch auch Sie entkommen dem berüchtigten Dreieck nicht. Ihr Flugzeug stürzt ab.
Nach mehrstündiger Bewußtlosigkeit finden Sie sich auf einer offenbar unbewohnten Insel wieder. Ihr einziger Begleiter, der Pilot, ist tot. Was nun?
Wie sie weiter vorgehen, bleibt Ihnen überlassen. Auch die Anleitung gibt keine Hinweise. Machen Sie sich also auf die Suche nach Einwohnern und spüren sie mit deren Hilfe die Wracks der verschwundenen Schiffe und Flugzeuge auf. Dabei gelten die gleichen Bedingungen wie in der Realität, das heißt, Sie müssen auch für Nahrung und Trinkwasser sorgen und dürfen Ihrer

Fortsetzung auf S. 14

Games für kleine Geldbeutel

Daß gute Software nicht unbedingt teuer sein muß, haben uns diverse Hersteller wieder einmal mit einer geballten Ladung Billigspiele bewiesen.



Alle hier getesteten Billigspiele werden in der gewohnten Kunststoffverpackung auf Kassette geliefert. Zu erhalten sind die Games für etwa zehn bis fünfzehn Mark in den Fachabteilungen der Kaufhäuser und im Computer-Fachhandel.

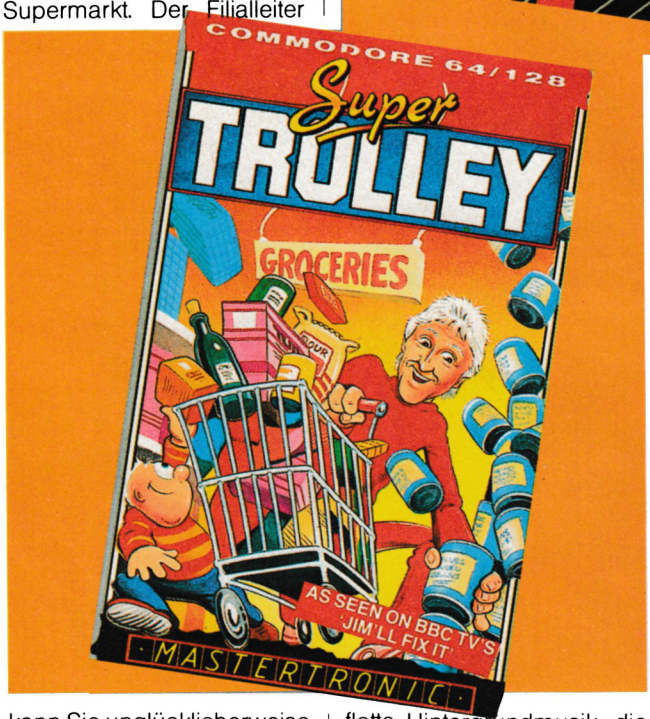
Starquake

Eine geheime Macht hat vom Universum Besitz ergriffen. Zahlreiche Bösewichte haben die friedliebenden Bewohner verjagt und treiben nun ihr zerstörerisches Unwesen. Ihre Aufgabe liegt auf der Hand: Stürzen Sie sich ins Getümmel und machen Sie den unerwünschten Eindringlingen den Garaus. Starquake, das bereits vor einem guten Jahr auf den Markt kam, ist ein Arcade-Action-Game für einen Spieler. Mastertronic hat, wie in letzter Zeit schon oft, die Vertriebsrechte erworben und bringt Starquake als Billigspiel unters Volk. Wer das Game noch nicht hat, der sollte es sich so schnell wie möglich besorgen. Gute Grafiken und flotte Musikuntermalung lassen so schnell keine Langeweile aufkommen. Starquake gibts für CPC, Commodore 64 und Spectrum 48K.



Super Trolley

Sie sind ein kleiner Hilfsangestellter in einem bekannten Supermarkt. Der Filialleiter kann Sie unglücklicherweise nicht ausstehen und schikaniert Sie, wo er nur kann. Heute hat er sich wieder was ganz Fieses ausgedacht: Erst müssen Sie diverse Waren sortieren, dann mit einem Einkaufswagen die Regale leeren und die Artikel anderswo wieder einräumen. Das alles natürlich in möglichst kurzer Zeit und ohne dabei die Kunden anzurempeln.



Bei Super Trolley, dem neuesten Billigspiel von Mastertronic, ist die Grafik zwar nicht gerade berauschend, aber noch ausreichend. Als kleine Entschädigung ertönt zu dem ganzen Spektakel eine flotte Hintergrundmusik, die vom Sound-Hexer Rob Hubbard programmiert wurde. Ob das Game auch Spaß macht, hängt vom persönlichen Geschmack ab. Computerneulingen kann man es vielleicht empfehlen, da die Aufgaben nicht allzu anspruchsvoll sind. Am besten, Sie sehen sich das Game vor dem Kauf genau an. Super Trolley gibt's für Commodore 64.

Subterranea – Rack it

In 16 Levels müssen Sie bei Subterranea Ihr Joystick-Gefühl unter Beweis stellen. Wieder einmal sind bösartige Aliens in unser friedliches Universum eingefallen. Die sollen Sie vertreiben, am besten durch kräftiges Drücken auf den Feuerknopf. Wie bei Airwolf steuern sie in diesem Game von Hewson ein Raumschiff durch die unterschiedlichsten Höhlensysteme. Dabei begegnen Ihnen ständig feindliche Flugobjekte, die sofort das Feuer eröffnen. Es empfiehlt sich, ebenfalls auf die Feuertaste zu drücken und zwar möglichst schnell, denn sonst sind Sie Ihr Raumschiff und damit auch eines der drei Leben unweigerlich los.

Subterranea könnte durchaus als Ballerspiel Erfolg haben, zumal auch Grafik und Sound nicht gerade schlecht sind. Leider jedoch existiert keine Möglichkeit, das Raumschiff mit dem Joystick zu steuern, so daß Sie sich mit einer äußerst umständlichen Tastaturbelegung die Finger verrenken dürfen. So kommt einfach nicht der rechte Spielspaß auf. Getestet haben wir Subterranea – Rack it auf dem Commodore 64 mit Farbmonitor.

Super Gran

Von Tynesoft, vor allem bekannt durch das Sportspiel

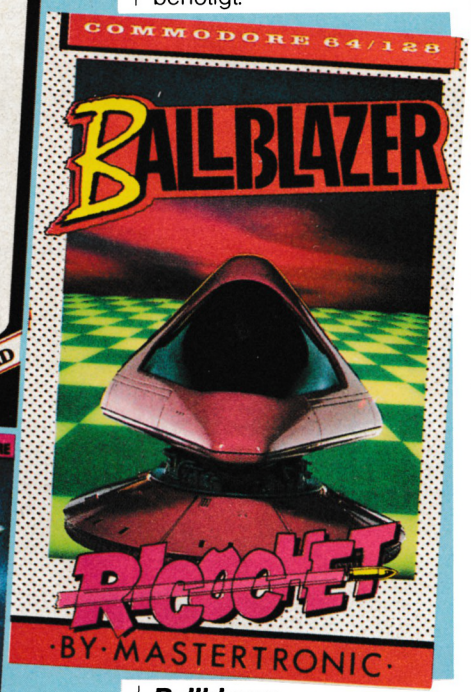
Winter Olympiad '88, kommt ein neues Spiel für den Commodore 16.
Bei Super Gran handelt es sich um das offizielle Computerspiel zur gleichnamigen englischen Fernsehserie. In einem amateurhaft gebastelten Helicopter müssen Sie einige Abenteuer bestehen. Grafisch schneidet Super Gran nicht sonderlich gut ab, da sich Sprites und Hintergrund kaum unterscheiden lassen. Auch das Spielprinzip reißt den Spieler nicht gerade vom Hocker. Wem's aber gefällt…





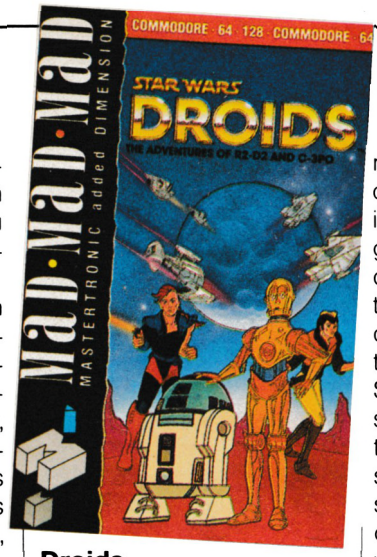
C16 Compilation

Eine preiswerte Spielesammlung bietet Mastertronic jetzt für alle C16/Plus4-Besitzer an. Auf der Kassette, die den eindrucksvollen Namen C16 Compilation trägt, sind vier Spiele, die in Sachen Grafik und Motivation eine preiswerte Alternative zu den meist überteuerten Spielesammlungen sind.
Im einzelnen handelt es sich um Pacmaina, ein Pac-Man-Verschnitt, der von der Geschwindigkeit her seinesgleichen sucht, Disasterblaster, ein gelungenes Weltraum-Ballerspiel, Laza, ebenfalls ein Ballerspiel, allerdings nicht ganz so motivierend, und schließlich das Sportspiel Downhill. Hier müssen Sie einen Abfahrtslauf hinter sich bringen. Für alle vier Spiele wird ein Joystick benötigt.



Ballblazer

Vor etwa 18 Monaten brachte die Firma Lucasfilm Games das Geschicklichkeitsspiel Ballblazer heraus, für das nun Mastertronic die Vertriebsrechte aufgekauft hat. Angeboten wird Ballblazer für den Commodore 64/128 und den Schneider PC. Es ist ein Spiel, bei dem der Bildschirm gesplittet ist, damit zwei Spieler gleichzeitig gegeneinander antreten können. Spaß macht Ballblazer allemal, so daß dem Kauf nichts im Wege steht. Beim CPC empfiehlt sich ein Farbmonitor.

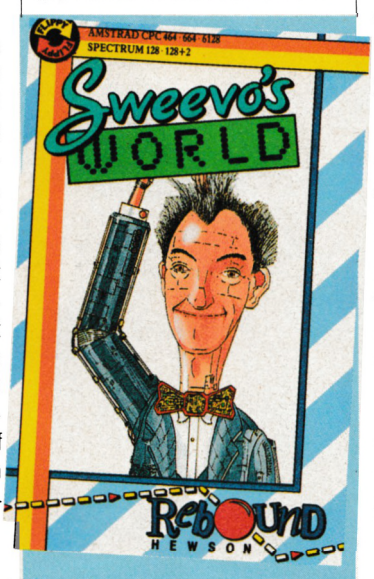


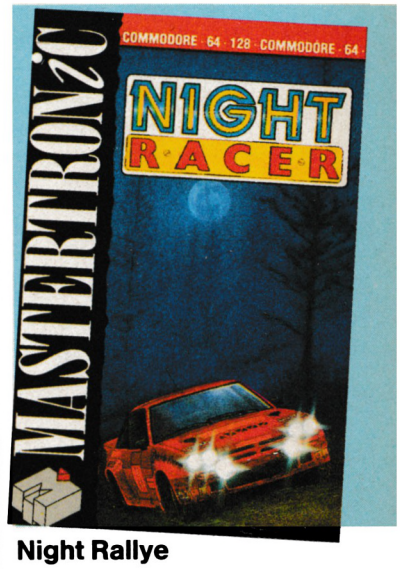
Droids

Droids ist ein neues Ballerspiel aus der Mastertronic-MAD-Serie, das durch hervorragende Grafiken und flotten Sound angenehm überrascht. Ihre Aufgabe ist es, die beiden Roboter R2-D2 und C-3P0 durch ein feindliches Raumschiff zu steuern (möchte bloß wissen, wie die da reingekommen sind!) und dabei so viele Aliens wie möglich zu zerstören. Wenn Ihnen die Namen der Robotis bekannt vorkommen: Ja, es sind die aus Star Wars, wie Sie am Untertitel erkennen können. Droids können Sie künftig auf dem Spectrum 48/128/Plus2 und dem Schneider CPC 464 spielen.

Sweevo's World

Den Vertrieb des vor zwei Jahren sehr erfolgreichen Spiels Sweevo's World hat die Firma Mastertronic übernommen und für den Schneider CPC und den Spectrum in neuer Verpackung herausgebracht. Sie übernehmen in diesem Game die Rolle der tollpatschigen Ente Sweevo, die ihren Planeten von bösartigen Sprites säubern muß. Sweevo's World ist ein typischer Vertreter der bekannten 3D-Adventures, wie es sie seit Knight Lore dutzendweise auf dem Markt gibt. Allerdings bietet das Spiel wesentlich bessere Grafiken als seine „Soft-Kollegen". Wer also noch nicht allzu viele Games dieser Kategorie zu Hause hat, darf sich Sweevo's World ohne Bedenken zulegen. Der Preis von knapp 15 Mark ist sicher nicht zu hoch.





Night Rallye

Eine Rallye-Simulation für den Commodore 64/128 bietet ebenfalls Mastertronic an. Allerdings darf man sich vom Titel nicht irreführen lassen, handelt es sich doch wieder einmal nur um ein simples Autorennspiel, das überhaupt nichts Neues bietet. Night Rallye kann nur allein und nur mit Joystick gespielt werden. Grafisch ist Night Rallye ganz gut gelungen, auch die Soundeffekte können sich hören lassen. Da das Spiel auch nur knapp über 10 Mark kostet, dürfte es eine interessante Alternative zu den teuren Autorennspielen sein. Thomas Bosch ■

Fortsetzung von S. 11

Spielfigur keine allzu große Anstrengung zumuten. Um ans Ziel zu kommen, ist es außerdem ratsam, jedes Objekt, sei es ein Stein, ein Baum oder eine Trinkwasserquelle, peinlich genau zu untersuchen.
Gesteuert wird die Spielfigur mit der Maus. Über die beiden Maustasten lassen sich auch durch Pulldown-Menüs die verschiedenen Aktionen anwählen. Ihre Spielfigur kann Objekte untersuchen, aufnehmen, ablegen, benutzen, verbinden und trennen. Außerdem können Sie sich jederzeit den Status Ihres Reporters betrachten. Dort finden Sie auch eine entsprechende Warnung, wenn der Held kurz vor der Erschöpfung steht.
Daß das abzusuchende Gebiet relativ groß ist, kann es ziemlich lange dauern, bis das Spielende erreicht ist. Glücklicherweise haben die Autoren auch Optionen zum Abspeichern und Wiedereinladen des aktuellen Spielstandes eingebaut. Hierzu muß sich eine frisch formatierte Diskette im Laufwerk A befinden.
Die Grafik von Bermuda-Projekt beschränkt sich zwar lediglich auf ein paar Objekte, ist aber völlig ausreichend. Da es notwendig ist, sämtliche Objekte zu untersuchen, würde sich das Spiel lange hinziehen, wenn der Bildschirm damit überfüllt wäre. Außerdem dürfen Sie nicht vergessen, daß Sie sich vorwiegend am Sandstrand einer einsamen Insel aufhalten, wo es keine Zivilisation gibt.
Zu Beginn des Spieles erklingt aus dem Lautsprecher ein digitalisiertes Musikstück. Ansonsten sind nur diverse Hintergrundgeräusche zu hören.

Fazit

Bermuda-Projekt ist ein Spiel für Leute mit viel Zeit und Geduld. Wer also einmal länger als eine Stunde an einem Computergame sitzen möchte und außerdem eine Vorliebe für Rätsel hat, dem kann ich dieses Spiel nur empfehlen. TB ■

Titel:	**Bermuda Project**
Getestet:	**Atari ST**
Umsetzungen:	**Amiga (in Vorbereitung)**

Im Test: ☒ ☐ **Preis (DM):** 59,- **Joystick** ☐ **Tastatur** ☐ **Maus** ☒

Wertung	0	25	50	75	100
Grafik					
Sound					
Bedienung					
Motivation					

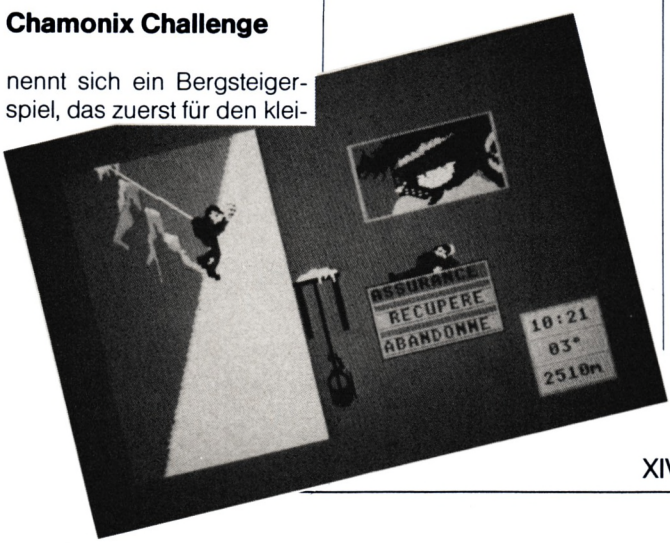
Kurzberichte

Dieser Monat brachte leider nicht viele Neuerscheinungen und Umsetzungen für alle Computer. Sowohl für den C-64 als auch für Atari ST und Amiga hat wohl die Sommerpause eingesetzt.

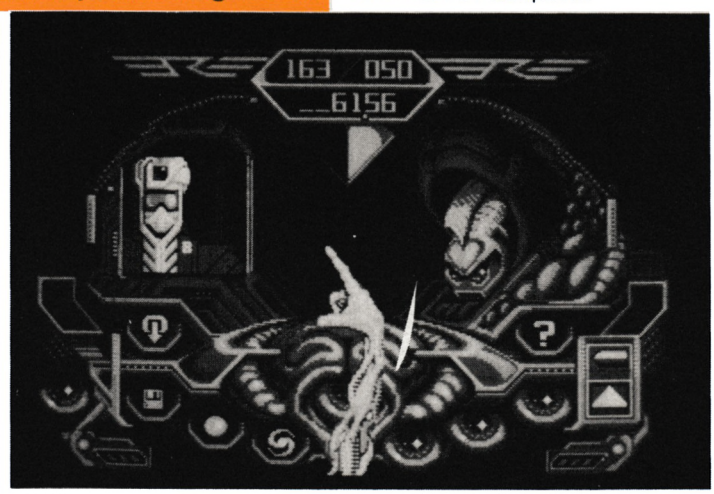
Zwei Umsetzungen brachte Infogames für den Amiga auf den Markt. Beide sind schon seit einiger Zeit für andere Computer wie C-64 und ST erhältlich und wurden von uns getestet.

Chamonix Challenge

nennt sich ein Bergsteigerspiel, das zuerst für den kleinen Commodore erschien. Ziel des Spiels ist die Besteigung eines oder mehrerer Berggipfel. Dazu müssen sowohl Gletscher als auch Felswände erfolgreich überwunden werden. Das Game ist ein technisch hochwertiges



und recht schwieriges Geschicklichkeitsspiel.
Die Grafik ist für Amiga-Maßstäbe jedoch enttäuschend ausgefallen. Sie unterscheidet sich nur geringfügig von der des Commodore 64. Die Schwierigkeit und der Spielwitz sind jedoch gleichgeblieben. Bergfans sollten sich Chamonix Challenge einmal ansehen. Für andere ist es empfehlenswert, erst einmal probezuspielen.
Das andere Game wurde vom Atari ST adaptiert:



Die Arche des Captain Blood

Captain Blood ist ein sehr kompliziertes Spiel, das in

keine Kategorie so recht einzuordnen ist. Aufgabe ist es, als Programmierer seine Einzelteile wieder aufzusammeln, die in einem Universum innerhalb des Computer verstreut sind. Dazu muß er Kontakt mit fremden Lebewesen aufnehmen. Exotische Planeten anzufliegen und gegebenenfalls auch zu zerstören ist hier angesagt.
Die Grafik ist identisch mit der des Atari ST. Das Spiel hat rasante Fraktalgrafik und herrliche Farbeffekte vorzuweisen. Musik und Sounduntermalung sind auch nicht zu verachten. Insgesamt eine gelungene Umsetzung eines gelungenen Spiels.



The Tree Stooges

Nur einen Kurztest war uns The Tree Stooges für den Amiga wert. Ziel des Spiels, das sich an der Fernsehserie orientiert, ist, für eine verarmte Familie Geld aufzutreiben. Dazu lasssen sich die drei Clowns allerlei einfallen.
Grafisch ist das Spiel allererste Güte, der Spielwitz bleibt jedoch auf der Null-Marke. Das, was sich die Programmierer haben einfallen lassen, mag ja als Grafikdemo ganz gut sein, für ein Spiel ist jedoch nichts drin. Oder finden Sie daran Gefallen, wenn sich einige Leute minutenlange Tortenschlachten liefern oder ein Watschenspiel veranstalten, bei dem es einzige Aufgabe des Spielers bleibt, den Feuerknopf zu drücken?
Um ein gutes Spiel zu gestalten, hätten sich die Designer mehr einfallen lassen müssen als einige schwachsinnige Comicstrips mit teilweise recht langen Sequenzen dazwischen, in denen der Spieler nichts zu tun hat als kräftig zu gähnen.



Gauntlet II für Atari ST

Der glänzende Spielautomat und Nachfolger des Arcade-Games Gauntlet wurde jetzt auch für den Atari ST adaptiert. Die Umsetzung ist vom Automaten kaum zu unterscheiden. Das Game hat schnelle und hochauflösende Grafik und hervorragenden Sound vorzuweisen. Sogar an die Sprachausgabe des Automaten wurde gedacht. Ein besonderer Gag ist, daß auch am ST vier Spieler mitspielen können. Allerdings ist das leider nur mit einem speziellen Joystick-Interface möglich. Trotzdem ist diese Version wesentlich besser gelungen als sein Vorgänger Gauntlet I.

Atari-Umsetzung von ZARCH

Das bekannte Archimedes-Spiel Zarch können nun auch ST-Besitzer genießen. Das schnelle Aktiongame, bei dem außerirdische Wesen aller Art abgewehrt werden müssen, nennt sich auf dem ST Virus. Die Grafik bleibt im wesentlichen gleich, nur die Geschwindigkeit hat gelitten, was aber nichts heißen will. Das Game ist noch schnell genug, um einen fließenden Spielverlauf zu garantieren. Atari-Besitzer sollten sich Virus unbedingt ansehen.

Aktion-Game für Atari ST

She-Fox nennt sich ein neues Aktion-Spiel, das unter anderem auch für den ST erscheint. Das Game ist ein simples Scroll-Spiel, bei dem ausnahmsweise einmal eine Heldin die Hauptrolle spielt. Die Amazone muß sich am laufenden Band mit ihrer Peitsche gegen wilde Tiere und ein Zeitlimit behaupten. Grafisch ist das Spiel ganz gut gelungen, vom Spielwitz blieb allerdings nichts mehr übrig. Bereits nach einigen Spielminuten wird die ständige Raserei in eine vorgegebene Richtung langweilig. Insgesamt also ein sehr mittelmäßiges Spiel.

Pink Panther für den Commodore 64

Das in der letzten Ausgabe getestete Amiga-Spiel The Pink Panther wurde für den C-64 adaptiert. Kaum zu glauben: Die Grafik ist zwar nicht so detailreich, dafür aber schneller als auf dem Amiga. Auch der Sound ist für C-64-Maßstäbe hervorragend. Der Spielwitz hat nicht im geringsten gelitten. Aus dem kleinen Commodore haben die Programmierer herausgeholt, was er hergibt. Eine Umsetzung, wie sie nicht besser sein könnte. mn ■

Anzeige

◀◀◀ SSS ▶▶▶ Siggis Software Shop ◀◀◀ SSS ▶▶▶
★ Knüllerpreise ★ Ein Preisvergleich lohnt sich immer ★ Knüllerpreise ★

Spiele	Amiga/ST	Spiele	Amiga/ST	Spiele	Amiga/ST
Arctic Fox	54,50/54,50	Jump Jet	42.50/42,50	Superstar Icehoc	64,50/64,50
Arkanoid	64,50/42,50	Karting Grand Prix	26,50/26,50	Terramex	54,50/54,50
Backlash	45,50/44,50	King of Chicago	58,50/58,50	Test Drive	74,50/74,50
Bad Cat	49,50/54,50	Las Vegas	26,50/26,50	Tetris	54,50/54,50
BMX Simulator	42,50/44,50	Leaderboard Golf	64,50/64,50	Time & Magic	54,50/54,50
California Games	64,50/64,50	Lurkins Horror	74,50/74,50	Two & Two Basketb.	64,50/64,50
Chessmaster 2000	72,50/74,50	Moebius	64,50/64,50	Willy the Kid	26,50/26,50
Clever & Smart	54,50/56,50	Phantasie 3	54,50/58,50	Winterolympiade 88	56,50/56,50
Defender of Crown	66,50/64,50	Ports of Call	88,50/—,—	Wizball	54,50/54,50
Dungeon Master	64,50/64,50	Planetfall	—,—/74,50	Xenon	54,50/54,50
Eagles Nest	54,50/56,50	Rings of Zilfin	64,50/64,50		
Emerald Mine	26,50/26,50	Roadwar 2000	54,50/54,50	Anwenderprogramme	Amiga/ST
Fam. Feuerstein	54,50/54,50	Roadwar Europa	68,50/68,50	C-64 Emulator	156,50/—,—
Flight-Simulator 2	99,50/98,50	Roadwars	54,50/54,50	Cambridge Lisp	—,—/358,50
Frightnigth	64,50/64,50	Rolling Thunder	64,50/54,50	Lattice C	528,50/—,—
Ferrari Formula 1	72,50/—,—	Sindbad	68,50/58,50	Lisp	478,50/—,—
Giana Sisters	52,50/54,50	Slaygon	54,50/54,50	Macro Assembler	178,50/—,—
Gridstart	26,50/26,50	Soccer King	26,50/26,50	MCC Pascal	—,—/248,50
Hotball	64,50/64,50	Strip Poker 2 Plus	42,50/44,50	Pascal	248,50/—,—
Ifidon	42,50/—,—	Sub Battle Sim.	64,50/64,50	Pro Sprite Designer	—,—/ 99,50

S. Gebauer
Parkstr. 7a
5880 Lüdenscheid
Tel. (0 23 51) 2 45 02

◀◀ SSS ▶▶
◀◀ SSS ▶▶
◀◀ SSS ▶▶
◀◀ SSS ▶▶

Versandkosten: Vorkasse + DM 4,50 / Nachnahme + DM 7,50. Zur Auslieferung gelangt ausschließlich nur Originalware. Angebote freibleibend. Liefermöglichkeiten vorbehalten. Bei großer Nachfrage nicht jeder Artikel sofort lieferbar.

Players' Pages

Hallo, liebe Spielefreaks und Joystickartisten. Willkommen bei den Players Pages der LOAD & RUN. Wer Tips, Tricks, Pokes und Lösungen zu jedem x-beliebigen Spiel und für jedes x-beliebige System hat oder sucht, ist hier richtig!

Letzten Monat hat sich wieder viel getan bei uns. Entsprechend umfangreich fallen die Players Pages aus. Doch ich will mich nicht mit langen Vorreden aufhalten, sondern gleich loslegen.

Ranarama

Für Ranarama-Spieler auf dem C64/128 haben wir folgenden Poke bereit, der nach dem üblichen load-reset-poke-sys-Rezept eingebaut wird.

POKE 37104,96
POKE 33969,234
POKE 33970,234

Game Over Part II

Für Part II von Game Over finden viele kein passendes Codewort. Wie wär's denn mal mit ZAPPA? Übrigens werden für dieses Spiel noch jede Menge Strategie-Hinweise gesucht. Für alle Game-Over-Spieler auf dem Spectrum habe ich auch noch einen Code anzubieten: 18024.

Terramex

Haben Sie schon mal versucht, mit dem Ballon in die Lüfte zu steigen? Sicherlich, denn mit dem Schirm geht es ja einwandfrei. Aber wer kann uns verraten, wie der Ballon wieder zum Landen überredet werden kann? Die LOAD & RUN-Redaktion hat es auch nach mehreren durchwachten Nächten noch nicht geschafft. Übrigens: Wenn Sie es nicht schaffen, warum gehen sie dann nicht in den Untergrund. Wenn Sie in den Brunnen klettern, kommen noch jede Menge weitere Räume, in denen unzählige Gefahren lauern wie beispielsweise eine Schlange, die aber durch Flötenmusik beruhigt werden kann.

Die Erbschaft

Viele Leser haben sich über die Tips zum dritten Teil der Erbschaft in LOAD & RUN 4/88 gefreut. Andere dagegen konnten damit nichts anfangen, weil sie nicht mal den ersten Teil beenden konnten. Um ihrem Leiden nun ein Ende zu bereiten, hier eine Liste mit den Gegenständen, welche die einzelnen Personen erhalten:

Neger.....Trompete
Mann mit Glatze .....Taschenlampe
Mafioso .....Pistole
Chinese.....Kerzenhalter
Mann mit Scheiten..... Kaktus
Punkerin ..... Bügeleisen
Mann mit Glatze & Bart..... Kette

Bei der Gelegenheit noch ein kleiner Tip: Es empfiehlt sich, stets zu Fuß zu gehen und den Aufzug zu meiden. Im Zimmer gleich links nach Ihrem eigenen Wohnraum finden Sie in der Schublade ein zusätzliches Taschengeld.

ANTI-BUSINESS-PROGRAMME!

Samantha Fox-Strip-Poker
Stud-Poker müssen Sie schon gespielt haben. Ob Sie es schaffen, Ihre Spielkarten stets im Auge zu behalten? Dann Samantha ... auch! Für C-64 Cassette: DM 13,90
Für C-64 Cassette: Cass: DM 13,90
Für Schneider Cass: DM 13,90
Für Spectrum Cass: DM 13,90
Für Schneider Disk: DM 19,90

WIR LIEFERN AUSSCHLIESSLICH ORIGINAL-PROGRAMME DER HERSTELLER – ZU NIEDRIGEN PREISEN!

AMIGA-Software

Amiga Tools-Virus Killer	49,90		
Backgammon	29,90	Jump Jet	39,90
Bard's Tale I	79,90	Kikstart II	29,90
Bard's Tale II	69,90	Ports of Call	89,90
Emulator C-64 m. Kabel	159,90	Red October	59,90
Chessmaster 2000	79,90	Roadwars	59,90
Dark Castle	69,90	Shadow Gate	69,90
Defender of the Crown	79,90	Space Quest	69,90
Diablo	49,90	Star Glider	69,90
Feud	29,90	Star Wars	59,90
Flight Simulator II	129,90	Strike Force Harrier	79,90
Flintstones	59,90	Terramex	59,90
Formule One Grand Prix	59,90	Terropods	59,90
Garrison	59,90	Tetris	59,90
Garrison II	59,90	The Pawn	69,90
Giana Sisters	59,90	Vampire's Empire	59,90
Grand Slam Tennis	69,90	Wizball	69,90
Hollywood Strip Poker	49,90	World Games	69,90
Impact	39,90	Zork I	89,90
In 80 Tagen um die Welt	59,90	Zork II	89,90

ATARI ST Software

3-D-Galax dt.	49,90		
Backlash dt.	59,90		
Bad Cat	49,90		
Bard's Tale I	89,90		
Black Cauldron	119,90		
Blueberry	59,90		
Broker dt.	59,90		
Bubble Bobble	49,90		
Championship Wrestling	39,90		
Color-Emulator	129,90		
Deep Space	39,90		
Defender of the Crown	89,90		
Flight Simulator II	129,90		
Giana Sisters	59,90		
Hollywood Strip Poker	39,90		
Impact	39,90		
In 80 Tagen um die Welt	59,90		
Kaiser dt.	129,90		
Karate dt.	49,90		
King's Quest I, II, III	99,90		
Leisure Suit Larry	59,90		
Little Computer People	59,90		
Lucky Luke dt.	59,90		

C-64 Software

Alternative Worldgames	C29,90	D39,90
Arkanoid II	C29,90	D39,90
Armaggedon Man		D49,90
Bard's Tale I		D49,90
Bard's Tale II		D49,90
Bard's Tale III	C19,90!!!	
Boulderdash Constr.Kit	C29,90	D39,90
Championship Sprint	C35,90	D49,90
Chessmaster 2000	C25,90	D35,90
Championship Wrestling	C39,90	D49,90
Chuck Yeager's AFT	C29,90	D39,90
Deflector		D69,90
Flightsimulator	C29,90	D39,90
Fred Feuerstein	C29,90	D49,90
Garfield	C29,90	D39,90
Gauntlet II	C29,90	D39,90
Giana Sisters	!!!	D39,90
Helloween DEUTSCH	C29,90	D39,90
Impact	C29,90	D39,90
Implosion	C29,90	D39,90
Impossible Mission II		D39,90
In 80 Tagen um die Welt	C49,90	D69,90
Jagd auf Roter Oktober	C39,90	D49,90
Laser Basic DEUTSCH	C49,90	D69,90
Laser Compiler DEUTSCH	C39,90	D49,90
Laser Genius Assembler		D59,90
Legacy of the Ancients	C19,90	SUPER!!
Marble Madness SUPER!!!		D39,90
Mini Putt	C25,90	D35,90
Outrun	C19,90	D35,90
Pi R 2	C49,90	D59,90
Pirates!	C49,90	D59,90
Shoot 'em up Constr. Kit	C29,90	D35,90
Sidearms	C29,90	D39,90
Silent Service	C29,90	D39,90
Solid Gold	C25,90	D35,90
Star Wars	C29,90	D39,90
Slifflip & Co.	C29,90	D39,90
Superstar Icehockey		D29,90
Taipan SUPERPREIS!!!!	C29,90	D39,90
Task 3	C29,90	D39,90
Terramex	C29,90	D39,90
To be on Top	C25,90	D35,90
Trantor		D59,90
Vermeer DEUTSCH	C29,90	D39,90
Volleyball Simulator		D39,90
Winter Edition		

Football Manager II
Brandneu: Die Sport-Strategie-Simulation zu einem Super-Preis:
Für C64 Diskette: DM 39,90
Für Schneider Diskette: DM 39,90
Für Amiga Diskette: DM 59,90
Für Atari ST Disk: DM 59,90
Für C-64 Cassette: DM 29,90
Für Schneider Cass: DM 29,90
Für IBM-PCs Disk: DM 59,90

C-16 Software

Joystick-Adapter			
Aliens	8,90	Las Vegas Video Poker	9,90
Auf Wiedersehen Monty	25,90	Megabolts (Action-Game!)	9,90
BMX-Simulator	19,90	Molecule-Man	9,90
Bubble Trouble	9,90	Music-Synthesizer	9,90
Datei DEUTSCH	9,90	Obido	29,90
European Games (Sport)	29,90	Pinpoint	9,90
5-Star-Games I	24,90	Pogo-Pete	19,90
Formula One	28,90	Powerball (Arcade-Game!)	9,90
Frank Bruno's Boxing	9,90	Saboteur	9,90
Gwnn (Super-Action-Spiel)	9,90	Scooby Doo	9,90
Hektik	9,90	Speed-King (Motorrad)	25,90
International Karate	8,90	Star Force Nova	9,90
Kikstart	14,90	Summer Events	9,90
Krösus	9,90	Text DEUTSCH	29,90
	29,90		29,90

Schneider CPC Software

3D-Voice-Chess	C39,90	D49,90
Arkanoid Rev. of Doh	C29,90	D39,90
Bard's Tale I	C29,90	D39,90
Battleships (super!)	C 9,90	
Bubble Bobble	C29,90	D39,90
California Games	C29,90	D39,90
Equinox	C13,90	D19,90
Flunky (DIE Hektik!)	C25,90	D35,90
Formula One	C 9,90	
Glass (Space-Arcade)	C 9,90	
Gryzor		D39,90
International Karate	C 9,90	D19,90
Jagd auf Roter Oktober	C49,90	D59,90
Laser Basic dt.	C29,90	D39,90
Laser Compiler dt.	C39,90	D49,90
Pirates! nur 6128		D59,90
The Rocky Horror Show	C15,90	D19,90
Sidearms	C29,90	D39,90
Starwars	C25,90	D39,90
Terramex	C29,90	D39,90
Tetris	C29,90	D39,90
Volleyball Simulator	C29,90	D39,90

SOFORT-BESTELLUNG PER TELEFON: 0911/28 82 86

ATARI ST Software

Marble Madness dt.	79,90	Vampire's Empire	59,90
Mercenary	39,90	Solomon's Key	49,90
Mono-Emulator	129,90	Space Quest II	59,90
Mortville Manor dt.	69,90	Starwars	59,90
Outrun	59,90	Sub Battle Simulator	39,90
Phantasie dt.	39,90	Super Cycle	59,90
Pinball Factory	59,90	Terramex	79,90
Power-Play	69,90	Testdrive	49,90
Psion Chess dt.	69,90	Wizball	59,90
Silent Service		Virus	

Alle Preise sind unsere Ladenpreise. Bei Versand berechnen wir anteilige Selbstkosten: bei Vorkasse mit Scheck: DM 2,50, bei Versand per Nachnahme DM 5,90 je Sendung.

BESTELLUNG + INFO ANFORDERUNG

☐ Hiermit bestelle ich für den Computer _____
nachstehende Programme per ☐ Nachnahme (+ Kosten 5,90) ☐ Vorkasse und Scheck (+ Kosten 2,50)

☐ Ich möchte ein kostenloses Gesamtinfo über Software für meinen Computer.

Bitte Anschrift nicht vergessen _____ Unterschrift
1 3 4

T.S. Datensysteme · Denisstraße 45 · 8500 Nürnberg 80

T.S. Datensysteme
DENISSTRASSE 45 · 8500 NÜRNBERG 80 · TELEFON 0911/28 82 86

1 L&R

ANZEIGENSERVICE

Die große Börse für jeden Zweck in CPC-WELT. Kostenlos für Privat-Inserenten. Spottbillig für gewerbliche Anbieter. Einfach Coupon ausschneiden, fotokopieren o.ä., ausfüllen und ab die Post — Freimachen nicht vergessen! — Unsere Adresse steht auf dem Coupon, ebenso die Preise für gewerbliche Anbieter! Wir weisen ausdrücklich darauf hin, daß wir offensichtlich gewerbliche Anzeigen nicht kostenlos veröffentlichen und uns jedweden Abdruck kostenloser Anzeigen vorbehalten müssen, insbesondere, wenn deren Inhalt gegen geltendes Recht verstößt. Private Chiffreanzeigen werden nicht aufgenommen. Für Privatanbieter: etwa bis zu acht Zeilen à 28 Anschläge. Für gewerbliche Anbieter: 5 DM p. Zeile bei beliebiger Größe.

CPC-WELT
ANZEIGENABTLG.
POSTFACH 1161
D-8044 UNTERSCHLEISSHEIM

Name _____
Vorname _____
Straße/Hausnr. _____
PLZ/Ort _____

DAS SONDERANGEBOT: PRIVATE KLEINANZEIGEN SIND KOSTENLOS

Das bietet Ihnen CPC-WELT. KLEINANZEIGEN SIND KOSTENLOS FÜR PRIVATANBIETER! Suchen Sie etwas, haben Sie etwas zu verkaufen, zu tauschen, wollen Sie einen Club gründen? Coupon ausfüllen, auf Postkarte kleben oder in Briefumschlag stecken und abschicken. So einfach geht das. Wollen Sie das Heft nicht zerschneiden, können Sie den Coupon auch fotokopieren. Oder einfach den Anzeigentext uns so schicken, auf Postkarte oder im Brief. Aber bitte mit Druckbuchstaben oder in Schreibmaschinenschrift!
Und: Einschließlich Ihrer Adresse und/oder Telefonnummer sollten acht Zeilen à 28 Anschläge nicht überschritten werden.
ACHTUNG: WICHTIGER HINWEIS!
Wir veröffentlichen nur Kleinanzeigen privater Inserenten kostenlos, gewerbliche Anzeigen kosten pro Zeile zu 29 Buchstaben DM 5,00 plus Mehrwertsteuer!
Wir versenden für Privat-Inserenten keine Belegexemplare!
DIE INSERTION IST NICHT VOM HEFTKAUF ABHÄNGIG!
Chiffre-Anzeigen sind nicht gestattet! Wir behalten uns vor, Anzeigen, die gegen rechtliche, sittliche oder sonstige Gebote verstoßen, abzulehnen!
Anzeigenabdruck in der Reihenfolge ihres Eingangs, kein Rechtsanspruch auf den Abdruck in der nächsten Ausgabe!
Wir behalten uns vor, Anzeigen, die nicht zum Themenkreis des Heftes — Computer — gehören, nur insoweit zu berücksichtigen, wie es der Umfang des kostenlosen Anzeigenteils zuläßt.

VERDIENEN SIE GELD MIT IHREM COMPUTER!

WIR ZAHLEN BIS ZU 1000 DM!

Haben Sie einen CPC 464 oder 664 ? Einen 6128? Können Sie programmieren? In Basic oder Maschinensprache? Dann bietet CPC-WELT Ihnen die Möglichkeit, mit diesem Hobby Geld zu verdienen!

Wie? Ganz einfach. Sie senden uns die Programme, die Sie für einen Abdruck als geeignet halten, zusammen mit einer Kurzbeschreibung, aus der auch die verwendete Hardware — eventuelle Erweiterungen — benutzte Peripherie — hervorgehen muß, ein.

Benötigt werden: eine Datenkassette oder Diskette! Wenn die Redaktion sich überzeugt hat, daß dieses Programm läuft und sich zum Abdruck eignet, zahlen wir Ihnen pro Programm je nach Umfang bis zu DM 300,—! Für das ,,Listing des Monats'' sogar DM 1.000,'—.

Sie erhalten Ihre Kassette/Diskette selbstverständlich zurück, wenn Sie einen ausreichend frankierten Rückumschlag mit Ihrer Adresse beifügen.

Bei der Einsendung müssen Sie mit Ihrer Unterschrift garantieren, daß Sie der alleine Inhaber der Urheber-Rechte sind! Benutzen Sie bitte anhängendes Formular! (Wir weisen darauf hin, daß auch die Reaktion englische Fachzeitschriften liest und ,,umgestaltete'' Programme ziemlich schnell erkennt).

Um Ihnen die Arbeit zu erleichtern, finden Sie hier ein Formular. Sie können es ausschneiden oder fotokopieren.

Name des Einsenders: _____

Straße/Hausnr./Tel.: _____

PLZ/Ort: _____

Hiermit biete ich Ihnen zum Abdruck folgende(s) Programm(e) an:

_____

_____

_____

_____

Benötigte Geräte: _____

_____

Beigefügt ☐ Listings ☐ Kassette ☐ Diskette

Ich versichere, der alleinige Urheber des Programms zu sein!
Hiermit ermächtige ich die Redaktion, dieses Programm abzudrucken und wirtschaftlich zu verwerten. Sollte es in den Kassetten-Service aufgenommen werden, erhalte ich auch dafür eine entsprechende Vergütung. Das Copyright geht an den Verlag über.

_____

Rechtsverbindliche Unterschrift

CPC-WELT
PROGRAMM-REDAKTION
POSTFACH 1161
D-8044 UNTERSCHLEISSHEIM

TOPSTAR

Ballerspiele haben noch immer Hochkonjunktur. Besonders wenn sie schnell sind, grafisch gute Effekte haben und abwechslungsreiche Spielsituationen aufweisen.

Unser Autor Christian Will hat ein Spiel geschrieben, bei dem sich das Abtippen wirklich lohnt. In Topstar haben Sie die Aufgabe, die Erde vor außerirdischen Angreifern und tödlichen Kometen zu schützen. Aber achten Sie auf Ihren Energievorrat. Denn jeder Schuß und jeder feindliche Treffer kosten Brennstoff.
Sie haben jedoch die Möglichkeit, Ihre Vorräte zu ergänzen. Dazu brauchen Sie lediglich auf der Startrampe zu landen. Dies ist allerdings nur möglich, solange sie noch nicht zerstört ist. Das Game läßt sich mit der Tastatur oder dem Joystick spielen. Entscheiden Sie sich für die Tastatur, so gilt folgende Belegung: Auf=R, Ab=F, Links=O, Rechts=P, Feuer=Leertaste.
Das Programm ist gut strukturiert und in reinem BASIC geschrieben. Durch die gute Kommentierung können Sie, wenn Sie wollen, Veränderungen vornehmen oder das Spiel ergänzen. Die Geschwindigkeit ist sehr hoch. Sie müssen auf der Hut sein, damit Sie Ihre Jäger nicht verlieren. Da nach einiger Zeit auch die Objekte, die Sie abschießen sollen, ihr Aussehen verändern und noch aggressiver werden, ist für Abwechslung gesorgt. Das Spiel läuft auf allen CPC-Typen. *JE* □

```
10 '*****************************  <2397>
20 '*        TOPSTAR            *  <23A6>
30 '*          von              *  <2310>
40 '*     Christian Will        *  <239F>
50 '*         fuer              *  <2309>
60 '*   SCHNEIDER CPC-WELT      *  <23D8>
70 '*    CPC 464/664/6128    ME*  <235F>
75 '*****************************  <2318>
80 ON BREAK GOSUB 3980             <0AE3>
90 GOSUB 3160:MODE 0               <0C66>
100 MODE 0                         <076C>
110 '*********                     <107B>
120 'VARIABELN                     <103D>
130 '*********                     <10A3>
140 x=11:y=23:sym=200:sc=19:sc2=CI
NT(RND*15+1):fa=-1:ru=1:ru2=1:tot=
0                                  <5D80>
150 leb=3:e=502:land=0:rud=1       <29C3>
160 tra1$=CHR$(22)+CHR$(1):tra2$=C
HR$(22)+CHR$(0)                    <2E6D>
170 '******                        <0D95>
180 'FARBEN                        <0D4F>
190 '******                        <0DBD>
200 BORDER 0:INK 0,0:INK 1,12:INK
2,15:INK 3,21:INK 4,18:INK 5,13    <2A90>
210 INK 6,19:INK 7,24,25:INK 8,3:I
NK 9,11:INK 10,12:INK 11,12:INK 12
,6                                 <3291>
220 INK 13,6                       <0A06>
230 GOSUB 3460:GOSUB 2220:GOSUB 26
50                                 <13D5>
240 ON ru GOSUB 1940,1970,1960,195
0,1980,1990,2000                   <2706>
250 '*********                     <1092>
260 'STEUERUNG                     <1043>
270 '*********                     <10BB>
280 DI:xx=x:yy=y                   <1C03>
290 FOR s=1 TO 15 STEP rud:NEXT    <180D>
300 IF INKEY(72)=0 OR INKEY(50)=0
THEN y=y-1                         <2316>
310 IF y=0 THEN y=1                <13D5>
320 IF INKEY(73)=0 OR INKEY(53)=0
THEN y=y+1                         <23BE>
330 IF y=23 THEN leb=leb-1:aa=REMA
IN(0):bb=REMAIN(1):GOSUB 1690      <3A33>
340 IF INKEY(74)=0 OR INKEY(34)=0
THEN x=x-1:sym=200                 <2D7A>
350 IF x=0 THEN x=1                <130E>
360 IF INKEY(75)=0 OR INKEY(27)=0
THEN x=x+1:sym=201                 <2D56>
370 IF x=21 THEN x=20              <1529>
380 IF xx<>x OR yy<>y THEN LOCATE
xx,yy:PRINT" "                     <2DE2>
390 LOCATE x,y:PEN 1:PRINT CHR$(sy
m)                                 <1EB6>
400 IF y=21 AND x=11 AND land=0 TH
EN GOSUB 2800                      <24A9>
410 IF y=sc2 AND x=sc THEN leb=leb
-1:aa=REMAIN(0):bb=REMAIN(1):GOSUB
 1690                              <4914>
420 IF INKEY(76)=0 OR INKEY(47)=0
THEN GOSUB 480                     <1C3C>
430 EI                             <06CF>
440 GOTO 280                       <092B>
450 '******                        <0DC6>
460 'SCHUSS                        <0DFB>
```

```
470 '******                                <0DEE>
480 DI                                     <0630>
490 IF sym=200 THEN ri=-300:ri2=-2
5 ELSE ri=300:ri2=15                       <3ACB>
500 po1=32*x-8:po2=405-16*y                <28E5>
510 PLOT po1+ri2,po2:DRAW po1+ri,p
o2,2                                       <319A>
520 PLOT po1+ri2,po2-2:DRAW po1+ri
,po2-2,2                                   <35CF>
530 SOUND 1,200,20,15,1,1                  <1482>
540 PLOT po1+ri2,po2:DRAW po1+ri,p
o2,0                                       <31CF>
550 PLOT po1+ri2,po2-2:DRAW po1+ri
,po2-2,0                                   <3502>
560 IF y=sc2 THEN IF sym=200 AND s
c<x OR sym=201 AND sc>x THEN pu=pu
+10:GOSUB 1390                             <50A2>
570 'ABZUG DES ENERGIESTREIFENS  *
**********                                 <2F45>
580 e=e-2:PLOT e,10,9:DRAW e,15:IF
 e<315 THEN aa=REMAIN(0):bb=REMAIN
(1):leb=leb-1:GOSUB 1690                   <5B7B>
590 RETURN                                 <06C5>
600 '******************                    <1AA0>
610 'GEGNERISCHES SCHIFF                   <1ABF>
620 '******************                    <1AC8>
630 DI                                     <065C>
640 LOCATE sc,sc2:PRINT" "                 <17B2>
650 z%=RND*10                              <0F51>
660 IF z%=5 THEN fa%=RND*2-1:fa2%=
RND*2-1                                    <270F>
670 sc=sc+fa%:sc2=sc2+fa2%                 <2B91>
680 IF sc<3 THEN fa%=1:sc=sc+fa%           <2724>
690 IF sc>18 THEN fa%=-1:sc=sc+fa%
                                           <296C>
700 IF sc2<3 THEN fa2%=1:sc2=sc2+f
a2%                                        <2CB1>
710 IF sc2>18 THEN fa2%=-1:sc2=sc2
+fa2%                                      <2E61>
720 LOCATE sc,sc2:PEN 6:PRINT CHR$
(sb)                                       <20D0>
730 RETURN                                 <06DC>
740 '******                                <0D0A>
750 'RAKETE                                <0D6F>
760 '******                                <0D32>
770 DI:INK 6,13                            <0C0E>
780 LOCATE sc,sc2:PRINT" "                 <17CC>
790 IF sc>x THEN sc=sc-1:bol=214           <2856>
800 IF sc<x THEN sc=sc+1:bol=220           <282F>
810 IF sc2>y THEN sc2=sc2-1:GOTO 8
30                                         <26B7>
820 IF sc2<y THEN sc2=sc2+1                <217B>
830 LOCATE sc,sc2:PEN 6:PRINT CHR$
(bol)                                      <214D>
840 RETURN                                 <06BA>
850 '******                                <0DE7>
860 'SCHUSS                                <0D1D>
870 '******                                <0D10>
880 DI                                     <0651>
890 ps1=32*sc-8:ps2=405-16*sc2             <2B35>
900 zi2=405-16*y:ziel1=x*32-12:zie
l2=RND*50-25+zi2                           <44CE>
910 tes=TEST(ziel1,ziel2)                  <2169>
920 PLOT ps1,ps2:DRAW ziel1,ziel2,
9                                          <281E>
930 PLOT ps1,ps2-1:DRAW ziel1,ziel
2-1,9                                      <2C58>
940 SOUND 4,40,20,7,,2                     <128A>
950 PLOT ps1,ps2-1:DRAW ziel1,ziel
2-1,0                                      <2C5E>
960 PLOT ps1,ps2:DRAW ziel1,ziel2,
0                                          <284A>
970 IF tes=1 THEN SOUND 4,0,100,15
,6,,25:FOR st=0 TO 5:PLOT e,10+st,
9:DRAW e-25,10+st:NEXT:e=e-25:IF e
<315 THEN aa=REMAIN(0):bb=REMAIN(1
):leb=leb-1:GOSUB 1690                     <938A>
980 RETURN                                 <06D1>
990 '*******                               <0EE7>
1000 'KOMETEN                              <0EAB>
1010 '*******                              <0E10>
1020 DI                                    <066A>
1030 LOCATE sc,sc2:PRINT" "                <17C1>
1040 sc=sc+ze:sc2=sc2+1                    <263D>
1050 IF sc<1 THEN ze=1:sc=sc+ze:IF
 ru=2 THEN sm=216:sm2=218 ELSE 107
0                                          <485D>
1060 IF sc>20 THEN ze=-1:sc=sc+ze:
IF ru=2 THEN sm=215:sm2=217 ELSE 1
070                                        <4A1C>
1070 PRINT tra1$                           <0D66>
1080 LOCATE sc,sc2:PEN 6:PRINT CHR
$(sm)                                      <20FB>
1090 LOCATE sc,sc2:PEN 13:PRINT CH
R$(sm2)                                    <2269>
1100 PRINT tra2$                           <0DA6>
1110 IF sc2>22 THEN IF ru=2 THEN G
OSUB 1270 ELSE sc2=sc2-1:GOSUB 130
0:GOSUB 1390                               <3822>
1120 RETURN                                <06EB>
1130 '*********                            <1077>
1140 '2.RAKETEN                            <10F6>
1150 '*********                            <109F>
1160 DI                                    <0682>
1170 LOCATE sc,sc2:PRINT" "                <17D8>
1180 sc=sc+ze:sc2=sc2+1                    <2654>
1190 IF sc2=23 THEN GOTO 1270              <1459>
1200 IF sc=1 THEN sc=19:GOTO 1220          <1B51>
1210 IF sc>19 THEN sc=2                    <167C>
1220 LOCATE sc,sc2:PEN 6:PRINT CHR
$(bol)                                     <215B>
1230 RETURN                                <06C6>
1240 '******************                   <1B38>
1250 'EINSCHLAG d. KOMETEN                 <1BBE>
1260 '******************                   <1B60>
1270 DI                                    <065F>
1280 LOCATE sc,sc2:PEN 1:PRINT CHR
$(219)                                     <1D91>
1290 SOUND 4,0,200,15,7,,25                <1566>
1300 IF sc=10 OR sc=11 OR sc=12 TH
EN land=1                                  <2A61>
1310 IF ru<>6 THEN sc=CINT(RND*19+
1):sc2=1:ze%=RND*1                         <33B9>
1320 IF ru=2 THEN IF ze%=0 THEN ze
```

```
=-1:sm=215:sm2=217 ELSE ze=1:sm=21
6:sm2=218                              <4E98>
1330 IF ru=5 THEN sc2=CINT(RND*10+
4):z%=RND*1:IF z%=1 THEN sc=19:ze=
-1:bol=214 ELSE sc=1:ze=1:bol=220      <69C7>
1340 IF ru=6 THEN sm=221:sm2=221       <208A>
1350 RETURN                            <06B8>
1360 '**********************           <1C55>
1370 'EXPLOSION des GEGNERS            <1C36>
1380 '**********************           <1C7D>
1390 DI:ge=ge+1                        <144A>
1400 LOCATE sc,sc2:PRINT" "            <17A5>
1410 LOCATE 1,1:PRINT tra1$            <128B>
1420 LOCATE sc,sc2:PEN 8:PRINT CHR
$(209)                                 <1D8E>
1430 LOCATE sc,sc2+1:PEN 8:PRINT C
HR$(210)                               <1F07>
1440 LOCATE sc,sc2:PEN 2:PRINT CHR
$(207)                                 <1D9E>
1450 LOCATE sc,sc2+1:PEN 2:PRINT C
HR$(208)                               <1F06>
1460 '****  VARIABELZUWEISUNG  ***
*                                      <245E>
1470 s1=sc:s2=sc2                      <1D53>
1480 z%=RND*1:IF z%=0 THEN sc=2 EL
SE sc=19                               <28B9>
1490 sc2=CINT(RND*15+2):IF s2=sc2
THEN 1490                              <2963>
1500 IF ru=5 THEN sc2=CINT(RND*8+6
):z%=RND*1:IF z%=1 THEN ze=-1:bol=
214 ELSE ze=1:bol=220                  <57F9>
1510 IF ru=2 OR ru=6 THEN sc2=1:sc
=CINT(RND*19+1)                        <30F8>
1520 IF ru=6 THEN sm=221:sm2=221       <20F1>
1530 SOUND 2,0,104,15,3,,10            <15CB>
1540 LOCATE 1,1:PRINT tra2$            <1295>
1550 AFTER 5,3 GOSUB 1620              <0D00>
1560 IF ge=zahl THEN ge=0:ru=ru+1:
ru2=ru2+1:ON ru GOSUB 1940,1970,19
60,1950,1980,1990,2000                 <5C17>
1570 IF ru=8 THEN leb=leb+1:GOSUB
2420:ru=1:rud=rud+1:ON ru GOSUB 19
40,1970,1960,1950,1980,1990,2000       <5D94>
1580 RETURN                            <0685>
1590 '**********************           <1A60>
1600 'LOESCHEN DER EXPOL.              <1AA0>
1610 '**********************           <1A88>
1620 DI                                <061C>
1630 LOCATE s1,s2:PRINT" "             <16AE>
1640 LOCATE s1,s2+1:PRINT" "           <18DA>
1650 RETURN                            <0610>
1660 '**********************           <1D07>
1670 'EXPLOSION des SCHIFFES           <1D85>
1680 '**********************           <1D2F>
1690 DI                                <06A8>
1700 LOCATE xx,yy:PRINT" "             <16FE>
1710 PRINT tra1$                       <0D69>
1720 LOCATE x,y:PEN 8:PRINT CHR$(2
09)                                    <1AF0>
1730 LOCATE x,y+1:PEN 8:PRINT CHR$
(210)                                  <1C43>
1740 LOCATE x,y:PEN 2:PRINT CHR$(2
07)                                    <1AE0>
1750 LOCATE x,y+1:PEN 2:PRINT CHR$
(208)                                  <1C43>
1760 PRINT tra2$:e=502:land=0          <2086>
1770 z%=RND*1:IF z%=0 THEN sc=2 EL
SE sc=19                               <28FE>
1780 sc2=CINT(RND*15+2):IF s2=sc2
THEN 1490                              <29A8>
1790 SOUND 2,0,500,15,5,,30            <16EF>
1800 FOR var=1 TO 4000:NEXT            <14AB>
1810 '******************               <19CE>
1820 'ANZAHL DER SCHIFFE               <19DE>
1830 '******************               <19F6>
1840 tot=0                             <0D59>
1850 IF leb=0 THEN GOTO 3040           <133A>
1860 GOSUB 2930                        <091B>
1870 GOSUB 2250                        <09E7>
1880 GOSUB 2650                        <0980>
1890 ON ru GOSUB 1940,1970,1960,19
50,1980,1990,2000                      <27F1>
1900 RETURN                            <0606>
1910 '***************                  <1510>
1920 'UNTERPROGRAMME                   <1587>
1930 '***************                  <1538>
1940 GOSUB 2050:EVERY 12-rud,0 GOS
UB 630:EVERY 150,1 GOSUB 880:RETUR
N                                      <26AA>
1950 GOSUB 2070:EVERY 11-rud,0 GOS
UB 630:EVERY 130,1 GOSUB 880:RETUR
N                                      <26A2>
1960 GOSUB 2090:EVERY 11-rud,0 GOS
UB 770:aa=REMAIN(1):RETURN             <28F4>
1970 GOSUB 2130:EVERY 11-rud,0 GOS
UB 1020:aa=REMAIN(1):RETURN            <282F>
1980 GOSUB 2160:EVERY 11-rud,0 GOS
UB 1160:aa=REMAIN(1):RETURN            <28D8>
1990 GOSUB 2190:aa=REMAIN(1):EVERY
 10-rud,0 GOSUB 1020:RETURN            <28AD>
2000 GOSUB 2110:EVERY 10-rud,0 GOS
UB 630:EVERY 110,1 GOSUB 880:RETUR
N                                      <2674>
2010 '********************             <1AAA>
2020 'AUSSEHEN DER GEGNER              <1A22>
2030 '********************             <1AD2>
2040 '1. GEGNER                        <1044>
2050 sb=206:INK 6,19:zahl=20:RETUR
N                                      <2041>
2060 '2. GEGNER                        <1069>
2070 sb=213:INK 6,6:zahl=20:RETURN
                                       <1F40>
2080 '3.GEGNER                         <0F0C>
2090 INK 6,14:zahl=12:RETURN           <17EF>
2100 '4.GEGNER                         <0F36>
2110 sb=222:INK 6,11:zahl=20:RETUR
N                                      <2010>
2120 'KOMETEN                          <0E71>
2130 INK 6,15:INK 13,6:zahl=20:sc=
CINT(RND*19+1):sc2=1:ze%=RND*1:IF
ze%=0 THEN ze=-1:sm=215:sm2=217 EL
SE ze=1:sm=216:sm2=218                 <829D>
2140 RETURN                            <06E7>
2150 'RAKETEN                          <0E2D>
2160 INK 6,6:zahl=15:sc2=CINT(RND*
8+6):z%=RND*1:IF z%=1 THEN ze=-1:s
```

```
c=19:bol=214 ELSE ze=1:bol=220:sc=
1                                   (6F7B)
2170 RETURN                         (0622)
2180 'BOMBEN                        (0DD5)
2190 INK 6,13:INK 13,13:zahl=18:sc
=CINT(RND*19+1):sc2=1:ze%=RND*1:IF
 ze%=0 THEN ze=-1:sm=221:sm2=221 E
LSE ze=1:sm=221:sm2=221             (8366)
2200 RETURN                         (065E)
2210 '*****************             (17B7)
2220 'BILDSCHIRMAUFBAU              (174F)
2230 '*****************             (17DF)
2240 'BAUME   *******               (1688)
2250 FOR ba=1 TO 20                 (1036)
2260 IF ba=10 OR ba=11 OR ba=12 TH
EN 2290                             (2461)
2270 zuf%=RND*1:IF zuf%=0 THEN sy=
202:pe=4 ELSE sy=203:pe=3           (3D9B)
2280 LOCATE ba,23:PEN pe:PRINT CHR
$(sy)                               (206D)
2290 NEXT                           (06AE)
2300 PLOT 0,30:DRAW 640,30,10       (1583)
2310 'RAMPE   *****                 (146C)
2320 FOR la=0 TO 30                 (10D5)
2330 PLOT 304,60:DRAW 304-la,31,5   (1CBF)
2340 PLOT 360,60:DRAW 360+la,31,5   (1CFC)
2350 NEXT                           (0628)
2360 PLOT 304,60:DRAW 360,60        (1433)
2370 FOR lee=1 TO 500:NEXT          (14F6)
2380 LOCATE 11,23:PEN 1:PRINT CHR$
(200)                               (161E)
2390 'STERNE   *****                (15D0)
2400 FOR var=1 TO 100:PLOT RND*640
,RND*300+100,7:NEXT                 (2787)
2410 'ANZAHL DER SCHIFFE  ********
*                                   (243F)
2420 IF leb=1 THEN 2480             (128C)
2430 FOR anzahl=1 TO leb-1          (1A4E)
2440 LOCATE anzahl*2-1,25:PEN 11:P
RINT CHR$(201)                      (22DC)
2450 NEXT                           (06F0)
2460 IF ru=8 THEN RETURN            (0FB1)
2470 'ENERGIESTREIFEN  ***********
**                                  (25BA)
2480 LOCATE 9,25:PEN 12:PRINT CHR$
(211);CHR$(212)                     (1DFC)
2490 FOR stt=308 TO 500             (1471)
2500 PLOT stt,10,12:DRAW stt,15     (1D39)
2510 NEXT                           (0667)
2520 PLOT 306,0:DRAW 306,24,9       (15AD)
2530 IF ta=1 THEN ta=0:RETURN       (1757)
2540 '*********************         (1B65)
2550 'START und LANDEPHASE          (1BF4)
2560 '*********************         (1B8E)
2570 DI                             (068D)
2580 a$=INKEY$:IF a$<>""THEN 2590
ELSE 2580                           (1E04)
2590 SOUND 2,800,270,4,,4           (148A)
2600 FOR lee=0 TO 24                (1115)
2610 FOR st=1 TO 80:NEXT            (12D1)
2620 PLOT 334-lee,60,0:PLOT 332+le
e,60,0                              (26D5)
2630 NEXT                           (0659)
2640 RETURN                         (06D1)
2650 FOR var=1 TO 100:NEXT          (130E)
2660 SOUND 1,0,280,10,2,,1          (1590)
2670 FOR po=1 TO 12                 (1074)
2680 FOR var=1 TO 150:NEXT          (136F)
2690 IF s=205 THEN s=204 ELSE s=20
5                                   (1E28)
2700 LOCATE 11,23-po:PEN 1:PRINT C
HR$(200)                            (1C26)
2710 LOCATE 11,24-po:PEN 2:PRINT C
HR$(s)                              (1E27)
2720 IF po>1 THEN LOCATE 11,25-po:
PRINT" "                            (1F3D)
2730 NEXT                           (0620)
2740 LOCATE 11,11:PRINT" "          (10D2)
2750 LOCATE 11,12:PRINT" "          (105F)
2760 x=11:y=11                      (149D)
2770 PLOT 304,60:DRAW 360,60,5      (1642)
2780 RETURN                         (06E8)
2790 '********** KLEINES UNTERPROG
RAMM  *********                     (328E)
2800 DI:aa=REMAIN(0):ab=REMAIN(1)   (1E64)
2810 LOCATE sc,sc2:PRINT" "         (17AF)
2820 GOSUB 2570                     (0961)
2830 LOCATE 11,21:PRINT" ":LOCATE
11,23:PRINT CHR$(sym)               (230A)
2840 PLOT 304,60:DRAW 360,60,5      (16CF)
2850 FOR var=1 TO 1500:NEXT         (14AE)
2860 SOUND 1,180,140,3,,3:ta=1:GOS
UB 2490                             (1FB9)
2870 FOR var=1 TO 2000:NEXT:tot=0:
e=502:sc=19:sc2=CINT(RND*15+1)      (4293)
2880 GOSUB 2570:GOSUB 2650          (0E5E)
2890 ON ru GOSUB 1940,1970,1960,19
50,1980,1990,2000:RETURN            (2965)
2900 '**************                (146D)
2910 'ZWISCHENSTAND                 (146A)
2920 '**************                (1495)
2930 DI:MODE 1:INK 3,11:PEN 3       (128C)
2940 LOCATE 2,5:PRINT"Leider haben
 Sie ein Schiff aus Ihrer"          (3269)
2950 LOCATE 2,7:PRINT"Staffel verl
oren. Ihr Punktestand"              (2ED0)
2960 LOCATE 2,9:PRINT"liegt bei";p
u;"Punkten. Wenn Sie"               (3050)
2970 LOCATE 2,11:PRINT"bereit sind
 ein naechstes Schiff zu"           (31A5)
2980 LOCATE 2,13:PRINT"starten dru
ecken Sie die Taste ))j((."         (3300)
2990 a$=INKEY$                      (0C99)
3000 IF a$="j"OR a$="J"THEN MODE 0
:INK 3,21:RETURN ELSE GOTO 2990     (2837)
3010 '****                          (0B21)
3020 'ENDE                          (0B10)
3030 '****                          (0B49)
3040 MODE 1:PEN 3:INK 3,11          (106B)
3050 LOCATE 2,5:PRINT"Leider haben
 Sie die";ru2;".Runde nicht"        (37B5)
3060 LOCATE 2,7:PRINT"ueberlebt. I
hr Punktestand lag bei"             (2F74)
3070 LOCATE 1,9:PRINT pu;"Punkten.
"                                   (1B8F)
3080 LOCATE 2,11:PRINT"Wollen Sie
```

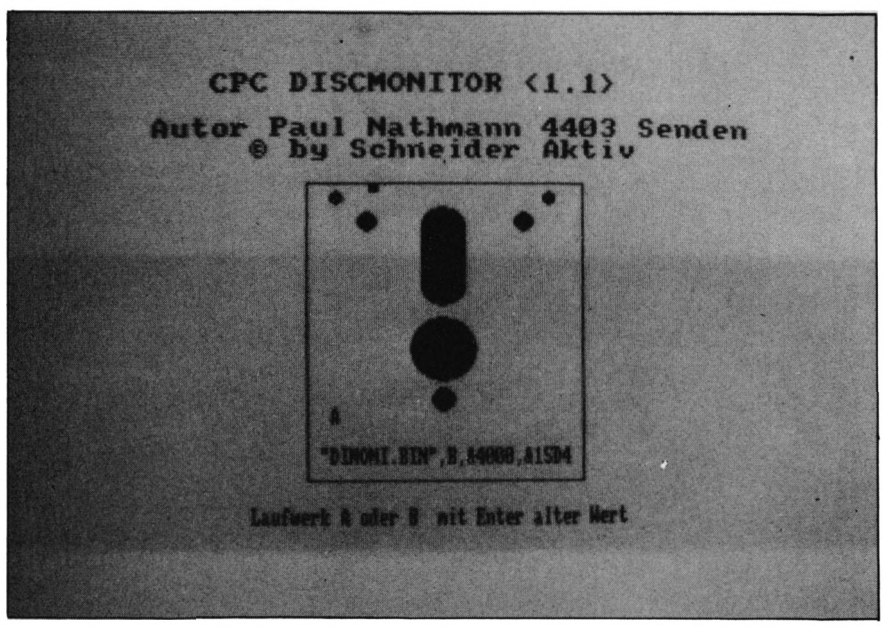
```
nocheinmal spielen (j/n)"          <31C3>
3090 a$=INKEY$                     <0C63>
3100 IF a$="j"OR a$="J"THEN RUN    <198F>
3110 IF a$="n"OR a$="N"THEN END    <1958>
3120 GOTO 3090                     <0923>
3130 '*********                    <101F>
3140 'TITELBILD                    <109F>
3150 '*********                    <1047>
3160 MODE 1:p=2                    <0ECF>
3170 BORDER 0:INK 0,0:INK 2,24:INK
 3,2:INK 1,24,2:TAG                <21FB>
3180 SPEED INK 5,5                 <0A6B>
3190 PLOT 190,305,3:PRINT"TOPSTAR
von";                              <1EDE>
3200 PLOT 190,289,3:PRINT"--------
";                                 <1BA4>
3210 PLOT 190,241:PRINT"Christian
Will";                             <1E21>
3220 TAGOFF:PRINT CHR$(22);CHR$(1)
                                   <149C>
3230 LOCATE 13,7:PEN 2:PRINT"TOPST
AR von"                            <1C62>
3240 LOCATE 13,8:PEN 2:PRINT"-----
---"                               <19DF>
3250 LOCATE 13,11:PEN 2:PRINT"Chri
stian Will"                        <202C>
3260 FOR a=1 TO 3000:NEXT          <1228>
3270 '**********                   <117D>
3280 'ERKLAERUNG                   <113D>
3290 '**********                   <11A5>
3300 CLS                           <06FF>
3310 LOCATE 14,1:PEN 1:PRINT"Erkla
erung"                             <1BF2>
3320 FOR y=3 TO 23                 <0F63>
3330 READ stri$:TAG                <0FF5>
3340 PLOT 18,417-(y*16),3:PRINT st
ri$;:TAGOFF                        <24CB>
3350 PRINT CHR$(22);CHR$(1)        <1208>
3360 LOCATE 2,y:PEN 2:PRINT stri$;
CHR$(22);CHR$(0)                   <2521>
3370 NEXT                          <0624>
3380 LOCATE 9,25:PEN 1:PRINT"Weite
r mit Leer-Taste"                  <26AE>
3390 'DATAS ***************        <1BB2>
3400 DATA "Sie haben die Aufgabe d
ie Erde vor","ausserirdischen Angr
eifern und Kometen","zu schuetzen.
 Aber achten Sie auf Ihre"         <7D62>
3410 DATA "Energiescala, denn jede
r Schuss den Sie","abgeben und jed
er feindliche Strahl","der Sie tri
fft, kostet Energie. Um neue"      <8024>
3420 DATA "Energie zu bekommen lan
den Sie auf","Ihrer Startrampe die
 nur dann","funktioniert, wenn Sie
 nicht durch","Bomben oder Kometen
 zerstoert wurde."                 <97B7>
3425 DATA "Ihren Punktestand erfah
ren Sie nach ","jedem neuen Schiff
."                                 <43A3>
3430 DATA " ","Sie koennen mit Joy
stick oder","Tastatur spielen:","
","    Auf=R","     Ab=F","Rechts=P"
,"  Links=O","  Feuer=Leer-Taste"   <82F0>
3440 IF INKEY$=" "THEN RETURN ELSE
 3440                              <13BD>
3450 '******                       <0D42>
3460 'SOUNDS                       <0D0C>
3470 '******                       <0D6A>
3480 ENT 2,20,1,1                  <0E52>
3490 ENT 1,20,-10,1                <1006>
3500 ENT 3,100,-2,2.5              <1412>
3510 ENT 4,100,-3,2.7              <146F>
3520 ENV 1,10,-1,2                 <0F50>
3530 ENV 2,3,-1,70                 <0F78>
3540 ENV 3,13,-1,8                 <0F1A>
3550 ENV 5,15,-1,34                <1046>
3560 ENV 6,10,-1,10                <10FA>
3570 ENV 7,5,-3,40                 <0F27>
3580 '*******                      <0E2E>
3590 'SYMBOLE                      <0E02>
3600 '*******                      <0E57>
3610 SYMBOL AFTER 32               <097D>
3620 'RAUMSCHIFF                   <115F>
3630 SYMBOL 200,2,2,6,30,63,255,62
,0                                 <1C6E>
3640 SYMBOL 201,64,64,96,120,252,2
55,124                             <1DA3>
3650 'BAUME                        <0C79>
3660 SYMBOL 202,16,56,124,124,56,1
6,16,56                            <2066>
3670 SYMBOL 203,16,56,16,56,124,16
,16,56                             <2071>
3680 'FLAMME                       <0D1A>
3690 SYMBOL 204,42,34,20,12,8,0,0,
0                                  <1C1C>
3700 SYMBOL 205,42,34,34,22,20,20,
8,8                                <1E14>
3710 '1.UFO                        <0C3F>
3720 SYMBOL 206,0,60,126,219,255,0
,0,0                               <1C11>
3730 '2.UFO                        <0CA7>
3740 SYMBOL 213,0,24,60,126,195,12
6,60,24                            <1F30>
3750 '3.UFO                        <0C0E>
3760 SYMBOL 214,0,0,1,67,255,67,1,
0                                  <1B88>
3770 SYMBOL 220,0,0,128,194,255,19
4,128,0                            <1DD8>
3780 '4.UFO                        <0C8A>
3790 SYMBOL 222,60,126,255,36,36,2
55,126,60                          <203C>
3800 'KOMETEN                      <0E96>
3810 SYMBOL 215,7,30,126,124,220,2
16,248,112                         <1F0F>
3820 SYMBOL 216,224,120,126,62,59,
27,31,14                           <2056>
3830 SYMBOL 217,0,0,0,48,80,80,32,
0                                  <1CF3>
3840 SYMBOL 218,0,0,0,12,10,10,4,0
                                   <1BE7>
3850 'STERNE                       <0DB1>
3860 SYMBOL 221,3,3,0,3,3,3,3      <1607>
3870 'EINSCHLAG                    <1075>
3880 SYMBOL 219,0,0,0,32,112,170,9
3,170                              <1D56>
```

```
3890 'EXPOLSION                          <1007>
3900 SYMBOL 207,24,36,36,98,32,133
,147,79                                  <2060>
3910 SYMBOL 208,66,74,133,145,66,4
4,56,16                                  <2098>
3920 SYMBOL 209,0,24,24,19,108,126
,102,50                                  <1F1A>
3930 SYMBOL 210,30,124,112,94,94,1
12,0,0                                   <1E50>
3940 'SONSTIGES                          <10B7>
3950 SYMBOL 211,0,7,4,4,7,4,4,7          <18DF>
3960 SYMBOL 212,0,0,0,0,0,0,32           <1716>
3970 RETURN                              <0639>
3980 BORDER 1:INK 0,1:INK 1,26:PEN
 1:MODE 2:STOP                           <1A65>
```

Diskettenmonitor

Der Diskettenmonitor wurde von unserem Autor Paul Nathmann zu 100 Prozent in Maschinensprache geschrieben. Die extrem hohe Arbeitsgeschwindigkeit macht dies deutlich. Lassen Sie sich von den gewaltigen DATA Zeilen nicht abschrecken, denn mit Diskmonitor erhalten Sie ein äußerst komfortables Programm zur Bearbeitung Ihrer Disketten.

Unser Disketten-Monitor bietet neben den „üblichen" Funktionen wie Sektor laden oder Sektor ändern weitere nützliche Routinen, etwa Drukkerausgabe, Berechnung von Spur und Sektor und Informationen zum Directory-Bereich.
Nach Programmstart erscheint zunächst ein Einschaltbild. Hier können Sie ein Laufwerk auswählen. Mit der ENTER-Taste wird die aktuelle Floppy übernommen. Jetzt erst kommen Sie in den Diskettenmonitor. Der Bildschirmaufbau sieht folgendermaßen aus: In der ersten Zeile werden der aktuelle Track, Sektor und Blocknummer ausgegeben. Auch die jeweiligen Recordteile und das Format der Diskette (DATA oder CPM) werden genannt. Im unteren Viertel des Monitors haben Sie ständig alle Funktionen von Diskmonitor im Blick. Den Rest des Bildschirms nimmt der aktuelle Sektor in Anspruch; er wird in Hex-Zahlen und ASCII-Codes ausgegeben.
Mit CTRL-E wird der Monitor verlassen, der CPC springt in den



Ready-Modus. Mit der Taste C können Sie das Laufwerk ändern oder sich schlicht und einfach ein Directory ausgeben lassen. Taste B steht für Spur und Sektor berechnen. Dies wird nach Eingabe der Blocknummer erledigt und das Ergebnis ausgegeben. Mit P wird ein ganzer Sektor gedruckt. Ist der Drucker nicht bereit, wird nach fünf Sekunden ins Programm zurückgesprungen. Durch Betätigen der Taste I wird der Aufbau des Directories angezeigt – eine äußerst nützliche Sache, wenn Sie mit Diskmonitor im Directory herumPOKEn wollen.
Taste L steht für Sektor laden, Taste S für Sektor auf Diskette schreiben. Da ein vollständiger Sektor nicht komplett auf den

Monitor paßt, wird er vom Programm in zwei Seiten aufgeteilt. Mit CTRL-CURSOR LINKS wird eine halbe Sektoranzeige zurück-, mit CTRL-CURSOR RECHTS eine halbe Sektoranzeige vorgeblättert. Mit dieser Funktion können Sie die ganze Diskette durchforsten.
Mit Taste A kommen Sie ins Herz von Diskmonitor, nämlich in den Menüpunkt Verändern des Diskettenspeichers. Es erscheint ein neues Menü. Mit den Cursorpfeilen geht es in alle Richtungen. Beim Drücken von COPY wechselt der Cursor vom Hex- ins ASCII-Feld oder zurück. Im Hex-Feld können nur Hexadezimal-Werte eingegeben werden, im ASCII-Feld die ASCII-Werte null bis 127. Mit dem Pfund-Zeichen läßt sich in beiden Feldern das Füllbyte (&E5) eingeben. Halten Sie die Taste gedrückt, können Sie einen Sektor schnell überschreiben beziehungsweise reinigen.
Mit CTRL-TAB wird das siebte Bit des jeweiligen Bytes gewechselt. Dies eignet sich gut zum Ändern von Byte neun und zehn im Directory:

File R/O (Read Only): Bit sieben.
File R/W (Read Write): Bit sieben gleich Null
File SYS (unsichtbar): Bit sieben gleich Eins
File DIR (sichtbar): Bit sieben gleich Null

Mit ESC wird der Änderungsmodus wieder verlassen, nachdem gefragt wurde, ob abgespeichert werden soll oder nicht.
Die Druckerausgabe ist auf Entwurfsdruck eingestellt. Wer lieber eine saubere NLQ-Ausgabe wünscht, dem kann mit POKE &4ACC,&C9 geholfen werden. POKE &4ACC,&3E wechselt wieder auf Entwurfsdruck. Vor dem Start kann mit CALL &4989: POKE &4B58,1 der blinkende Cursor ausgeschaltet werden. CALL &4948 bringt ihn wieder zum Blinken.
Tippen Sie den BASIC-Lader "DISMONI.BAS" ein und speichern Sie ihn auf Diskette ab. Dieses Programm enthält eine spezielle Laderoutine für den CPC 6128, mit der eine bestimmte Bank gewählt werden kann. Für den CPC 464 und 664 wird ganz normal das Binärprogramm geladen. Als nächstes muß der Datalader "DISMONI.LAD" eingegeben und vorsichtshalber abgespeichert werden. Danach wird er mit RUN gestartet. Dadurch wird ein Binärfile abgespeichert, und der CPC springt in den Ready-Modus. Jetzt tippen Sie "GOTO 662" (ENTER) ein. Das Binärfile wird wieder eingeladen, an einem anderen Speicherplatz (&4000) abgelegt und wieder abgespeichert. Beachten Sie, daß **nur** die zweite Version lauffähig ist! *TB/Paul Nathmann* □

```
1 '******************************  <24B1>
2 '*       DISKMONITOR          *  <2420>
3 '*            VON             *  <24D0>
4 '*       PAUL NATHMANN        *  <24B7>
5 '*            FUER            *  <2424>
6 '*    SCHNEIDER CPC-WELT      *  <24EF>
7 '*     CPC 464/664/6128    TB*  <24DB>
8 '******************************  <24BF>
10 '                                <071F>
20 MODE 2                           <07D5>
30 IF PEEK(&BB4F)=&74 THEN 70:' CP
C 6128                              <21CD>
40 GOSUB 170                        <0973>
50 PRINT"Starten mit CALL &4000"    <1E05>
60 END                              <06DA>
70 INPUT"In welche Bank 4-7 sonst
normal ";b$                         <2D27>
80 b=VAL(b$):IF b<4 OR b>7 THEN 40
                                    <25AD>
90 e=b+&C0                          <12F3>
100 OUT &7F00,e:' bei CPC 6128 Ban
k einschalten                       <2F20>
110 h=HIMEM:GOSUB 170               <115F>
120 CALL &BD5B:' wieder normale Ba
nk, call ohne Parameter AKKU = 0    <3E27>
130 MEMORY h:PRINT"Starten mit"     <194E>
140 t$="MEMORY &3FFF:call &BD5B"+M
ID$(",1,2,3,4,5,6,7",1,b*2)+":CALL
 &4000"                             <4EB1>
150 PRINT t$                        <0A4D>
160 END                             <06A3>
170 MEMORY &3FFF                    <095D>
180 LOAD"DISMONI.BIN",&4000         <17A4>
190 RETURN                          <06A2>
```

```
100 '******************************  <2234>
110 '* DATALADER ZU DISKMONITOR*     <2253>
120 '*   ERZEUGT <DIMONI.BIN>  *     <22C3>
130 '*           VON           *     <2275>
140 '*      PAUL NATHMANN      *     <22C0>
150 '*          FUER           *     <2228>
160 '*   SCHNEIDER CPC-WELT    *     <2247>
170 '*    CPC 464/664/6128   TB*     <22DE>
180 '******************************  <22D4>
200 MEMORY &7FFF                    <0998>
650 a=&8000:e=&95D4:zb=1000:e=e+1   <2CB9>
660 FOR i=a TO e:IF i=e THEN SAVE"
DISMONI.BIN",B,&8000,&15D4:PRINT"J
etzt eingeben: goto 662":DELETE 67
0-                                  <5D2A>
661 IF i<e THEN GOTO 670            <1497>
662 MEMORY &3FFF:LOAD"DISMONI.BIN"
,&4000:SAVE"DISMONI.BIN",B,&4000,&
15D4:END                            <3A6F>
```

Diskettenmonitor

```
670 READ d$:POKE i,VAL("&"+d$)     <1D5A>
730 IF i<e THEN NEXT i             <1526>
1001 DATA ED,73,6E,4B,CD,00,BB,0E <1E43>
1002 DATA 00,CD,15,B9,21,70,4B,CD <1E07>
1003 DATA D4,BC,D2,17,43,3E,FF,CD <1E57>
1004 DATA 1B,00,21,71,4B,CD,D4,BC <1E1A>
1005 DATA D2,17,43,22,5F,4B,79,32 <1E68>
1006 DATA 61,4B,21,72,4B,CD,D4,BC <1E4F>
1007 DATA D2,17,43,22,62,4B,79,32 <1EFB>
1008 DATA 64,4B,11,7B,4B,CD,D3,45 <1E02>
1009 DATA 3A,58,4B,B7,20,03,CD,48 <1E70>
1010 DATA 49,ED,7B,6E,4B,AF,32,9F <1E53>
1011 DATA A8,32,DF,A8,32,5D,4B,32 <1E4B>
1012 DATA 6B,4B,32,01,A7,CD,57,41 <1EA1>
1013 DATA CD,76,41,3E,15,CD,5A,BB <1E5B>
1014 DATA CD,50,41,F5,3E,06,CD,5A <1E7B>
1015 DATA BB,F1,CA,2C,43,CD,41,49 <1EA4>
1016 DATA 06,02,28,19,06,00,3E,C1 <1EA6>
1017 DATA BE,28,12,11,D9,4C,CD,D3 <1E33>
1018 DATA 45,11,6B,4C,CD,D3,45,CD <1EBE>
1019 DATA C8,45,C3,49,40,78,32,5C <1E78>
1020 DATA 4B,32,5D,4B,3E,01,32,5E <1EA7>
1021 DATA 4B,CD,BA,43,CD,CF,43,11 <1E4C>
1022 DATA 26,4E,CD,D3,45,CD,84,BB <1EA6>
1023 DATA CD,C8,45,FE,FA,28,2D,FE <1E2E>
1024 DATA FB,28,2F,FE,05,28,64,E6 <1E39>
1025 DATA DF,FE,4C,28,2B,FE,42,28 <1E76>
1026 DATA 30,FE,49,28,32,FE,41,28 <1E38>
1027 DATA 34,FE,50,28,42,FE,53,28 <1E14>
1028 DATA 35,FE,43,28,40,00,00,00 <1ED8>
1029 DATA 00,00,18,C9,CD,F2,45,C3 <1E9F>
1030 DATA AD,40,CD,2A,46,C3,AD,40 <1EDB>
1031 DATA CD,61,46,D2,49,40,C3,A7 <1E10>
1032 DATA 40,CD,86,46,C3,A7,40,CD <1E1A>
1033 DATA 60,4A,C3,A7,40,CD,DF,46 <1E35>
1034 DATA D2,49,40,C3,A7,40,CD,25 <1EC8>
1035 DATA 49,D2,49,40,C3,A7,40,CD <1E8B>
1036 DATA 96,49,C3,A7,40,CD,34,41 <1EA8>
1037 DATA C3,75,40,CD,6C,BB,21,70 <1E66>
1038 DATA 4B,CD,D4,BC,D2,17,43,AF <1E15>
1039 DATA CD,1B,00,C9,CD,57,41,CD <1EFA>
1040 DATA 76,41,21,05,01,CD,8B,44 <1E91>
1041 DATA CD,50,41,CA,2C,43,11,BE <1E6D>
1042 DATA 4C,CD,D3,45,CD,C8,45,C9 <1ED5>
1043 DATA 11,00,60,CD,9B,BC,C9,11 <1E55>
1044 DATA 83,4B,CD,D3,45,21,0F,C0 <1E99>
1045 DATA 3E,15,CD,F0,42,21,A9,C0 <1E47>
1046 DATA 3E,1F,CD,F0,42,21,03,C1 <1E9E>
1047 DATA 3E,14,CD,F0,42,C9,21,54 <1E87>
1048 DATA 00,11,C5,00,D5,CD,EA,BB <1E46>
1049 DATA D1,21,3E,01,E5,CD,F6,BB <1ECB>
1050 DATA E1,11,A9,01,D5,CD,F6,BB <1E67>
1051 DATA D1,21,54,00,E5,CD,F6,BB <1EC0>
1052 DATA E1,11,C6,00,D5,CD,F6,BB <1EED>
1053 DATA D1,21,3E,01,E5,CD,F6,BB <1ED3>
1054 DATA E1,11,AA,01,D5,CD,EA,BB <1E5A>
1055 DATA D1,21,54,00,CD,F6,BB,0E <1E08>
1056 DATA 0D,41,21,D7,00,16,01,AF <1E90>
1057 DATA DD,21,82,55,CD,C8,42,3C <1E46>
1058 DATA 41,DD,21,9A,55,CD,C8,42 <1E43>
1059 DATA 0E,05,41,21,9F,00,AF,DD <1E21>
1060 DATA 21,9C,55,CD,C8,42,41,DD <1E10>
1061 DATA 21,A4,55,3C,CD,C8,42,0E <1EAC>
1062 DATA 04,41,21,28,01,16,00,AF <1EA4>
1063 DATA DD,21,A6,55,CD,C8,42,41 <1ED1>
1064 DATA DD,21,AC,55,3C,CD,C8,42 <1E97>
1065 DATA 41,21,28,01,16,01,AF,DD <1E68>
1066 DATA 21,AE,55,CD,C8,42,41,DD <1E1E>
1067 DATA 21,B4,55,3C,CD,C8,42,0E <1ED8>
1068 DATA 03,41,21,38,01,AF,DD,21 <1E1F>
1069 DATA B6,55,CD,C8,42,41,DD,21 <1EBE>
1070 DATA BA,55,3C,CD,C8,42,41,21 <1E23>
1071 DATA 38,01,16,00,AF,DD,21,BC <1EBE>
1072 DATA 55,CD,C8,42,41,DD,21,C0 <1EAE>
1073 DATA 55,3C,CD,C8,42,0E,08,41 <1ED6>
1074 DATA 21,2D,01,16,01,AF,DD,21 <1EC5>
1075 DATA C2,55,CD,C8,42,41,DD,21 <1ED5>
1076 DATA D0,55,21,EF,00,3C,CD,C8 <1EB3>
1077 DATA 42,11,47,01,06,25,C5,D5 <1E86>
1078 DATA 21,F0,00,CD,EA,BB,D1,D5 <1E03>
1079 DATA 21,1C,01,CD,F6,BB,D1,1B <1E43>
1080 DATA C1,10,EB,11,0E,4C,CD,D3 <1E6A>
1081 DATA 45,3E,83,CD,D9,45,CD,58 <1E28>
1082 DATA 43,CD,C8,45,FE,0D,28,1D <1EC2>
1083 DATA E6,5F,FE,41,28,04,FE,42 <1ED2>
1084 DATA 20,EF,CB,FF,21,5B,4B,77 <1EE4>
1085 DATA CD,D4,BC,D2,17,43,AF,CD <1E9B>
1086 DATA 1B,00,DA,2C,43,11,5D,4C <1EDE>
1087 DATA CD,D3,45,3A,02,A7,C6,41 <1EE0>
1088 DATA CD,5A,BB,11,6B,4C,CD,D3 <1E44>
1089 DATA 45,CD,C8,45,CD,84,BB,C9 <1EBE>
1090 DATA C5,F5,E5,DD,5E,00,D5,CD <1ECD>
1091 DATA C0,BB,D1,DD,5E,01,D5,CD <1ED1>
1092 DATA F6,BB,D1,E1,2D,2D,F1,B7 <1E35>
1093 DATA 20,08,DD,23,DD,23,C1,10 <1EB7>
1094 DATA DF,C9,DD,2B,DD,2B,18,F6 <1EE1>
1095 DATA 11,00,08,E5,F5,4A,7E,23 <1E9C>
1096 DATA 06,04,1F,CB,1E,CB,2E,10 <1EF0>
1097 DATA F9,2B,06,04,1F,CB,1E,CB <1EF5>
1098 DATA 2E,10,F9,19,0D,20,E7,F1 <1EB9>
1099 DATA E1,23,23,3D,20,DD,C9,CD <1E66>
1100 DATA 57,41,11,FA,4C,CD,D3,45 <1E56>
1101 DATA CD,58,43,CD,C8,45,CD,23 <1E36>
1102 DATA 41,C3,00,00,F5,CD,57,41 <1EF4>
1103 DATA CD,58,43,F1,CB,77,28,0E <1E83>
1104 DATA 11,24,55,CB,4F,20,0A,11 <1E5F>
1105 DATA 53,55,CB,47,20,03,11,ED <1E40>
1106 DATA 54,CD,D3,45,11,BE,4C,CD <1E9D>
1107 DATA D3,45,CD,C8,45,C3,49,40 <1EAC>
1108 DATA 3E,01,01,00,00,CD,32,BC <1E29>
1109 DATA C9,01,02,02,21,59,4B,CD <1EE5>
1110 DATA D4,44,20,08,3A,5D,4B,CD <1E80>
1111 DATA 8C,45,B7,C9,11,00,00,CD <1EAC>
1112 DATA 1B,45,FE,28,30,05,32,5D <1E9D>
1113 DATA 4B,B7,C9,CD,70,44,37,C9 <1EFB>
1114 DATA 01,01,01,21,59,4B,CD,D4 <1EC3>
1115 DATA 44,20,06,3A,5E,4B,C3,8C <1E96>
1116 DATA 45,2B,CD,A5,44,7E,D6,30 <1EAD>
1117 DATA 28,05,32,5E,4B,B7,C9,CD <1E69>
1118 DATA 70,44,37,C9,11,83,4D,CD <1E55>
1119 DATA D3,45,CD,C8,45,E6,5F,FE <1EA8>
1120 DATA 4E,C9,AF,32,65,4B,06,01 <1EE0>
1121 DATA C3,38,45,3A,02,A7,21,9F <1EFF>
1122 DATA A8,B7,C8,21,DF,A8,C9,11 <1EB1>
```

```
1123 DATA 73,4B,CD,D3,45,CD,41,49 <1EC2>
1124 DATA 20,08,11,AC,4D,CD,D3,45 <1E15>
1125 DATA 18,06,11,B1,4D,CD,D3,45 <1E73>
1126 DATA 11,B6,4D,CD,D3,45,3A,5D <1E1B>
1127 DATA 4B,CD,8C,45,CD,DC,45,3A <1E65>
1128 DATA 5D,4B,CD,60,45,CD,DC,45 <1E23>
1129 DATA 3A,5E,4B,CD,8C,45,CD,DC <1E69>
1130 DATA 45,3A,65,4B,3C,CD,8C,45 <1E72>
1131 DATA D5,CD,E3,4A,D1,CD,DC,45 <1E2A>
1132 DATA 11,00,5D,3A,65,4B,B7,28 <1ECE>
1133 DATA 03,11,00,5E,0E,00,21,04 <1E8E>
1134 DATA 01,E5,CD,75,BB,E1,06,10 <1EE6>
1135 DATA 3A,65,4B,CD,60,45,79,CD <1E69>
1136 DATA 60,45,D5,3E,20,CD,5A,BB <1E62>
1137 DATA 3E,20,CD,5A,BB,1A,CD,60 <1E44>
1138 DATA 45,13,10,F4,3E,20,CD,5A <1EE9>
1139 DATA BB,CD,5A,BB,D1,06,10,1A <1EBC>
1140 DATA E5,D5,C5,CD,5D,BB,C1,D1 <1E92>
1141 DATA E1,13,10,F3,3E,10,81,4F <1EF7>
1142 DATA 3E,14,2C,BD,20,BB,37,C9 <1EA6>
1143 DATA CD,84,BB,21,18,30,11,9C <1E60>
1144 DATA 4D,CD,D0,45,11,BE,4C,CD <1ED6>
1145 DATA D3,45,CD,C8,45,C3,81,BB <1EB1>
1146 DATA 21,18,01,CD,75,BB,3E,14 <1E5C>
1147 DATA C3,5A,BB,21,18,01,11,74 <1EDE>
1148 DATA 4D,CD,D0,45,3A,5E,4B,CD <1E41>
1149 DATA 8C,45,C3,DC,45,E5,E5,3E <1ECA>
1150 DATA 08,CD,5A,BB,3E,20,CD,5A <1E2D>
1151 DATA BB,E1,7E,CD,5A,BB,E1,C9 <1E6C>
1152 DATA 11,66,4D,21,17,01,CD,D0 <1E08>
1153 DATA 45,3A,5D,4B,18,D9,11,34 <1E9A>
1154 DATA 4D,21,16,01,CD,D0,45,3A <1E42>
1155 DATA 5C,4B,18,CB,CD,81,BB,CD <1ED2>
1156 DATA C8,45,FE,7F,28,25,FE,0D <1E12>
1157 DATA 28,33,FE,30,38,F1,FE,3A <1EB0>
1158 DATA 30,ED,77,CD,5A,BB,23,10 <1E57>
1159 DATA E6,CD,C8,45,FE,7F,28,0F <1ED7>
1160 DATA FE,0D,28,19,3E,07,CD,5A <1E10>
1161 DATA BB,18,EE,78,B9,28,D0,3E <1E09>
1162 DATA 08,CD,5A,BB,3E,10,CD,5A <1E4F>
1163 DATA BB,04,2B,18,C2,CD,84,BB <1EBE>
1164 DATA 78,B9,C9,21,59,4B,FE,01 <1EC0>
1165 DATA 28,0F,7E,D6,30,87,4F,87 <1E82>
1166 DATA 87,81,57,23,7E,D6,30,82 <1E06>
1167 DATA C9,CD,A5,44,16,00,18,F4 <1E83>
1168 DATA 3A,5E,4B,4F,0D,CD,C3,43 <1E32>
1169 DATA 7E,81,4F,3A,02,A7,5F,3A <1E45>
1170 DATA 5D,4B,57,21,00,5D,05,20 <1E11>
1171 DATA 09,F3,DF,5F,4B,FB,D2,2C <1E5D>
1172 DATA 43,C9,F3,DF,62,4B,18,F5 <1E03>
1173 DATA F5,C5,4F,0F,0F,0F,0F,E6 <1E20>
1174 DATA 0F,CD,75,45,79,E6,0F,CD <1EC1>
1175 DATA 75,45,C1,F1,C9,FE,0A,38 <1E45>
1176 DATA 02,C6,07,C6,30,47,3A,6B <1E9A>
1177 DATA 4B,B7,20,04,78,C3,5A,BB <1E18>
1178 DATA 78,C3,DB,4A,D5,C5,F5,FE <1E5B>
1179 DATA 0A,38,2C,4F,06,04,58,D6 <1E41>
1180 DATA 0A,38,02,10,FA,7B,90,47 <1E00>
1181 DATA 79,C6,06,10,FC,5F,CB,3F <1E9D>
1182 DATA CB,3F,CB,3F,CB,3F,C6,30 <1E11>
1183 DATA CD,7D,45,7B,E6,0F,C6,30 <1EBB>
1184 DATA CD,7D,45,F1,C1,D1,C9,F5 <1EC0>
1185 DATA 3E,20,CD,7D,45,F1,18,EE <1EF5>
1186 DATA CD,09,BB,38,FB,C3,06,BB <1E8B>
1187 DATA CD,75,BB,1A,FE,80,F2,DF <1E59>
1188 DATA 45,CD,5A,BB,13,18,F4,FE <1E44>
1189 DATA FF,C8,FE,90,30,F3,D6,80 <1EDE>
1190 DATA 47,3E,20,CD,5A,BB,10,FB <1E7B>
1191 DATA 18,EA,3A,65,4B,B7,28,07 <1E9E>
1192 DATA 97,32,65,4B,C3,CF,43,3C <1E3D>
1193 DATA 32,65,4B,3A,5E,4B,FE,01 <1E6D>
1194 DATA 28,0B,3D,32,5E,4B,CD,BA <1EB8>
1195 DATA 43,3E,01,18,E4,3E,09,32 <1EA9>
1196 DATA 5E,4B,3A,5D,4B,B7,28,06 <1E95>
1197 DATA 3D,32,5D,4B,18,E8,3E,27 <1ED7>
1198 DATA 18,F7,3A,65,4B,B7,20,07 <1E33>
1199 DATA 3C,32,65,4B,C3,CF,43,3A <1E43>
1200 DATA 5E,4B,FE,09,28,0A,3C,32 <1E0D>
1201 DATA 5E,4B,CD,BA,43,C3,CF,43 <1E3D>
1202 DATA 3E,01,32,5E,4B,3A,5D,4B <1EF8>
1203 DATA FE,27,28,0A,3C,32,5D,4B <1EE3>
1204 DATA CD,BA,43,C3,CF,43,97,18 <1EEC>
1205 DATA F4,21,15,01,CD,8B,44,CD <1EF5>
1206 DATA C6,44,CD,B8,44,CD,61,43 <1E0E>
1207 DATA 38,F8,CD,93,44,CD,88,43 <1E4F>
1208 DATA 38,F8,CD,AC,43,37,C8,CD <1E6B>
1209 DATA BA,43,D0,C3,CF,43,11,2E <1E41>
1210 DATA 4F,CD,D3,45,CD,81,BB,CD <1E79>
1211 DATA D4,48,D0,F5,21,17,01,11 <1E3C>
1212 DATA 48,4F,CD,D0,45,F1,D5,06 <1EC0>
1213 DATA 00,60,6F,29,11,12,00,19 <1E6D>
1214 DATA 1E,09,B7,04,ED,52,30,FB <1E28>
1215 DATA 05,B7,19,4D,CD,41,49,28 <1EB8>
1216 DATA 02,05,05,78,D1,CD,8C,45 <1EA9>
1217 DATA F5,CD,DC,45,F1,CD,60,45 <1EAB>
1218 DATA CD,DC,45,79,3C,CD,8C,45 <1E93>
1219 DATA CD,84,BB,11,BE,4C,CD,D3 <1EC5>
1220 DATA 45,CD,C8,45,C3,88,44,11 <1E38>
1221 DATA 72,4F,CD,D3,45,21,04,07 <1E98>
1222 DATA 22,66,4B,7C,32,6D,4B,CD <1E46>
1223 DATA 75,BB,CD,81,BB,3E,38,32 <1EAE>
1224 DATA 6C,4B,21,00,5D,3A,65,4B <1E26>
1225 DATA B7,28,03,21,00,5E,22,68 <1EF2>
1226 DATA 4B,CD,C8,45,FE,F0,28,4C <1EF5>
1227 DATA FE,F1,28,4D,FE,F2,28,4E <1E3B>
1228 DATA FE,F3,28,31,FE,E0,28,53 <1EA5>
1229 DATA FE,E1,28,61,FE,A3,CA,C5 <1EDC>
1230 DATA 48,FE,FC,CA,DA,47,FE,67 <1ED7>
1231 DATA 30,D7,FE,30,38,D3,CD,DA <1EDD>
1232 DATA 48,F5,2A,68,4B,77,3A,6C <1E34>
1233 DATA 4B,CD,6F,BB,F1,CD,5D,BB <1E45>
1234 DATA CD,96,48,18,06,CD,96,48 <1E9C>
1235 DATA D2,09,47,3A,6D,4B,CD,C0 <1E85>
1236 DATA 47,C3,09,47,CD,95,47,18 <1E98>
1237 DATA A8,CD,A6,47,18,A3,CD,67 <1E2B>
1238 DATA 48,38,9E,3A,6D,4B,CD,C0 <1E13>
1239 DATA 47,18,96,3A,6C,4B,CD,8E <1EB9>
1240 DATA 47,C3,F8,47,3A,6D,4B,CD <1EE5>
1241 DATA 8E,47,C3,09,47,CD,CA,47 <1E61>
1242 DATA F5,CD,60,45,18,B0,32,67 <1EE6>
1243 DATA 4B,CD,6F,BB,C9,3A,66,4B <1EA4>
1244 DATA FE,05,D8,3D,CD,B3,47,B7 <1E1B>
1245 DATA ED,52,22,68,4B,C9,3A,66 <1E89>
1246 DATA 4B,FE,13,D0,3C,CD,B3,47 <1E03>
```

Diskettenmonitor

```
1247 DATA 19,18,EF,32,66,4B,CD,72 <1EE0>
1248 DATA BB,2A,68,4B,11,10,00,C9 <1EAD>
1249 DATA 32,67,4B,2A,66,4B,CD,75 <1E3E>
1250 DATA BB,C9,2A,68,4B,7E,CB,7F <1EDB>
1251 DATA 28,04,CB,BF,18,02,CB,FF <1E77>
1252 DATA 77,C9,21,19,1E,11,19,50 <1EF1>
1253 DATA CD,D0,45,CD,F0,47,37,C0 <1EAC>
1254 DATA CD,84,BB,06,FF,C3,38,45 <1ED4>
1255 DATA CD,C8,45,E6,5F,FE,4A,C9 <1E0B>
1256 DATA CD,C8,45,FE,F0,28,47,FE <1EB9>
1257 DATA F1,28,48,FE,F2,28,49,FE <1EF1>
1258 DATA F3,28,2E,FE,E0,CA,7C,47 <1ECA>
1259 DATA FE,E1,28,4D,FE,A3,28,45 <1E02>
1260 DATA FE,FC,CA,DA,47,FE,81,D2 <1EE6>
1261 DATA F8,47,F5,2A,68,4B,77,CD <1EBB>
1262 DATA 5D,BB,3A,6D,4B,CD,6F,BB <1E21>
1263 DATA F1,CD,60,45,CD,96,48,18 <1E4B>
1264 DATA 05,CD,96,48,30,BA,3A,6C <1EB8>
1265 DATA 4B,CD,C0,47,18,B2,CD,95 <1E40>
1266 DATA 47,18,AD,CD,A6,47,18,A8 <1E25>
1267 DATA CD,67,48,38,A3,3A,6C,4B <1EC1>
1268 DATA CD,C0,47,18,9B,3E,E5,18 <1EFF>
1269 DATA C1,CD,CA,47,F5,18,C0,3A <1E50>
1270 DATA 6C,4B,FE,39,38,15,3D,32 <1E43>
1271 DATA 6C,4B,3A,6D,4B,D6,03,32 <1E95>
1272 DATA 6D,4B,2A,68,4B,2B,22,68 <1E90>
1273 DATA 4B,B7,C9,3A,66,4B,FE,05 <1E71>
1274 DATA D8,3D,32,66,4B,3E,47,32 <1EA0>
1275 DATA 6C,4B,3E,34,18,E1,3A,6C <1E87>
1276 DATA 4B,FE,47,30,15,3C,32,6C <1E3D>
1277 DATA 4B,3A,6D,4B,C6,03,32,6D <1E39>
1278 DATA 4B,2A,68,4B,23,22,68,4B <1E3B>
1279 DATA 37,C9,3A,66,4B,FE,13,D0 <1E03>
1280 DATA 3C,32,66,4B,3E,38,32,6C <1E82>
1281 DATA 4B,3E,07,18,E1,3E,45,CD <1E6D>
1282 DATA 5A,BB,3E,35,CD,5A,BB,3E <1E40>
1283 DATA E5,C3,39,47,CD,C8,45,FE <1E6D>
1284 DATA 0D,C8,06,02,21,59,4B,18 <1EB3>
1285 DATA 08,3E,07,CD,5A,BB,CD,C8 <1E31>
1286 DATA 45,FE,30,38,F4,FE,47,38 <1ED0>
1287 DATA 0C,FE,61,38,EC,FE,67,30 <1EDC>
1288 DATA E8,D6,20,18,08,FE,3A,38 <1ECD>
1289 DATA 06,FE,41,38,DC,D6,07,77 <1E6F>
1290 DATA FE,3A,38,02,C6,07,CD,5A <1E95>
1291 DATA BB,23,10,D2,2B,2B,7E,D6 <1E16>
1292 DATA 30,87,87,87,87,5F,23,7E <1E81>
1293 DATA D6,30,83,37,C9,21,15,01 <1E0E>
1294 DATA CD,8B,44,CD,C6,44,CD,B8 <1EBD>
1295 DATA 44,CD,61,43,38,F8,CD,93 <1EF2>
1296 DATA 44,CD,88,43,38,F8,C3,DA <1E5D>
1297 DATA 47,CD,C3,43,3E,41,BE,C9 <1E5A>
1298 DATA 21,51,4B,06,80,0E,00,11 <1E6A>
1299 DATA 67,49,CD,EF,BC,21,4B,4B <1E64>
1300 DATA 11,12,00,01,12,00,CD,E9 <1EA2>
1301 DATA BC,3E,07,32,58,4B,C9,F3 <1EC7>
1302 DATA F5,C5,D5,E5,3A,4A,4B,B7 <1EC7>
1303 DATA 28,09,CD,7E,BB,97,32,4A <1E4D>
1304 DATA 4B,18,08,CD,7B,BB,3E,FF <1EF4>
1305 DATA 32,4A,4B,E1,D1,C1,F1,FB <1E67>
1306 DATA C9,F3,21,4B,4B,CD,EC,BC <1E19>
1307 DATA AF,32,58,4B,FB,C9,11,13 <1E1E>
1308 DATA 4F,CD,D3,45,CD,F0,47,C0 <1E5F>
1309 DATA 3E,0D,4F,06,10,79,CD,2B <1E5A>
1310 DATA BD,38,03,10,F8,C9,3E,01 <1E94>
1311 DATA 32,6B,4B,CD,CC,4A,CD,41 <1E7F>
1312 DATA 49,11,B1,4D,20,03,11,AC <1E33>
1313 DATA 4D,CD,C2,4A,11,B6,4D,CD <1E9F>
1314 DATA C2,4A,3A,5D,4B,F5,CD,8C <1E45>
1315 DATA 45,CD,C9,4A,F1,CD,60,45 <1E6A>
1316 DATA CD,C9,4A,3A,5E,4B,CD,8C <1EB9>
1317 DATA 45,CD,56,4A,11,DE,4D,CD <1EA7>
1318 DATA C2,4A,CD,51,4A,11,00,5D <1EBB>
1319 DATA 0E,00,21,00,00,CD,0B,4A <1EBB>
1320 DATA 0E,00,21,00,01,CD,0B,4A <1EE5>
1321 DATA 11,2F,50,CD,C2,4A,AF,32 <1EB8>
1322 DATA 6B,4B,C9,06,10,7C,CD,60 <1E6A>
1323 DATA 45,79,CD,60,45,D5,3E,20 <1E6F>
1324 DATA CD,89,45,3E,20,CD,89,45 <1EAE>
1325 DATA 1A,CD,60,45,13,10,F4,3E <1E7E>
1326 DATA 20,CD,89,45,CD,89,45,D1 <1E34>
1327 DATA 06,10,1A,FE,20,38,04,FE <1EE3>
1328 DATA 81,38,02,3E,2E,CD,89,45 <1E63>
1329 DATA 13,10,EF,3E,10,81,4F,CD <1EC4>
1330 DATA 56,4A,3E,10,2C,BD,20,BB <1E1B>
1331 DATA C9,3E,0A,CD,DB,4A,3E,0D <1EAF>
1332 DATA CD,DB,4A,3E,0A,C3,DB,4A <1E4D>
1333 DATA CD,41,49,20,10,3A,5D,4B <1E6B>
1334 DATA FE,03,30,1E,3A,5E,4B,FE <1E2E>
1335 DATA 05,30,17,18,0D,3A,5D,4B <1EA6>
1336 DATA B7,20,0F,3A,5E,4B,FE,05 <1E0E>
1337 DATA 30,08,11,58,50,CD,D3,45 <1E3E>
1338 DATA 18,26,11,DB,52,CD,D3,45 <1ED0>
1339 DATA D5,21,00,5D,E5,21,00,00 <1E58>
1340 DATA 54,06,43,E3,7E,23,E3,5F <1E10>
1341 DATA 19,10,F8,D1,7C,CD,60,45 <1E44>
1342 DATA 7D,CD,60,45,D1,CD,DC,45 <1E8E>
1343 DATA 11,BE,4C,CD,D3,45,CD,C8 <1EF6>
1344 DATA 45,21,04,01,CD,8B,44,C3 <1E76>
1345 DATA 18,44,1A,FE,FF,C8,CD,DB <1E50>
1346 DATA 4A,13,18,F6,3E,1B,CD,DB <1E75>
1347 DATA 4A,3E,78,CD,DB,4A,3E,30 <1E71>
1348 DATA C3,DB,4A,CD,2E,BD,38,FB <1E0E>
1349 DATA C3,31,BD,3A,5D,4B,87,87 <1E6A>
1350 DATA 5F,6F,26,00,54,29,29,29 <1E23>
1351 DATA 19,3A,5E,4B,87,87,5F,19 <1E25>
1352 DATA 0E,00,1E,04,B7,ED,52,28 <1E97>
1353 DATA 07,ED,52,28,17,0C,18,F5 <1EE0>
1354 DATA CD,2B,4B,D0,CD,3B,4B,3A <1E85>
1355 DATA 65,4B,B7,CA,D3,45,CD,36 <1E0C>
1356 DATA 4B,C3,D3,45,CD,2B,4B,D0 <1E7D>
1357 DATA CD,3B,4B,CD,36,4B,CD,36 <1EC0>
1358 DATA 4B,18,E4,CD,41,49,C0,79 <1EB6>
1359 DATA D6,09,3F,D0,4F,C9,13,13 <1E44>
1360 DATA 13,13,C9,11,C6,54,CD,D3 <1E7A>
1361 DATA 45,79,CD,60,45,CD,DC,45 <1ECC>
1362 DATA 13,C9,00,00,00,00,00,00 <1ECF>
1363 DATA 00,00,00,00,00,00,00,00 <1E4D>
1364 DATA 00,00,00,00,00,00,00,00 <1E4F>
1365 DATA 00,00,00,00,00,00,00,00 <1E51>
1366 DATA 00,00,00,00,00,00,00,00 <1E53>
1367 DATA 81,84,85,1F,4F,14,13,1F <1EFE>
1368 DATA 01,01,FF,1D,16,16,1C,00 <1E09>
1369 DATA 16,16,FF,04,02,1C,01,16 <1ECA>
1370 DATA 16,1F,10,01,43,20,50,20 <1E92>
```

```
1371 DATA 43,83,44,20,49,20,53,20 (1EE1)
1372 DATA 43,20,4D,20,4F,20,4E,20 (1EBC)
1373 DATA 49,20,54,20,4F,20,52,83 (1E3C)
1374 DATA 3C,20,31,20,2E,20,31,20 (1ECA)
1375 DATA 3E,1F,0A,03,41,20,75,20 (1E35)
1376 DATA 74,20,6F,20,72,83,50,20 (1E4F)
1377 DATA 61,20,75,20,6C,83,4E,20 (1EE4)
1378 DATA 61,20,74,20,68,20,6D,20 (1E22)
1379 DATA 61,20,6E,20,6E,83,34,20 (1E4A)
1380 DATA 34,20,30,20,33,83,53,20 (1EDE)
1381 DATA 65,20,6E,20,64,20,65,20 (1E97)
1382 DATA 6E,1F,14,04,A4,83,62,20 (1EEC)
1383 DATA 79,83,53,20,63,20,68,20 (1EBF)
1384 DATA 6E,20,65,20,69,20,64,20 (1E48)
1385 DATA 65,20,72,83,41,20,6B,20 (1E47)
1386 DATA 74,20,69,20,76,FF,1F,20 (1EB3)
1387 DATA 06,FF,1F,1C,11,41,1F,1B (1EA0)
1388 DATA 13,22,44,49,4D,4F,4E,49 (1EAC)
1389 DATA 2E,42,49,4E,22,2C,42,2C (1E3B)
1390 DATA 26,34,30,30,30,2C,26,31 (1EC2)
1391 DATA 35,44,34,1F,14,16,4C,61 (1EDF)
1392 DATA 75,66,77,65,72,6B,20,41 (1E48)
1393 DATA 20,6F,64,65,72,20,42,82 (1EA2)
1394 DATA 6D,69,74,20,45,6E,74,65 (1EB7)
1395 DATA 72,20,61,6C,74,65,72,20 (1E19)
1396 DATA 57,65,72,74,FF,11,1F,23 (1E33)
1397 DATA 16,4C,61,75,66,77,65,72 (1E17)
1398 DATA 6B,20,FF,1F,14,18,42,69 (1E1F)
1399 DATA 74,74,65,20,64,69,65,20 (1E19)
1400 DATA 75,6E,74,65,72,73,75,63 (1EBD)
1401 DATA 68,65,6E,64,65,20,44,69 (1EA7)
1402 DATA 73,6B,65,74,74,65,20,65 (1E46)
1403 DATA 69,6E,6C,65,67,65,6E,2C (1E31)
1404 DATA 1F,18,19,64,61,6E,61,63 (1E31)
1405 DATA 68,20,69,72,67,65,6E,64 (1E8D)
1406 DATA 20,65,69,6E,65,20,54,61 (1E48)
1407 DATA 73,74,65,20,64,72,75,65 (1E14)
1408 DATA 63,6B,65,6E,2E,FF,1F,32 (1E15)
1409 DATA 19,4D,69,74,20,54,61,73 (1E9B)
1410 DATA 74,65,6E,64,72,75,63,6B (1EEF)
1411 DATA 20,77,65,69,74,65,72,2E (1EF1)
1412 DATA FF,1F,1A,0C,2A,20,2A,20 (1E5A)
1413 DATA 2A,20,46,61,6C,73,63,68 (1E3E)
1414 DATA 65,20,44,69,73,6B,65,74 (1E2F)
1415 DATA 74,65,20,2A,20,2A,20,2A (1E2E)
1416 DATA 07,FF,1F,0C,0C,2A,20,2A (1EB3)
1417 DATA 20,2A,20,42,65,66,65,68 (1EC3)
1418 DATA 6C,20,6E,69,63,68,74,20 (1E5E)
1419 DATA 67,65,66,75,6E,64,65,6E (1E75)
1420 DATA 20,2D,20,42,69,74,74,65 (1E09)
1421 DATA 20,6E,65,75,20,73,74,61 (1EA5)
1422 DATA 72,74,65,6E,20,2A,20,2A (1E8B)
1423 DATA 20,2A,07,FF,44,69,72,65 (1E53)
1424 DATA 63,74,6F,72,79,20,61,75 (1E7D)
1425 DATA 66,20,54,72,61,63,6B,20 (1EB0)
1426 DATA FF,83,53,65,6B,74,6F,72 (1E28)
1427 DATA 20,28,6F,68,6E,65,20,4F (1EB2)
1428 DATA 66,66,73,65,74,29,20,31 (1EEF)
1429 DATA 20,2D,20,34,21,FF,14,54 (1EC6)
1430 DATA 72,61,63,6B,85,28,FF,29 (1E57)
1431 DATA 84,3F,20,FF,14,53,65,6B (1E42)
1432 DATA 74,6F,72,84,28,FF,29,84 (1E17)
1433 DATA 3F,20,FF,1F,32,19,41,6C (1E7F)
1434 DATA 6C,65,73,20,72,69,63,68 (1E8C)
1435 DATA 74,69,67,82,28,20,2F,6E (1EFE)
1436 DATA 29,20,3F,FF,46,61,6C,73 (1E80)
1437 DATA 63,68,65,72,20,57,65,72 (1E62)
1438 DATA 74,21,07,FF,43,50,2F,4D (1EC3)
1439 DATA FF,44,41,54,41,FF,20,46 (1EAE)
1440 DATA 6F,72,6D,61,74,20,54,72 (1EC9)
1441 DATA 61,63,6B,20,FF,20,20,26 (1EC7)
1442 DATA FF,20,20,53,65,6B,74,6F (1E47)
1443 DATA 72,20,FF,20,20,53,65,69 (1E9F)
1444 DATA 74,65,20,FF,0D,0A,41,64 (1E2A)
1445 DATA 72,2E,20,20,20,30,20,20 (1EFD)
1446 DATA 31,20,20,32,20,20,33,20 (1E56)
1447 DATA 20,34,20,20,35,20,20,36 (1EB0)
1448 DATA 20,20,37,20,20,38,20,20 (1EEE)
1449 DATA 39,20,20,41,20,20,42,20 (1EB0)
1450 DATA 20,43,20,20,44,20,20,45 (1E54)
1451 DATA 20,20,46,20,20,30,31,32 (1E69)
1452 DATA 33,34,35,36,37,38,39,41 (1E22)
1453 DATA 42,43,44,45,46,FF,1F,01 (1E7D)
1454 DATA 14,14,0A,18,20,43,54,52 (1E47)
1455 DATA 4C,20,2B,20,F2,20,F3,20 (1E3E)
1456 DATA 18,20,6E,61,65,63,68,73 (1E68)
1457 DATA 74,65,20,53,65,6B,74,6F (1E0B)
1458 DATA 72,68,61,65,6C,66,74,65 (1E67)
1459 DATA 85,18,20,42,20,18,20,54 (1E19)
1460 DATA 72,61,63,6B,20,75,2E,20 (1EDB)
1461 DATA 53,65,6B,74,6F,72,20,65 (1E9A)
1462 DATA 72,72,65,63,68,6E,65,6E (1E76)
1463 DATA 0D,0A,18,20,4C,20,18,20 (1E5E)
1464 DATA 4C,65,73,65,6E,20,6D,69 (1E4A)
1465 DATA 74,20,53,65,6B,74,6F,72 (1E5B)
1466 DATA 65,6E,20,65,69,6E,67,61 (1E86)
1467 DATA 62,65,8A,18,20,53,20,18 (1EE4)
1468 DATA 20,53,63,68,72,65,69,62 (1E3F)
1469 DATA 65,6E,20,61,75,66,20,44 (1E87)
1470 DATA 69,73,6B,65,74,74,65,0D (1E97)
1471 DATA 0A,18,20,43,20,18,20,43 (1ED8)
1472 DATA 61,74,20,6F,64,65,72,20 (1E71)
1473 DATA 44,69,73,6B,65,74,74,65 (1EF5)
1474 DATA 6E,20,77,65,63,68,73,65 (1EEE)
1475 DATA 6C,8A,18,20,49,20,18,20 (1EBF)
1476 DATA 49,6E,66,6F,72,6D,61,74 (1E66)
1477 DATA 69,6F,6E,0D,0A,18,20,41 (1ED2)
1478 DATA 20,18,20,41,65,6E,64,65 (1E06)
1479 DATA 72,6E,89,18,20,50,20,18 (1EFD)
1480 DATA 20,44,72,75,63,6B,65,6E (1E2A)
1481 DATA 89,18,20,43,54,52,4C,20 (1E80)
1482 DATA 26,20,45,20,18,20,45,6E (1EE7)
1483 DATA 64,65,FF,1F,01,14,14,1F (1ECD)
1484 DATA 1C,18,18,20,44,72,75,63 (1E55)
1485 DATA 6B,65,6E,82,28,6A,2F,20 (1E23)
1486 DATA 29,20,3F,20,18,FF,1F,01 (1E99)
1487 DATA 14,14,0A,42,6C,6F,63,6B (1ECD)
1488 DATA 6E,72,2E,20,65,69,6E,67 (1E58)
1489 DATA 65,62,65,6E,20,3F,20,FF (1EB2)
1490 DATA 54,72,61,63,6B,6E,72,2E (1EF6)
1491 DATA 20,3D,20,FF,83,26,FF,86 (1E5A)
1492 DATA 53,65,6B,74,6F,72,6E,72 (1E6E)
1493 DATA 2E,20,3D,20,FF,49,6E,66 (1ED5)
1494 DATA 6F,72,6D,61,74,69,6F,6E (1EA9)
```

Diskettenmonitor

```
1495 DATA 20,FF,1F,01,14,14,1F,1C <1E58>
1496 DATA 15,41,65,6E,64,65,72,75 <1E58>
1497 DATA 6E,67,73,20,2D,20,4D,6F <1EC7>
1498 DATA 64,75,73,1F,06,16,F2,0B <1E38>
1499 DATA F0,08,0A,0A,F1,0B,F3,83 <1EF6>
1500 DATA 45,69,6E,67,61,62,65,20 <1E02>
1501 DATA 61,75,66,20,43,75,72,73 <1E12>
1502 DATA 6F,72,20,20,2D,20,20,43 <1E33>
1503 DATA 54,52,4C,20,2B,20,54,41 <1E31>
1504 DATA 42,20,20,42,69,74,20,37 <1ECB>
1505 DATA 20,73,65,74,7A,65,6E,20 <1E31>
1506 DATA 6F,64,65,72,20,6C,6F,65 <1EF5>
1507 DATA 73,63,68,65,6E,1F,16,17 <1EBF>
1508 DATA 6D,69,74,20,A3,20,4C,6F <1E61>
1509 DATA 65,73,63,68,62,79,74,65 <1E8A>
1510 DATA 20,26,45,35,20,2D,20,43 <1E76>
1511 DATA 4F,50,59,20,77,65,63,68 <1E09>
1512 DATA 73,65,6C,6E,1F,1B,18,45 <1ED6>
1513 DATA 69,6E,67,61,62,65,61,62 <1EBF>
1514 DATA 73,63,68,6C,75,73,73,20 <1EE8>
1515 DATA 6D,69,74,20,20,45,53,43 <1E55>
1516 DATA FF,18,20,53,70,65,69,63 <1E22>
1517 DATA 68,65,72,6E,83,28,6A,2F <1E24>
1518 DATA 20,29,20,3F,20,18,FF,0A <1E01>
1519 DATA 20,20,20,20,20,20,2A,20 <1EAD>
1520 DATA 2A,20,2A,20,20,41,75,73 <1E40>
1521 DATA 64,72,75,63,6B,20,62,65 <1E9C>
1522 DATA 65,6E,64,65,74,2E,20,20 <1ECF>
1523 DATA 2A,20,2A,20,2A,0D,0A,FF <1E37>
1524 DATA 1F,01,06,14,0A,45,69,6E <1EAC>
1525 DATA 20,44,69,72,65,63,74,6F <1E86>
1526 DATA 72,79,20,69,73,74,20,26 <1E84>
1527 DATA 32,30,20,42,79,74,65,73 <1E87>
1528 DATA 20,6C,61,6E,67,2E,0D,0A <1E45>
1529 DATA 42,79,74,65,20,30,30,20 <1EE5>
1530 DATA 3A,86,55,73,65,72,6E,75 <1E5D>
1531 DATA 6D,6D,65,72,20,30,2D,31 <1EED>
1532 DATA 35,20,28,20,45,35,20,67 <1E71>
1533 DATA 65,6C,6F,65,73,63,68,74 <1E6A>
1534 DATA 20,6F,64,65,72,20,6E,69 <1E9A>
1535 DATA 63,68,74,20,62,65,6C,65 <1E65>
1536 DATA 67,74,20,29,2E,0D,0A,42 <1E23>
1537 DATA 79,74,65,20,30,31,20,2D <1E87>
1538 DATA 20,30,38,20,3A,20,46,69 <1E0E>
1539 DATA 6C,65,6E,61,6D,65,20,69 <1E7C>
1540 DATA 6E,20,47,72,6F,73,73,62 <1EFC>
1541 DATA 75,63,68,73,74,61,62,65 <1E22>
1542 DATA 6E,2E,0D,0A,42,79,74,65 <1E90>
1543 DATA 20,30,39,20,2D,20,30,42 <1EC8>
1544 DATA 20,3A,20,46,69,6C,65,6E <1EC7>
1545 DATA 61,6D,65,20,2D,20,45,78 <1E4A>
1546 DATA 74,65,6E,73,69,6F,6E,20 <1E6E>
1547 DATA 69,6E,20,47,72,6F,73,73 <1E03>
1548 DATA 62,75,63,68,73,74,61,62 <1EDB>
1549 DATA 65,6E,2E,0D,0A,42,79,74 <1EC7>
1550 DATA 65,20,30,39,20,3A,86,3E <1E21>
1551 DATA 20,26,38,30,2C,20,42,69 <1EEA>
1552 DATA 74,20,37,3D,31,20,44,61 <1ED3>
1553 DATA 74,65,69,20,69,73,74,20 <1E3B>
1554 DATA 73,63,68,72,65,69,62,67 <1E0A>
1555 DATA 65,73,63,68,75,65,74,7A <1E3A>
1556 DATA 74,2E,0D,0A,42,79,74,65 <1E31>
1557 DATA 20,30,41,20,3A,86,3E,20 <1E47>
1558 DATA 26,38,30,2C,20,42,69,74 <1E24>
1559 DATA 20,37,3D,31,20,44,61,74 <1E4A>
1560 DATA 65,69,20,77,69,72,64,20 <1E47>
1561 DATA 62,65,69,20,44,49,52,20 <1ECD>
1562 DATA 6E,69,63,68,74,20,61,6E <1EF6>
1563 DATA 67,65,7A,65,69,67,74,2E <1E69>
1564 DATA 0D,0A,42,79,74,65,20,30 <1EE0>
1565 DATA 43,20,3A,86,45,78,74,65 <1E2C>
1566 DATA 6E,64,6E,72,2C,20,77,69 <1E7F>
1567 DATA 72,64,20,75,6D,20,31,20 <1E06>
1568 DATA 65,72,68,6F,65,74,20,73 <1E99>
1569 DATA 6F,62,61,6C,64,20,31,36 <1E37>
1570 DATA 20,4B,20,75,65,62,65,72 <1EB0>
1571 DATA 73,63,68,72,69,74,74,65 <1EF2>
1572 DATA 6E,20,77,65,72,64,65,6E <1E07>
1573 DATA 2C,0D,0A,8F,64,61,62,65 <1EA3>
1574 DATA 69,20,65,72,66,6F,6C,67 <1EBC>
1575 DATA 74,20,6A,65,77,65,69,6C <1EDB>
1576 DATA 73,20,65,69,6E,65,20,6E <1E45>
1577 DATA 65,75,65,20,44,69,72,65 <1EC1>
1578 DATA 63,74,6F,72,79,65,69,6E <1EF9>
1579 DATA 74,72,61,67,75,6E,67,2E <1E8E>
1580 DATA 0D,0A,42,79,74,65,20,30 <1E00>
1581 DATA 44,20,2D,20,30,45,20,3A <1E7A>
1582 DATA 20,6E,69,63,68,74,20,62 <1EFC>
1583 DATA 65,6E,75,74,7A,74,2E,0D <1E1A>
1584 DATA 0A,42,79,74,65,20,30,46 <1EC1>
1585 DATA 20,3A,86,41,6E,7A,61,68 <1EE1>
1586 DATA 6C,20,64,65,72,20,52,65 <1E59>
1587 DATA 63,6F,72,64,73,20,28,20 <1E72>
1588 DATA 31,20,52,65,63,6F,72,64 <1E2C>
1589 DATA 20,3D,20,31,32,38,20,42 <1E71>
1590 DATA 79,74,65,73,20,29,2E,0D <1E3D>
1591 DATA 0A,42,79,74,65,20,31,30 <1ECF>
1592 DATA 20,2D,20,31,46,20,3A,20 <1EAA>
1593 DATA 42,6C,6F,63,6B,6E,75,6D <1E92>
1594 DATA 6D,65,72,20,28,20,31,20 <1E7B>
1595 DATA 42,6C,6F,63,6B,20,3D,20 <1E50>
1596 DATA 32,20,53,65,6B,74,6F,72 <1E66>
1597 DATA 65,6E,20,3D,20,38,20,52 <1EC8>
1598 DATA 65,63,6F,72,64,73,20,29 <1EA4>
1599 DATA 2E,0D,0A,42,79,74,65,20 <1E45>
1600 DATA 31,30,20,3A,86,42,6C,6F <1EDB>
1601 DATA 63,6B,61,64,72,65,73,73 <1E7E>
1602 DATA 65,20,64,65,73,20,46,69 <1EFF>
1603 DATA 6C,65,68,65,61,64,65,72 <1E40>
1604 DATA 2E,02,FF,1F,01,09,14,0A <1EE6>
1605 DATA 44,61,74,65,6E,20,64,65 <1E8E>
1606 DATA 73,20,46,69,6C,65,68,65 <1E3D>
1607 DATA 61,64,65,72,73,2E,0D,0A <1E2F>
1608 DATA 42,79,74,65,20,30,30,20 <1E85>
1609 DATA 2D,20,30,42,20,3A,20,57 <1E25>
1610 DATA 69,65,20,62,65,69,20,44 <1E5A>
1611 DATA 69,72,65,63,74,6F,72,79 <1E23>
1612 DATA 65,69,6E,74,72,61,67,75 <1E35>
1613 DATA 6E,67,2E,0D,0A,42,79,74 <1EEF>
1614 DATA 65,20,31,32,86,3A,20,30 <1EE9>
1615 DATA 30,20,3D,20,42,41,53,49 <1EAA>
1616 DATA 43,20,50,72,6F,67,72,61 <1ED8>
1617 DATA 6D,6D,1F,10,0D,30,31,20 <1EFA>
1618 DATA 3D,20,67,65,73,63,68,75 <1E0C>
```

```
1619 DATA 65,74,7A,74,65,73,20,42 <1E6A>
1620 DATA 41,53,49,43,20,50,72,6F <1ED2>
1621 DATA 67,72,61,6D,6D,1F,10,0E <1E9B>
1622 DATA 30,32,20,3D,20,42,69,6E <1EAF>
1623 DATA 61,65,72,20,50,72,6F,67 <1ECB>
1624 DATA 72,61,6D,6D,0D,0A,42,79 <1EDE>
1625 DATA 74,65,20,31,36,20,2B,20 <1E25>
1626 DATA 31,35,20,3A,20,4C,61,64 <1E00>
1627 DATA 65,61,64,72,65,73,73,65 <1E85>
1628 DATA 0D,0A,42,79,74,65,20,31 <1E65>
1629 DATA 39,20,2B,20,31,38,20,3A <1E24>
1630 DATA 20,4C,61,65,6E,67,65,0D <1E3B>
1631 DATA 0A,42,79,74,65,20,30,42 <1E0E>
1632 DATA 20,2B,20,30,41,20,3A,20 <1E9E>
1633 DATA 53,74,61,72,74,61,64,72 <1E82>
1634 DATA 65,73,73,65,20,77,65,6E <1E77>
1635 DATA 6E,20,42,79,74,65,20,31 <1E00>
1636 DATA 32,20,3D,20,30,32,0D,0A <1ECB>
1637 DATA 42,79,74,65,20,34,31,20 <1ED3>
1638 DATA 2B,20,34,30,20,3A,20,4C <1EA5>
1639 DATA 61,65,6E,67,65,0D,0A,42 <1E3F>
1640 DATA 79,74,65,20,34,34,20,2B <1EE8>
1641 DATA 20,34,33,20,3A,20,50,72 <1E4C>
1642 DATA 75,65,66,73,75,6D,6D,65 <1EB2>
1643 DATA 20,64,65,72,20,42,79,74 <1ED0>
1644 DATA 65,73,20,30,30,20,2D,20 <1E19>
1645 DATA 34,32,0D,0A,42,79,74,65 <1E06>
1646 DATA 20,34,35,20,2D,20,37,46 <1E9A>
1647 DATA 20,3A,20,55,6E,77,69,63 <1E98>
1648 DATA 68,74,69,67,2E,0D,0A,0A <1E48>
1649 DATA 41,6B,74,75,65,6C,6C,65 <1E8F>
1650 DATA 20,50,72,75,65,66,73,75 <1EAF>
1651 DATA 6D,6D,65,20,64,65,72,20 <1E61>
1652 DATA 42,79,74,65,73,20,30,30 <1EAA>
1653 DATA 20,2D,20,34,32,20,3D,20 <1E3A>
1654 DATA 18,20,FF,20,18,0D,0A,77 <1E49>
1655 DATA 65,6E,6E,20,67,6C,65,69 <1ED2>
1656 DATA 63,68,20,64,61,6E,6E,20 <1E8E>
1657 DATA 42,41,53,49,43,2D,20,6F <1EB9>
1658 DATA 64,65,72,20,42,49,4E,41 <1E66>
1659 DATA 45,52,20,50,72,6F,67,72 <1EE9>
1660 DATA 61,6D,6D,0D,0A,73,6F,6E <1E37>
1661 DATA 73,74,20,41,53,43,49,49 <1EDE>
1662 DATA 2D,2C,20,43,50,2F,4D,20 <1E4E>
1663 DATA 50,72,6F,67,72,61,6D,6D <1EF9>
1664 DATA 20,6F,64,65,72,20,44,41 <1EDD>
1665 DATA 54,45,49,2E,02,FF,82,42 <1E3B>
1666 DATA 4C,4F,43,4B,20,26,FF,82 <1E4D>
1667 DATA 52,65,63,6F,72,64,74,65 <1EB4>
1668 DATA 69,6C,65,20,FF,31,2B,32 <1E2F>
1669 DATA FF,33,2B,34,FF,35,2B,36 <1E63>
1670 DATA FF,37,2B,38,FF,1F,0F,0C <1EB5>
1671 DATA 2A,20,2A,20,2A,20,44,69 <1E1E>
1672 DATA 73,6B,65,74,74,65,20,66 <1E67>
1673 DATA 65,68,6C,74,20,2F,20,4C <1ED4>
1674 DATA 61,75,66,77,65,72,6B,20 <1EEB>
1675 DATA 6E,69,63,68,74,20,62,65 <1EB9>
1676 DATA 72,65,69,74,20,2A,20,2A <1EBC>
1677 DATA 20,2A,07,FF,1F,13,0C,2A <1E0C>
1678 DATA 20,2A,20,2A,20,44,69,73 <1E5C>
1679 DATA 6B,65,74,74,65,20,69,73 <1E4D>
1680 DATA 74,20,73,63,68,72,65,69 <1E0B>
1681 DATA 62,67,65,73,63,68,75,65 <1E7A>
1682 DATA 74,7A,74,20,2A,20,2A,20 <1E96>
1683 DATA 2A,07,FF,1F,13,0C,2A,20 <1EF2>
1684 DATA 2A,20,2A,20,44,69,73,6B <1EBE>
1685 DATA 65,74,74,65,20,69,73,74 <1EEB>
1686 DATA 20,6E,69,63,68,74,20,66 <1EDD>
1687 DATA 6F,72,6D,61,74,69,65,72 <1ECB>
1688 DATA 74,20,2A,20,2A,20,2A,07 <1EA6>
1689 DATA FF,FF,38,30,40,29,44,26 <1E1A>
1690 DATA 46,23,49,21,4A,1F,4C,1E <1EA7>
1691 DATA 4D,1D,4E,1C,4F,1B,4F,1B <1EF5>
1692 DATA 50,1A,50,1A,37,32,3A,2F <1EE0>
1693 DATA 3C,2D,3D,2C,3E,2B,F8,F4 <1E65>
1694 DATA FB,F1,FD,EF,FE,EE,7A,76 <1E73>
1695 DATA 7D,73,7F,71,80,70,8E,8C <1E7A>
1696 DATA 91,89,92,88,DE,DC,E1,D9 <1EDF>
1697 DATA E2,D8,37,31,3E,2B,41,28 <1E1A>
1698 DATA 43,26,45,25,46,24,47,23 <1E1C>
1699 DATA 47,23,00,20,00          <155B>
```

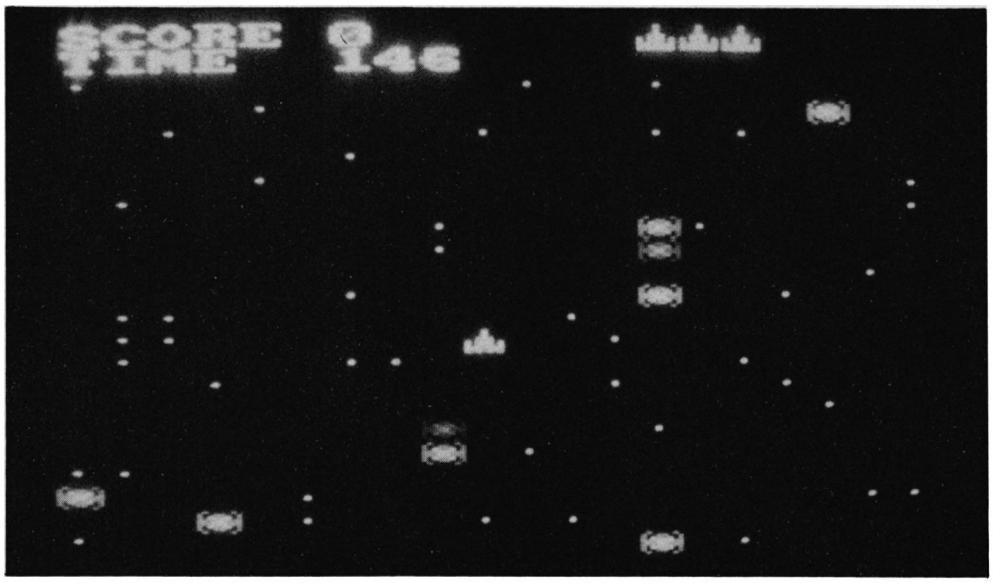
Zoff in der Galaxis

Der Spieler muß in der Rolle eines Weltraumpiloten und als Letzter seiner Art gegen die gigantischen Flottenverbände fremder Zivilisationen antreten. Einzige Waffe in diesem ungleichen Kampf sind die Galaxisminen, mit denen er die umhersausenden Ufos einkreisen kann.

Auf dem Bildschirm wird der Sternenhimmel dargestellt. In der Mitte erscheint das Raumschiff des Spielers, drumherum sind die feindlichen Ufos verstreut.

Mit dem Joystick muß der Spieler sein Raumschiff nun so steuern, daß er mit den Minen, die automatisch überall dort ausgelegt werden, wo sich das Schiff gerade befindet, die Ufos einkreist. Diese explodieren nämlich bei Berührung mit einer Mine. Sind alle Ufos vernichtet,

beginnt der nächste Level, in dem noch mehr Feinde auftauchen.
Bei all diesen Aktionen muß der Spieler darauf achten, daß er nicht mit seinen eigenen Minen kollidiert, nicht mit einem Ufo zusammenstößt und auch im Zeitlimit bleibt.
Das Spiel läuft auf allen CPC-Typen. *JE*□




```
10 '******************************* <24C3>
20 '*    ZOFF IN DER GALAXIS     * <24E0>
30 '*            VON             * <2407>
40 '*        MARTIN KEHR         * <2463>
50 '*            FUER            * <247E>
60 '*    SCHNEIDER CPC-WELT      * <245A>
70 '*     CPC 464/664/6128    JE* <24D1>
80 '******************************* <244E>
90 DEFINT a-z                     <0AC1>
100 RANDOMIZE TIME                <0804>
110 KEY 1,"mode 0"+CHR$(13):KEY 0,
"mode 2:pen 1:ink 1,26"+CHR$(13):K
EY 2,"edit "                      <44F2>
120 SYMBOL AFTER 236              <095B>
130 SYMBOL 236,32,96,64,16,16,64,9
6,32                              <20E1>
140 SYMBOL 237,36,0,&X11000011,255
,255,&X11000011,0,36              <205F>
150 SYMBOL 238,0,24,24,&X10011001,
&X10011001,24,24,0                <2094>
160 SYMBOL 239,60,126,255,24,24,25
5,126,60                          <20B6>
170 SYMBOL 240,16,16,16,56,&X10111
010,&X10111010,254,254            <2292>
180 SYMBOL 241,240,192,248,255,248
,192,240,0                        <1F70>
190 SYMBOL 242,127,127,&X1011101,&
X1011101,&X11100,8,8,8            <20C0>
200 SYMBOL 243,0,15,3,31,255,31,3,
15                                <1D4B>
210 SYMBOL 244,0,0,0,0,0,16,16,&X1
0010100                           <1CD1>
220 SYMBOL 245,0,128,0,224,0,128,0
,0                                <1B6A>
230 SYMBOL 246,&X101010,8,8,0,0,0,
0,0                               <1A6C>
240 SYMBOL 247,0,0,1,0,7,0,1,0    <189F>
250 SYMBOL 248,0,0,0,16,0,0,0,0   <19B3>
260 SYMBOL 249,0,60,66,66,66,66,60
,0                                <1ED9>
270 SYMBOL 250,0,0,0,24,24,0,0,0  <1AC8>
280 SYMBOL 251,66,&X10011001,&X101
11101,255,255,&X10111101,&X1001100
1,66                              <2479>
290 SYMBOL 252,0,0,36,24,24,36,0,0
                                  <1C27>
300 SYMBOL 253,131,&X1000110,60,62
,124,252,126,&X10011001           <22A4>
310 SYMBOL 254,0,0,36,&X1011000,24
,36,4,16                          <1E57>
320 SYMBOL 255,8,&X1000010,&X10010
0,1,136,34,64,8                   <1F0D>
330 DEF FNl=level MOD 4           <165B>
340 DIM farben(4,2):DIM c(4)      <1D13>
350 FOR n=1 TO 4:READ a,d,b:farben
(n,1)=a:farben(n,2)=d:c(n)=b:NEXT <5ECE>
360 DATA 7,17,251,10,21,237,10,21,
237,7,17,251                      <2CA0>
370 BORDER 0:PAPER 0:PEN 1        <0DC9>
380 feind=-2                      <105D>
390 score=0:schiffe=2:level=0     <270F>
400 REM initialisierung           <16C3>
410 MODE 0:IF feind=-2 THEN GOSUB
1910:CLS                          <1BF9>
420 INK 0,0:INK 1,26:INK 2,6:INK 3
,14                               <1AF3>
430 INK 4,9:INK 5,18:INK 6,farben(
FNl+1,1):INK 7,farben(FNl+1,2)    <3F6B>
440 INK 8,24,0:INK 9,15,0:INK 10,1
6                                 <1B05>
450 INK 11,12:INK 12,5            <112C>
460 feind=4+level*3+3:DIM a$(23):D
IM r(feind):DIM s(feind)          <4652>
470 FOR n=1 TO 23:a$(n)=SPACE$(20)
:NEXT                             <23A5>
480 FOR n=1 TO 50                 <0FD0>
490 x=1+INT(RND(1)*20):y=1+INT(RND
(1)*23)                           <2CBD>
```

```
500 MID$(a$(y),x,1)=CHR$(248):NEXT
 n                                  (2654)
510 '                               (0708)
520 'bildschirmaufbau               (1704)
530 '                               (0731)
540 WINDOW#0,1,20,3,25              (12CB)
550 WINDOW#1,1,20,1,2               (11B8)
560 ORIGIN 0,368                    (0BC8)
570 DEF FNfarb(x,y)=TEST(x*32-16,-
y*16+8)                             (33C9)
580 CLS:CLS#1                       (0A4B)
590 feind=4+3*level+3               (1C31)
600 INK 1,0:INK 3,0:PRINT#1,"SCORE
":PRINT#1,"TIME"                    (2501)
610 GOSUB 1110                      (0957)
620 PEN 1                           (07F4)
630 REM sterne                      (0DF5)
640 FOR n=1 TO 23:PRINT a$(n);:NEX
T                                   (1ED5)
650 INK 1,26:INK 3,14               (1085)
660 REM feindliche raumschiffe      (1D80)
670 FOR n=1 TO feind                (15FF)
680 r(n)=1+INT(RND(1)*20):s(n)=1+I
NT(RND(1)*23)                       (3815)
690 IF FNfarb(r(n),s(n))>1 THEN 68
0                                   (2BD1)
700 LOCATE r(n),s(n):PEN 6:PRINT C
HR$(c(FNl+1));                      (325A)
710 LOCATE r(n),s(n):PEN 7:PRINT C
HR$(22)+CHR$(1);CHR$(1+c(FNl+1));C
HR$(22)+CHR$(0);:PEN 1:NEXT n       (5775)
720 x=10:y=12:a=240                 (1C46)
730 gz=0:gx=1:gy=1                  (1C4D)
740 zaehl=1:t1=180-30*(level>2)     (28A6)
750 REM                             (06F4)
760 REM hauptschleife               (1477)
770 REM                             (061E)
780 x1=x:y1=y                       (1ADA)
790 LOCATE#1,6,2:PRINT#1,t1:t1=t1-
1                                   (24C4)
800 IF t1=-1 THEN 1010              (1288)
810 IF FNl+1>2 THEN GOSUB 1810      (148E)
820 GOSUB 1200:zaehl=zaehl+1:IF za
ehl>feind THEN zaehl=1              (3B98)
830 f=FNfarb(x,y):IF f>3 THEN 1010
                                    (2954)
840 j=JOY(0)                        (0F56)
850 IF j=8 THEN x=x+1:a=241         (20A6)
860 IF j=4 THEN x=x-1:a=243         (20C9)
870 IF j=1 THEN y=y-1:a=240         (2021)
880 IF j=2 THEN y=y+1:a=242         (2078)
890 IF x>20 OR x<1 OR y<1 OR y>23
THEN x=x1:y=y1                      (399F)
900 f=FNfarb(x,y):IF f>3 THEN 1010
                                    (29E0)
910 LOCATE x,y:PEN 3:PRINT CHR$(a)
;                                   (1DFE)
920 LOCATE x,y:PEN 2:PRINT CHR$(22
)+CHR$(1);CHR$(a+4);CHR$(22)+CHR$(
0);:PEN 1                           (3CF2)
930 IF x=x1 AND y=y1 THEN 780       (1F8C)
940 LOCATE x1,y1:PEN 4:PRINT CHR$(
249);                               (1DC2)
950 LOCATE x1,y1:PEN 5:PRINT CHR$(
22)+CHR$(1);CHR$(250);CHR$(22)+CHR
$(0);:PEN 1                         (3AE1)
960 MID$(a$(y1),x1,1)=CHR$(249)     (229C)
970 GOTO 780                        (09FA)
980 REM                             (06C1)
990 REM ein schiff verloren         (1A90)
1000 REM                            (06E9)
1010 schiffe=schiffe-1              (1C5C)
1020 LOCATE x1,y1:PEN 8:PRINT CHR$
(253);                              (1DC1)
1030 LOCATE x1,y1:PEN 9:PRINT CHR$
(22)+CHR$(1);CHR$(254);CHR$(22)+CH
R$(0);:PEN 1                        (3ABF)
1040 SOUND 1,0,20,15,,,5            (1305)
1050 FOR n=1 TO 50:SOUND 1,0,3,15-
(n/4),,,20:SOUND 1,0,3,15-(n/6),,,
28:NEXT n                           (453D)
1060 IF schiffe=0 THEN 1470         (165B)
1070 feind=4+3*level+3:GOTO 470     (2183)
1080 REM                            (068B)
1090 REM score                      (0CAF)
1100 REM                            (06B3)
1110 LOCATE#1,6,1:PRINT#1,STR$(sco
re)                                 (1DAE)
1120 IF(score MOD 100)=0 THEN schi
ffe=schiffe+1                       (2D7C)
1130 IF schiffe>7 THEN schiffe=7    (1FBE)
1140 LOCATE#1,14,1:PEN#1,3:PRINT#1
,STRING$(schiffe,CHR$(240))         (2D38)
1150 PEN#1,1                        (0AE0)
1160 RETURN                         (063A)
1170 REM                            (063E)
1180 REM feindliche schiffe         (19E8)
1190 REM                            (0666)
1200 IF r(zaehl)=0 THEN RETURN      (18B0)
1210 LOCATE r(zaehl),s(zaehl):PRIN
T MID$(a$(s(zaehl)),r(zaehl),1);    (4E08)
1220 IF c(FNl+1)=237 THEN r(zaehl)
=r(zaehl)-SGN(r(zaehl)-x):s(zaehl)
=s(zaehl)-SGN(s(zaehl)-y):GOTO 129
0                                   (87F6)
1230 t=RND(1)*4                     (1107)
1240 r1=r(zaehl):s1=s(zaehl)        (2EFF)
1250 r(zaehl)=r(zaehl)-(t=0)+(t=1)
                                    (3413)
1260 t=1+RND(1)*3                   (13C0)
1270 s(zaehl)=s(zaehl)-(t=2)+(t=3)
                                    (34DC)
1280 IF r(zaehl)>20 OR r(zaehl)<1
OR s(zaehl)<1 OR s(zaehl)>23 THEN
r(zaehl)=r1:s(zaehl)=s1             (75EE)
1290 IF FNfarb(r(zaehl),s(zaehl))=
5 THEN 1360                         (331B)
1300 LOCATE r(zaehl),s(zaehl):PEN
6:PRINT CHR$(c(FNl+1));             (3A5E)
1310 LOCATE r(zaehl),s(zaehl):PEN
7::PRINT CHR$(22)+CHR$(1);CHR$(1+c
(FNl+1));CHR$(22)+CHR$(0);:PEN 1    (5AA3)
1320 RETURN                         (067C)
1330 REM                            (0680)
```

Zoff in der Galaxis

```
1340 REM feind zerschellt                    (1712)
1350 REM                                     (06A8)
1360 LOCATE r(zaehl),s(zaehl):PRIN
T CHR$(255);                                 (2C23)
1370 FOR n=1 TO 10:SOUND 1,0,10,15
-n,,,7:NEXT                                  (252B)
1380 LOCATE r(zaehl),s(zaehl):PRIN
T" ";:MID$(a$(s(zaehl)),r(zaehl),1
)=" "                                        (5656)
1390 score=score+5:GOSUB 1110                (1D01)
1400 r(zaehl)=r(feind):s(zaehl)=s(
feind):feind=feind-1                         (543D)
1410 IF feind=0 THEN 1520                    (14E3)
1420 GOSUB 1110                              (09AE)
1430 RETURN                                  (0657)
1440 REM                                     (065B)
1450 REM game over                           (1087)
1460 REM                                     (0683)
1470 LOCATE 6,10:PRINT"GAME OVER"            (17C8)
1480 IF JOY(0)=16 THEN RUN ELSE 14
80                                           (1557)
1490 REM                                     (06BF)
1500 REM zone geschafft                      (1553)
1510 REM                                     (06E7)
1520 FOR n=1 TO 3000:NEXT                    (12BD)
1530 FOR n=0 TO t1                           (127D)
1540 score=score+1                           (1879)
1550 GOSUB 1110                              (09B2)
1560 LOCATE#1,6,2:PRINT#1,t1-n;" "
;                                            (20CA)
1570 NEXT n                                  (0ACF)
1580 SOUND 1,0,30,15,,,3                     (1323)
1590 FOR m=1 TO 10                           (0F5F)
1600 FOR n=1 TO 10:INK 4,15+n:SOUN
D 1,0,2,15,,,3+n:NEXT                        (2FA4)
1610 NEXT m                                  (0A1A)
1620 FOR n=15 TO 1 STEP-0.5:SOUND
1,0,2,n,,,13:NEXT                            (2AD2)
1630 INK 4,9                                 (09F5)
1640 ERASE a$,r,s:level=level+1              (28BE)
1650 CLS:CLS#1:ORIGIN 0,0                    (0FA2)
1660 FOR n=1 TO 14                           (0F7A)
1670 INK n,0                                 (0CB2)
1680 MOVE 224-n*10,208+n*10:DRAWR
192+20*n,0,n:DRAWR 0,-16-20*n,n:DR
AWR-192-20*n,0,n:DRAWR 0,16+20*n,n
                                             (69DD)
1690 NEXT                                    (06FD)
1700 INK 15,26:PEN 15:LOCATE 8,11:
PRINT"ZONE"STR$(level+1);                    (2C11)
1710 FOR m=1 TO 5:INK 14,26:INK 1,
0:FOR n=13 TO 1 STEP-1                       (281B)
1720 INK n,26:INK n+1,0                      (179E)
1730 FOR q=1 TO 100:NEXT q                   (1570)
1740 SOUND 0,n+50*m,4,15:SOUND 1,n
+45*m,4,15:SOUND 2,n+55*m,4,15               (439E)
1750 NEXT n:NEXT m                           (10D4)
1760 CLS                                     (06F1)
1770 PEN 1:GOTO 420                          (0C7C)
1780 REM                                     (0604)
1790 REM kamikaze                            (0F42)
1800 REM                                     (062E)
1810 IF gz=0 THEN gz=1:RETURN                (1713)
1820 IF gz=1 THEN gz=0                       (15DE)
1830 LOCATE gx,gy:PRINT MID$(a$(gy
),gx,1);:IF MID$(a$(gy),gx,1)=CHR$
(249)THEN LOCATE gx,gy:PEN 4:PRINT
 CHR$(249);:LOCATE gx,gy:PEN 5:PRI
NT CHR$(22)+CHR$(1);CHR$(250);CHR$
(22)+CHR$(0);                                (95F0)
1840 gx=gx+SGN(x-gx)                         (1FB2)
1850 gy=gy+SGN(y-gy)                         (1F51)
1860 LOCATE gx,gy:PEN 10:PRINT CHR
$(239);:PEN 1:LOCATE gx,gy:PRINT C
HR$(22)+CHR$(1);CHR$(236);CHR$(22)
+CHR$(0);:PEN 1                              (54F1)
1870 RETURN                                  (06CA)
1880 REM                                     (06CE)
1890 REM titelbild                           (100F)
1900 REM                                     (06F6)
1910 titel$="Zoff in der Galaxis"            (2339)
1920 INK 2,11:INK 0,0                        (0F41)
1930 INK 1,0:FOR n=1 TO 50                   (141B)
1940 PLOT INT(RND*640),INT(RND*400
),1:NEXT                                     (1FF6)
1950 INK 1,26                                (0AC2)
1960 LOCATE 2,8:PRINT CHR$(164)" M
ARTIN KEHR 1987"                             (2419)
1970 PEN 2                                   (078A)
1980 LOCATE 1,15:PRINT"BEWAHRE DAS
 UNIVER-   VERSUM VOR DEN       K
APUTTNIKS"                                   (4454)
1990 PEN 1                                   (07AE)
2000 FOR n=1 TO 19:LOCATE n,11:PEN
(n MOD 13)+3:PRINT MID$(titel$,n,1
):NEXT                                       (3B5A)
2010 FOR n=3 TO 15                           (0F7C)
2020 INK n,4+INT(RND*23)                     (16EF)
2030 IF JOY(0)=16 THEN RETURN                (107A)
2040 NEXT n:GOTO 2010                        (0FA8)
```

Alle Programme auf Disc erhältlich!

Bestell-Coupon Seite 81

der Eingabechecksummer für alle CPCs

Von Ihnen lange gefordert, ist er nun endlich da, der Checksummer für alle CPCs. Die nähere Besprechung dieses Eingabe-Prüfprogrammes erfolgt im Rahmen der Artikelreihe über die Programmierung in Maschinensprache. Hier nun nur die Bedienungsanleitung:

Tippen Sie das Programm sorgfältig ab, denn trotz der Prüfsummen am Ende jeder DATA-Zeile können Statementvertauschungen nicht erkannt werden. Nach dem Abtippen gleich abspeichern, damit bei einem eventuellen Eingabefehler nicht alles neu eingegeben werden muß!

Nach einem Programmlauf stehen zwei RSX-Befehle zur Verfügung: EIN und AUS. RSX-Befehle beginnen mit einem senkrechten Strich (Taste neben P mit Shift eingeben), und dann folgt der Befehlsname. Mit EIN wird der Checksummer aktiviert, mit AUS wieder abgeschaltet. Bei aktivem Checksummer erfolgt nach der Eingabe einer BASIC-Zeile die Ausgabe einer vierstelligen Hexademizalzahl, die in eckige Klammern eingeschlossen ist. Diese Prüfsumme muß mit der im einzugebenden Listing enthaltenen Summe (am Ende jeder BASIC-Zeile) übereinstimmen. Ist dies nicht der Fall, dann stimmt irgendetwas nicht! Geben Sie die Zeile neu ein (kann auch mit Edit Zeilennummer oder mittels der COPY-Cursor-Methode geschehen!).

Da unser Checker – im Gegensatz zu anderen Checksummern – auch die Zeilennummer mit überprüft, muß auch diese korrekt sein, damit die Prüfsumme stimmt.

Haben Sie ein Listing fehlerfrei abgetippt, dann speichern Sie dieses ab! Erst dann sollten Sie das abgetippte Programm starten, denn es könnte ja sein, daß das Programm dort, wo unser Checker seinen Platz hat, selbst aktiv wird oder eigene RSX-Befehle mit gleicher Aufrufsbefehlsfolge hat und es deshalb Fehler gibt!

Da die Firmware der drei CPC-Typen leider unterschiedlich ist, verhält sich unser "Eingabe-Prüfer" beim Befehl AUTO unterschiedlich. Dieser Befehl kann beim CPC 464 nicht benutzt wwerden, da dann keine Prüfsumme ausgegeben wird. Beim CPC 664 und 6128 ist der Einsatz dieses Befehles möglich.

Viel Glück mit dem Prüfprogramm! *LM*□

```
100 MEMORY &A2FF:'CHECKI.HEX
110 a=&A300:e=&A3F6:zb=1000:e=e+1
120 PRINT"Moment bitte...":FOR i=a TO e:
READ d$:IF LEFT$(d$,1)="&"THEN flag=1
130 IF(flag AND p<>VAL(d$))THEN PRINT"Fe
hler in Zeile "zb+1:END
140 IF(flag AND i=e)THEN 180
150 IF flag THEN i=i-1:zb=zb+1:p=0:d$=""
:flag=0:GOTO 170
160 POKE i,VAL("&"+d$):p=p+VAL("&"+d$)
170 IF i<e THEN NEXT i
180 typ=PEEK(&BD71):IF typ=&E8 THEN 260
190 IF typ=&55 THEN 220
200 IF typ=&14 THEN 240
210 PRINT"CPC nicht erkannt !":END
220 POKE &A331,&D4:POKE &A338,&69:POKE &
A339,&E8:POKE &A399,&D4:POKE &A39D,&AA:P
OKE &A39E,&E7:POKE &A3C0,&2
230 POKE &A3C1,&AC:POKE &A3CF,&5B:POKE &
A3E0,&5B:POKE &A3F2,&2:POKE &A3F3,&AC:PO
KE &A3EF,&8A:GOTO 260
240 POKE &A331,&CF:POKE &A338,&64:POKE &
A339,&E8:POKE &A399,&CF:POKE &A39D,&A5:P
OKE &A39E,&E7:POKE &A3C0,&2:POKE &A3C1,&
AC
250 POKE &A3CF,&5E:POKE &A3E0,&5E:POKE &
A3F2,&2:POKE &A3F3,&AC:POKE &A3EF,&8A
260 CALL &A300:END
1001 DATA 01,0A,A3,21,19,A3,CD,D1,&0329
1002 DATA BC,C9,12,A3,C3,C3,A3,C3,&0526
1003 DATA D4,A3,45,49,CE,41,55,D3,&043C
1004 DATA 00,00,00,00,00,CD,BF,A3,&022F
1005 DATA F5,C5,D5,E5,2A,EF,A3,28,&0558
1006 DATA 6A,E5,CD,98,A3,E1,30,63,&04CB
1007 DATA CD,04,EE,ED,53,EB,A3,CD,&055A
1008 DATA A3,E7,22,E5,A3,ED,43,E7,&054B
1009 DATA A3,09,22,E9,A3,21,00,00,&027B
1010 DATA 22,ED,A3,3A,E7,A3,32,EE,&0496
1011 DATA A3,D6,05,47,DD,21,40,00,&0303
1012 DATA B7,DD,7E,00,5F,3A,ED,A3,&043B
1013 DATA 83,B7,07,32,ED,A3,DD,23,&0403
1014 DATA 10,EE,2A,EB,A3,B7,85,B7,&04A9
1015 DATA 07,B7,84,32,ED,A3,2A,ED,&041B
1016 DATA A3,3E,20,CD,5A,BB,CD,5A,&040A
1017 DATA BB,3E,5B,CD,5A,BB,7C,CD,&047F
1018 DATA A2,A3,7D,CD,A2,A3,3E,5D,&046F
1019 DATA CD,5A,BB,E1,D1,C1,F1,C9,&060F
1020 DATA CD,04,EE,D0,CD,D2,E6,37,&054B
1021 DATA 9F,C9,5F,0F,0F,0F,0F,E6,&02E9
1022 DATA 0F,CD,B3,A3,7B,E6,0F,CD,&046F
1023 DATA B3,A3,C9,FE,0A,38,02,C6,&0427
1024 DATA 07,C6,30,CD,5A,BB,C9,CF,&0477
1025 DATA 98,AA,F7,3E,FF,32,00,AC,&0454
1026 DATA 01,03,00,21,F4,A3,11,3A,&0207
1027 DATA BD,ED,B0,C9,3E,00,32,00,&0393
1028 DATA AC,01,03,00,21,F1,A3,11,&0276
1029 DATA 3A,BD,ED,B0,C9,00,00,00,&035D
1030 DATA 00,00,00,00,00,00,00,A4,&00A4
1031 DATA AC,CF,98,AA,C3,1D,A3,&0440
```

Pull-Down-Menüs

GEM auf dem CPC

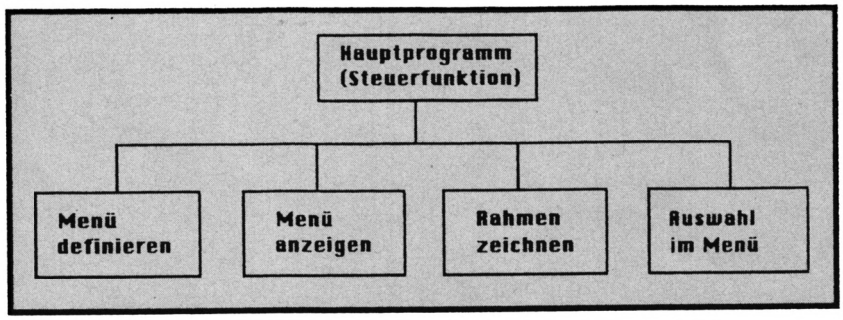
Komfortable Menü-Steuerung

Pull-Down-Menüs und Maussteuerung können Sie mit einfachen Mitteln auch in BASIC realisieren. Wir zeigen Ihnen, welche Eigenschaften Ihres CPC Sie kennen müssen und welche Methoden verwendet werden.

Die wenigsten CPC-Besitzer haben an ihr Gerät eine Maus angeschlossen. Daher wollen wir auf dieses praktische Eingabe-Gerät verzichten und stattdessen einen Joystick verwenden.
Unser Programm soll auf dem Bildschirm Menüs darstellen, in denen Sie mit einem Zeiger-Symbol einzelne Optionen auswählen können. Die Auswahl-Listen sollen automatisch umrandet, die gewählte Option invers dargestellt werden.

Erste Schritte zum Programm

Mit dieser Beschreibung ist bereits der wichtigste Schritt zur Erstellung eines Programms getan: Die gewünschten Funktionen wurden festgelegt. Sie können jetzt erste Überlegungen anstellen, welche Funktions-Blöcke das Listing enthalten muß.
Der wichtigste Block ist ohne Zweifel die Auswahl im Menü. Zwei weitere Routinen sind das Zeichnen des Rahmens und die Darstellung einer Auswahl-Liste. Außerdem ist es günstig, einen gesonderten Programmteil zur Erstellung eines Menüs zu verwenden.
Diese vier Funktionen realisieren Sie am besten in je einem Unterprogramm. Das gesamte Listing wird dadurch überschaubar. Auch Erweiterungen sind in der Regel ohne Probleme einzubauen, wenn ein Programm modular aufgebaut ist.



Eine Baumstruktur hilft, das Programm in Funktionsblöcke zu gliedern

Unterprogramme sorgen für den Durchblick

Die Subroutinen werden in irgendeiner Weise Daten austauschen müssen. Wie im Bereich der Hardware spricht man auch hier von Schnittstellen.
Solche Schnittstellen sind in BASIC Variablen, die Sie vor dem Aufruf des Unterprogramms mit Werten belegen, oder Werte, die das Unterprogramm erzeugt und die anschließend im Hauptteil von Bedeutung sind.
Doch beginnen wir der Reihe nach. Zunächst muß ein Pfeil-Symbol bestimmt werden. Um später einen kurzen PRINT-Befehl verwenden zu können, definieren wir das @-Symbol, den sogenannten Klammeraffen, entsprechend

um. Die Zeilen 150 und 160 übernehmen diese Aufgabe.

Daten in Arrays

Für die Auswahl in einem Menü sind einige Informationen erforderlich. Der Text einer Menüzeile ist in der Variablen **menueS()** enthalten. Da Sie außerdem keine festen Positionen für den Mauszeiger haben, muß ein Gültigkeitsbereich für die Bildschirmposition angegeben werden. Es wäre schließlich ein Unding, wenn der Anwender des Programms den richtigen Punkt erraten müßte. Dieser Bildschirmbereich wird in **xstart()**, **xende()**, **ystart()** und **yende()** abgelegt. Für ein Menü sollen maximal zehn Einträge zulässig sein. Zeile 180 dimensioniert die fünf benötigten Arrays. Nach dem Setzen der Grafikfarbe wird eine Startposition für den Mauszeiger festgelegt. Zeile 240 nimmt die Umrechnung in Grafik-Koordinaten vor.
Da der Bildschirm auf MODE 1 eingestellt ist, benötigt jedes Zeichen ein Quadrat von 16 mal 16 Bildpunkten. Ferner müssen Sie berücksichtigen, daß der Grafik-Schirm in der linken unteren Ecke mit den Koordinaten (0;0) beginnt, der Text-Schirm dagegen links oben mit (1;1).
Für die exakte Umrechnung der Werte erhalten Sie daher die Formel

```
x = (xt-1)*16
y = (25-yt)*16
```

Der Mauszeiger soll sich jedoch nicht exakt über einem Textzeichen befinden, sondern leicht nach rechts unten verschoben sein. Damit ist gewährleistet, daß die Spitze des Pfeils auf einen Buchstaben zeigt: Sie weist nämlich nach links oben. Addieren Sie zu den berechneten x- und y-Werten jeweils eine Acht, und diese Bedingung ist erfüllt.
Die Variablen **xschritt** und **yschritt**, die ebenfalls in Zeile

```
Routine: Menü definieren
Einsprung-Bedingungen: menue$, xstart$, xende$, ystart$, yende$ mit
   mindestens maxpunkte Elementen dimensioniert; DATA-Zeiger mit RESTORE
   auf richtige Zeile eingestellt.
Parameter: maxpunkte : Index der letzten Menü-Zeile
Funktion: Belegt Arrays für speicherinterne Darstellung eines Menüs
Rückgabe-Werte: menue$, xstart$, xende$, ystart$, yende$ enthalten bis
   zum Index maxpunkte die aus den DATA-Zeilen gelesenen und auf Grafik-
   Positionen umgerechneten Werte.
Aussprung-Bedingungen: DATA-Zeiger verändert.

Routine: Menü anzeigen
Einsprung-Bedingungen: keine
Parameter: keine
Funktion: Anzeige der Menü-Texte auf dem Bildschirm
Rückgabe-Werte: keine
Aussprung-Bedingungen: Menü ist auf dem Bildschirm dargestellt.

Routine: Rahmen zeichnen
Einsprung-Bedingungen: keine
Parameter: xr : Text-x-Koordinate für obere linke Ecke des Rahmens
           yr : Text-y-Koordinate für obere linke Ecke des Rahmens
           breite : Breite des Rahmens
           hoehe : Höhe des Rahmens
Funktion: Zeichnen eines Rahmens auf dem Bildschirm
Rückgabe-Werte: keine
Aussprung-Bedingungen: Rahmen ist auf dem Bildschirm dargestellt.

Routine: Auswahl
Einsprung-Bedingungen: keine
Parameter: x : Grafik-x-Koordinate des Mauszeigers
           y : Grafik-y-Koordinate des Mauszeigers
           xschritt : Schrittweite in x-Richtung
           yschritt : Schrittweite in y-Richtung
Funktion: Auswahl im Menü mittels Mauszeiger
Rückgabe-Werte: x : neue Grafik-x-Koordinate des Mauszeigers
                y : neue Grafik-y-Koordinate des Mauszeigers
                wahl : Index des angewählten Menü-Punktes
Aussprung-Bedingungen: Gewählter Punkt ist im Menü invers dargestellt.
```

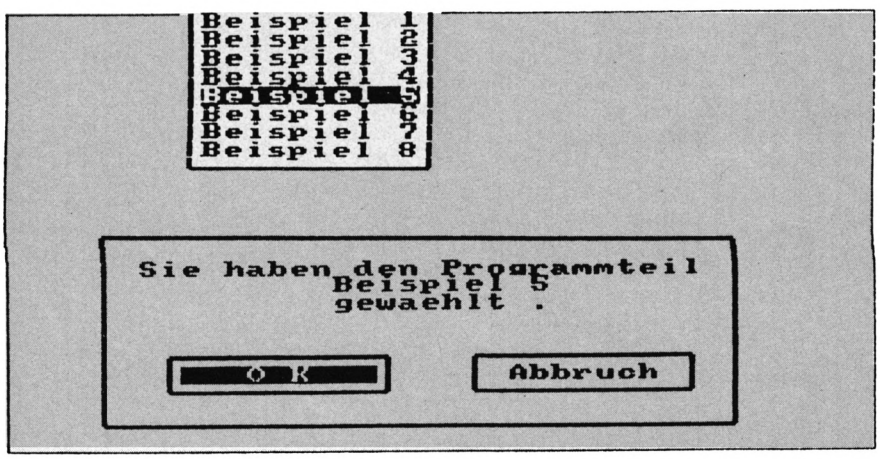
Dokumentation der Subroutinen ermöglicht eine spätere Wiederverwendung

240 definiert werden, stellen Parameter für das Unterprogramm zur Auswahl im Menü dar. Sie bestimmen, um wie viele Punkte der Mauszeiger bei Auslenkung des Joysticks jeweils springen soll.
Der Wert 16 entspricht genau einer Text-Position. Der Pfeil bewegt sich dadurch zwar etwas ruckartig, aber mit einer akzeptablen Geschwindigkeit über den Bildschirm. Durch die Bewegung um jeweils eine Buchstaben-Breite beziehungsweise -Höhe ist außerdem gewährleistet, daß Sie jedes Zeichen exakt ansteuern können.
Damit Sie sich die Mühe ersparen können, nach einer Bewegung des Zeigers den Bild-Hintergrund auf umständliche Art und Weise wiederherzustellen, aktiviert Zeile 260 den XOR-Modus.
Dieser spezielle Grafik-Modus bewirkt, daß jeder darzustellende Punkt mit dem Hintergrund Exklusiv-ODER-verknüpft wird. Das hat unter anderem zur Folge, daß zweimaliges Setzen desselben Punktes diesen Punkt wieder löscht, ohne den Hintergrund zu zerstören. Damit sind ideale Voraussetzungen für die Darstellung eines beweglichen Pfeils gegeben.
Nach all diesen Vorarbeiten soll endlich ein Menü definiert werden. Zeile 280 schafft die Voraussetzungen für den Aufruf des entsprechenden Unterprogramms.

Die Variable **maxpunkte** erhält den Wert acht. So viele Zeilen enthält das erste Menü. Die RESTORE-Anweisung sorgt dafür, daß die zugehörigen Texte und Koordinaten aus den richtigen DATA-Zeilen gelesen werden. Danach wird das zugehörige Unterprogramm aufgerufen, das sich ab Zeile 500 befindet.

Strenge Trennung der Ebenen

Bereits Zeile 510 ist auffällig. Entgegen der üblichen Konvention, die Variable i als Zählvariable einer FOR-Schleife zu verwenden, wurde hier die Bezeichnung ii gewählt.
Der Grund dafür ist, daß i ein gängiger Bezeichner für derartige



So sieht der Bildschirm-Aufbau unseres Demo-Programms aus

Variablen ist. Beim derzeitigen Stand der Programm-Entwicklung läßt sich nicht absehen, ob ein späterer Aufruf des Unterprogramms innerhalb einer Schleife erfolgt. Es ist daher günstig, Variablen-Namen mit einem Buchstaben durch das entsprechende Doppelzeichen zu ersetzen.
Soll innerhalb eines Unterprogramms eine weitere Subroutine aufgerufen werden, ist es empfehlenswert, die dort vorkommenden Bezeichner mit jeweils drei Buchstaben zu versehen, also zum Beispiel iii. Außerdem sollte strikt getrennt werden zwischen Unterprogrammen erster, zweiter und eventuell noch dritter Ebene.
Eine Subroutine erster Ebene wird nur vom Hauptprogramm aufgerufen, ein Unterprogramm zweiter Ebene nur von einer Routine erster Ebene und so weiter. Verzweigungen, bei denen eine Ebene übersprungen wird, sind zu vermeiden.
Auch bei der Reihenfolge der Routinen innerhalb des Programm-Textes sollten Sie diese Trennung deutlich machen. Die Subroutinen der einzelnen Ebenen stehen hintereinander, wobei die Unterprogramme auf der niedersten Ebene die höchsten Zeilennummern erhalten. Dahinter folgen nur noch die DATAs.

Vorausplanung ist unentbehrlich

Eine große Hilfe bei der Planung eines Programms ist die Darstellung der benötigten Unterprogramme in Form eines Baumes. Für unser Listing ist er recht einfach, da nur Subroutinen erster Ebene verwendet werden. Sie finden ihn in Abbildung 1.

Dokumentation ist wichtig

Wollen Sie eine Routine wiederverwenden oder das Programm später überarbeiten, so sollten Sie eine Aufstellung der Parameter anlegen. Sinnvolle Stichpunkte sind:

- **Einsprungbedingungen**
- **Parameter**
- **Funktion der Routine**
- **Rückgabe-Werte**
- **Aussprungbedingungen.**

Die Dokumentation aller Routinen unseres Programms finden Sie in Abbildung 2.

Ein Menü wird definiert

Was leistet nun aber die Routine „Menü definieren"? Innerhalb der FOR-Schleife werden die Texte und zulässigen Zeiger-Positionen aus den DATA-Zeilen gelesen.
Um die Programmierung zu erleichtern, stehen die Positionen dort als Text-Koordinaten. Die Zeilen 550 bis 580 übernehmen die Umrechnung in Grafik-Werte. Damit hat dieses Unterprogramm seine Aufgabe erfüllt. Die Arrays **menue$()**, **xstart()**, **xende()**, **ystart()** und **yende()** enthalten bis zum Index **maxpunkte** sinnvolle Werte. Das Hauptprogramm kann in Zeile 290 fortfahren.
Die nächste Zeile, die einen ausführbaren BASIC-Befehl enthält, hat die Nummer 300. Dort finden Sie einen Einsprung in das Unterprogramm zur Anzeige eines Menüs. Offensichtlich benötigt es keine Parameter.
Zeile 630 berechnet die Text-Positionen der einzelnen Menüpunkte. Die vorher durchgeführte Umrechnung in Grafik-Positionen muß hier rückgängig gemacht werden. Das geschieht durch Auflösung der oben genannten Formeln nach xt beziehungsweise yt. Sie erhalten:

```
xt = (x-8)/16+1
yt = 25-(y-8)/16
```

Anstelle von x und y werden hier die entsprechenden Werte aus den Arrays verwendet.
Zeile 640 stellt den Text einer Menüzeile an der errechneten Position dar. Nach Durchlaufen der Schleife befindet sich die komplette Liste auf dem Bildschirm. Das Hauptprogramm fährt mit einem weiteren Unterprogramm-Aufruf fort, und zwar in Zeile

320. Die nächste Subroutine soll einen Rahmen um das Menü ziehen. Dafür sind mehrere Parameter notwendig.
Die Text-Koordinaten für die linke obere Ecke des Rahmens werden in **xr** und **yr** abgelegt. Die Variablen **breite** und **hoehe** geben genau das an, was ihre Bezeichner aussagen. Dabei werden auch die zwei Positionen berücksichtigt, die links und rechts beziehungsweise oben und unten von den Linien der Umrandung benötigt werden. Der Befehl

```
GOSUB 670
```

springt ins Unterprogramm.
Diese Subroutine ist etwas trickreicher und ermöglicht die Darstellung echter Pull-Down-Menüs. Während bei der Anzeige der Menü-Zeilen vorausgesetzt wurde, daß sie auf dem Bildschirm Platz haben, ist dies hier nicht der Fall.
Für jede Zeile des Rahmens wird die vertikale Position überprüft. Stellt sich heraus, daß das obere oder untere Ende des Bildschirms erreicht ist, so wird die entsprechende Zeile nicht dargestellt.
Da der Zeichensatz des CPC bereits alle Symbole für die Darstellung eines Rahmens enthält, erübrigen sich weitere Definitionen. Die CHR$-Codes, die hier verwendet wurden, können Sie in Ihrem Handbuch nachschlagen.
Beim Ziehen der geraden Linien müssen die zwei Zeichen, die der Rand in Anspruch nimmt, berücksichtigt werden. Der erste Parameter der STRING$-Anweisung lautet daher

```
breite-2
```

Die ii-Schleife läuft von yr+1 bis yr+hoehe-2.
Das Menü befindet sich jetzt vollständig auf dem Bildschirm. Daher kann das Hauptprogramm die Subroutine zur Auswahl mittels Mauszeiger aufrufen.
Obwohl Zeile 340 zu der Annahme verleitet, das Unterprogramm benötige keine Parameter, ist dies nicht der Fall. Sie wurden bereits in Zeile 240 gesetzt.
Dies ist ein Beispiel für Werte, die ans Unterprogramm übergeben, dort verändert und ans Hauptprogramm zurückgegeben werden.
Der nächste Aufruf des Unterprogramms geschieht mit den veränderten Variablen-Inhalten. Die Startwerte werden aus diesem Grund zu Beginn des Hauptprogramms oder in einer eigenen Initialisierungs-Routine gesetzt.
Die Auswahl-Routine hat einen relativ komplizierten Aufbau. Sie hat auch einige Aufgaben zu erfüllen: das Positionieren des Mauszeigers, die Ermittlung der angewählten Menü-Zeile sowie deren Anzeige in inverser Darstellung.
Um den Mauszeiger positionieren zu können, wird die Textausgabe in Zeile 750 auf die Grafik-Position geleitet. Der TAG-Befehl des CPC-BASIC macht es überflüssig, jeden einzelnen Punkt des Pfeiles zu plotten und beschleunigt dadurch die Ausgabe erheblich.
Die Variablen **xalt** und **yalt** enthalten die jeweils letzte Position des Mauszeigers. Da im Moment keine „alte" Position bekannt ist, werden beide mit −1 belegt. Da dieser Punkt aufgrund der folgenden Abfrage nie erreicht werden kann, ist gewährleistet, daß die angenommene alte Position nicht mit der neuen Stelle übereinstimmt. Weshalb dies nötig ist, wird später erklärt.
Die Variable **aktion** ist ein sogenanntes Flag. Ein Flag gibt an, ob eine bestimmte Bedingung erfüllt ist oder nicht. In BASIC wird die Aussage „erfüllt" oder „wahr" in der Regel durch den Wert −1 ausgedrückt; eine Null steht für „nicht erfüllt" oder „falsch".
Die Bedingung, die durch das Flag **aktion** dargestellt wird, lautet „Feuertaste betätigt". Die Joystick-Abfrage in Zeile 800 setzt gegebenenfalls diesen Wert auf −1. Zugleich werden in dieser Zeile durch einige logische Verknüpfungen die neuen Werte für **x** und **y** ermittelt. Auf die Funktionsweise dieser Joystick-Abfrage gehen wir in einem gesonderten Beitrag ein.
Innerhalb der WHILE-Schleife von Zeile 770 bis 850 wird auch der Mauszeiger positioniert. Zeile 780 sorgt dafür, daß der Zeiger nur dann gesetzt wird, wenn sich seine Position verändert hat. Dieselbe Aufgabe hat Zeile 840. Sie entfernt den Pfeil auch vom Bildschirm, wenn der Feuerknopf gedrückt wurde.
Da der XOR-Modus eingeschaltet ist, bewirkt ein zweimaliges Setzen des Symbols die Wiederherstellung des Hintergrundes. Wurde der Joystick nicht betätigt, die Zeiger-Position also nicht verändert, so soll ein Löschen und sofortiges neues Plazieren verhindert werden. Das Pfeil-Symbol würde dadurch unangenehm flackern. Diese Aufgabe erfüllen die beiden IF-Abfragen. Hat sich die Position geändert, so wird gelöscht und neu plaziert. Hat hingegen keine Veränderung stattgefunden, so bleibt der Mauszeiger stehen.
Wurde schließlich der Feuerknopf gedrückt, so ist die WHILE-Schleife beendet. Ein TAGOFF sorgt dafür, daß Texte wieder an den normalen Positionen ausgegeben werden. Der Mauszeiger ist verschwunden.
Seine Position hingegen ist immer noch in den Variablen **x** und **y** enthalten. Sie muß jetzt mit allen gespeicherten zulässigen Positionen verglichen werden, damit der gewählte Menü-Punkt ermittelt werden kann.
In Zeile 870 werden dazu zwei Variablen definiert. Das Flag **korrekt** gibt nach der Überprüfung an, ob der Zeiger an einer gültigen Position steht. In **wahl** befindet sich der Index der gewählten Menü-Zeile.
Die FOR-Schleife von 880 bis 900 erledigt die Überprüfung. Befinden sich sowohl der x- als auch der y-Wert des Mauszeigers inner-

halb der für einen Menüpunkt vorgegebenen Gültigkeits-Grenzen, so wird das Flag **korrekt** auf –1 gesetzt und der Index dieses Menüpunktes in **wahl** abgelegt.
Zeile 910 fragt das Flag ab. War die Wahl nicht korrekt, so ertönt ein Piepsen, und das Unterprogramm wird von vorne durchlaufen. War sie korrekt, so ist die Routine beendet.

Gestaltung des Bildschirms

Die Zeilen 350 bis 400 demonstrieren, wie Sie ein zweites Menü aufbauen können, in dem die eben vorgenommene Auswahl Bestandteil des Textes ist.
Hier wird das neue Menü erst definiert, nachdem ein Teil des zweiten Menüs bereits auf dem Bildschirm steht. Für die Darstellung des ersten Teils wird die entsprechende Variable aus **menue$()** verwendet.
Eine weitere Möglichkeit besteht darin, den gewünschten Text in einer Hilfsvariablen zwischenzuspeichern, das neue Menü zu definieren und die Bildschirm-Darstellung unter Verwendung der Hilfsvariablen aufzubauen.
Der Rest des Programms verwendet die Unterprogramm-Aufrufe, die Sie bereits kennengelernt haben. Beachten Sie besonders die Zeilen 410 bis 440: Durch geschickte optische Gestaltung können Sie Ihren Programmen ein sehr professionelles Aussehen verleihen. Abbildung 3 liefert den Beweis.

Verwendung in eigenen Programmen

Die hier vorgestellten Routinen können Sie in eigenen Programmen verwenden oder um zusätzliche Funktionen erweitern.
Wenn Sie Ihr Programm modular aufbauen, bereitet das keine besonderen Schwierigkeiten.

AE □

```
100 'CPC-GEM                                    (0EC3)
110 '                                           (07E7)
120 'Bildschirmfarben                           (17A7)
130 INK 0,0:PAPER 0:INK 1,12:PEN 1
:BORDER 0:MODE 1                                (1B11)
140 'Mauspfeil auf Klammeraffe                  (203C)
150 SYMBOL AFTER 64                             (09E6)
160 SYMBOL 64,&F0,&E0,&F0,&B8,&C,&
6,&3,&1                                         (2863)
170 'Menue-Arrays vorbereiten                   (1FD7)
180 DIM menue$(10),xstart(10),xend
e(10),ystart(10),yende(10)                      (48FD)
190 'Grafik-Farbe                               (1309)
200 INK 2,24:PLOT 900,900,2                     (1583)
210 'Startposition Textschirm                   (1F45)
220 xt=20:yt=10                                 (1608)
230 'Startposition Grafikschirm                 (21D8)
240 x=(xt-1)*16+8:y=(25-yt)*16+8:x
schritt=16:yschritt=16                          (4B3D)
250 'XOR-Modus an                               (13FD)
260 PRINT CHR$(23);CHR$(1);                     (1395)
270 'Menue definieren                           (1760)
280 maxpunkte=8:RESTORE:GOSUB 500               (1ADA)
290 'Menue anzeigen                             (150A)
300 GOSUB 610                                   (0942)
310 'Rahmen zeichnen                            (16AA)
320 xr=9:yr=0:breite=12:hoehe=10:G
OSUB 670                                        (3244)
330 'Auswahl                                    (0E3B)
340 GOSUB 740                                   (09A6)
350 'Menue 2 erster Teil anzeigen               (2343)
360 LOCATE 7,15:PRINT"Sie haben de
n Programmteil":LOCATE 16,16:PRINT
 menue$(wahl):LOCATE 16,17:PRINT"g
ewaehlt ."                                      (57C7)
370 'Menue 2 definieren                         (1977)
380 maxpunkte=2:RESTORE 1050:GOSUB
 500                                            (1D0C)
390 'Menue 2 zweiter Teil anzeigen
                                                (242F)
400 GOSUB 610                                   (0909)
410 'Menue 2 Rahmen zeichnen                    (1E73)
420 xr=8:yr=20:breite=11:hoehe=3:G
OSUB 670                                        (322E)
430 xr=22:yr=20:breite=11:hoehe=3:
GOSUB 670                                       (3358)
440 xr=5:yr=13:breite=30:hoehe=12:
GOSUB 670                                       (33BA)
450 'Auswahl                                    (0E2C)
460 GOSUB 740                                   (0995)
470 'Ende                                       (0B91)
480 GOTO 480                                    (09B9)
490 '                                           (07DF)
500 'UP Menue definieren                        (1AA5)
510 FOR ii=1 TO maxpunkte                       (1AB0)
520 READ menue$(ii),xstart(ii),xen
de(ii),ystart(ii),yende(ii)                     (5715)
530 NEXT ii                                     (0BB0)
540 FOR ii=1 TO maxpunkte                       (1AED)
550 xstart(ii)=(xstart(ii)-1)*16+8
                                                (2F73)
560 xende(ii)=(xende(ii)-1)*16+8                (2D1A)
570 ystart(ii)=(25-ystart(ii))*16+
8                                               (30ED)
580 yende(ii)=(25-yende(ii))*16+8               (2E33)
590 NEXT ii                                     (0B29)
600 RETURN                                      (06D9)
610 'UP Menue anzeigen                          (18E0)
620 FOR ii=1 TO maxpunkte                       (1A8C)
630 xt=(xstart(ii)-8)/16+1:yt=25-(
ystart(ii)-8)/16                                (45BD)
640 LOCATE xt,yt:PRINT menue$(ii)               (2270)
650 NEXT ii                                     (0BA1)
660 RETURN                                      (0650)
670 'UP Rahmen zeichnen                         (192D)
680 IF yr>0 AND yr<26 THEN LOCATE
xr,yr:PRINT CHR$(150);STRING$(brei
te-2,CHR$(154));CHR$(156);                      (4AF7)
690 FOR ii=yr+1 TO yr+hoehe-2                   (249D)
700 IF ii>0 AND ii<26 THEN LOCATE
xr,ii:PRINT CHR$(149);STRING$(brei
te-2,CHR$(9));CHR$(149);                        (4935)
710 NEXT ii                                     (0B18)
720 IF yr+hoehe-1>0 AND yr+hoehe-1
<26 THEN LOCATE xr,yr+hoehe-1:PRIN
T CHR$(147);STRING$(breite-2,CHR$(
154));CHR$(153);                                (6B82)
730 RETURN                                      (06DC)
740 'UP Auswahl                                 (11E0)
750 TAG                                         (0621)
760 xalt=-1:yalt=-1:aktion=0                    (2629)
770 WHILE NOT aktion                            (104C)
780 IF x<>xalt OR y<>yalt THEN MOV
E x,y:PRINT"0";                                 (3017)
790 xalt=x:yalt=y                               (1ED8)
800 x=x-xschritt*(((JOY(0)AND 8)<>
0)-((JOY(0)AND 4)<>0)):y=y-yschrit
t*(((JOY(0)AND 1)<>0)-((JOY(0)AND
2)<>0)):aktion=((JOY(0)AND 16)<>0)
                                                (8575)
810 IF x<0 THEN x=x+xschritt ELSE
IF x>640 THEN x=x-xschritt                      (43D3)
820 IF y<0 THEN y=y+yschritt ELSE
IF y>400 THEN y=y-yschritt                      (4314)
830 FOR i=1 TO 50:NEXT i'Verzoeger
ung                                             (2326)
840 IF x<>xalt OR y<>yalt OR aktio
n THEN MOVE xalt,yalt:PRINT"0";                 (40B9)
850 WEND                                        (06FE)
860 TAGOFF                                      (0602)
870 korrekt=0:wahl=0                            (1B0F)
880 FOR ii=1 TO maxpunkte                       (1A95)
890 IF(x)=xstart(ii)AND x(=xende(i
i))AND(y)=ystart(ii)AND y(=yende(i
i))THEN korrekt=-1:wahl=ii                      (7BA6)
900 NEXT ii                                     (0B96)
910 IF NOT korrekt THEN PRINT CHR$
(7);:GOTO 740                                   (1E12)
920 xt=(xstart(wahl)-8)/16+1:yt=25
-(ystart(wahl)-8)/16                            (49F8)
930 LOCATE xt,yt:PRINT CHR$(24);me
nue$(wahl);CHR$(24);                            (334E)
940 RETURN                                      (0681)
950 '                                           (0779)
960 'Menue-Daten                                (1287)
970 DATA "Beispiel 1",10,19,1,1                 (1D0C)
980 DATA "Beispiel 2",10,19,2,2                 (1DBC)
990 DATA "Beispiel 3",10,19,3,3                 (1D2B)
1000 DATA "Beispiel 4",10,19,4,4                (1D73)
1010 DATA "Beispiel 5",10,19,5,5                (1D3B)
1020 DATA "Beispiel 6",10,19,6,6                (1D03)
1030 DATA "Beispiel 7",10,19,7,7                (1D51)
1040 DATA "Beispiel 8",10,19,8,8                (1D99)
1050 'Daten Menue 2                             (1440)
1060 DATA "   O K   ",9,17,21,21                (1D40)
1070 DATA " Abbruch ",23,31,21,21               (1EA4)
```

Die Fremden
ohne Heimat
besuchen

caritas
bei
uns

caritas

Jetzt gibt es Deutschlands erste Commodore-Zeitschrift mit Programm-Diskette für Ihren 64er und 128er!

COMMODORE
DISC
C64/
C128

Bis zu 180 KB Programme ohne Abtippen!

COMMODORE DISC

An guten Kiosken und im Bahnhofs-Buchhandel

COMMODORE DISC

Mathe mit dem CPC

Matrizen und Vektoren

Der CPC löst mathematische Probleme

Welcher Schüler plagt sich nicht mit Matrizen und Vektor-Rechnung? Und dies, obwohl zu Hause der CPC steht, der solche Aufgaben spielend lösen kann. Vorausgesetzt, Sie haben das richtige Programm dazu.

Vektoren und Matrizen sind mehr als graue Theorie. Anhand von Beispielen wollen wir Ihnen zeigen, wie diese mathematischen Verfahren in der Praxis eingesetzt werden und wie Sie solche Gleichungen mit Ihrem CPC lösen können.
Eine Matrix können Sie als eine Tabelle oder ein Gleichungssystem folgenden Aussehens betrachten:

```
 3a + 6b + 5c = 20
10a + 3b + 2c = 15
15a + 10b + 9c = 25
```

Solche Tabellen begegnen einem oft im Schulunterricht. Dann heißt es, mit einem mühsamen und zeitraubenden Additionsverfahren die Gleichungen aufzulösen und ineinander einzusetzen. Bis die Ergebnisse für a, b und c ermittelt sind, vergeht viel Zeit. Unser Listing schafft hier Abhilfe. Doch bevor wir den Aufbau des Programms besprechen, sollte am Beispiel des obigen Gleichungssystems erläutert werden, was Vektoren und Matrizen sind.
Die Spalten der Tabelle können wir vereinfacht als Vektoren bezeichnen. Die gesamte Matrix besteht in diesem Fall aus drei Spalten, also drei Vektoren. Die gesuchten Größen sind die Faktoren a, b und c.
Multiplizieren wir jeden Vektor mit dem zugehörigen Faktor, so wird die Tabelle wahr. Jede Zeile kann dann als richtig gelöste Gleichung angesehen werden. Doch wie erhalten wir die Werte der Faktoren? Betrachten wir hierzu folgende Matrix:

```
1a + 0b + 0c = –0,078
0a + 1b + 0c = 12,812
0a + 0b + 1c = –11,32
```

Wenn man es schafft, jedes Gleichungssystem (oder jede Matrix) auf diese Form zu bringen, so

	Disketten	Farbbänder	Papier	Einnahmen gesamt
Januar	15	9	13	330 DM
Februar	20	3	12	300 DM
März	-	12	2	90 DM

Wieviel verdient der Händler pro Stück?

wird ersichtlich, daß in jeder Zeile nur noch eine Variable steht. Vereinfacht ließe sich dies schreiben als

```
a = –0,078
b = 12,812
c = –11,32
```

Das System wäre damit gelöst. Eine Matrix, bei der alle Zahlen bis auf die Diagonale gleich Null sind, bezeichnet man als **Einheitsmatrix.** Die oben genannten Werte sind übrigens die Lösungen des eingangs gezeigten Gleichungssystems. Sie können dies mit Hilfe unseres Listings überprüfen.

Beispiele

Ein Beispiel aus der Wirtschaft zeigt, wie solche Gleichungssysteme in der Praxis auftreten können.
Ein kleiner Versandhandel, der Computer-Zubehör vertreibt, erstellt die Bilanz des ersten Quartals 1988. Seine Verkaufstabelle sieht für die Monate Januar bis März so aus, wie in Abbildung 1 gezeigt.
Die Stückzahlen der einzelnen Artikel und die Gesamteinnahmen sind bekannt. Wieviel aber wurde pro verkaufter Diskette und pro verkauftem Farbband eingenommen? Oder, mathematisch exakt formuliert: Wie heißt der gemeinsame Faktor für die drei Monate pro Spalte, der die Tabelle wahr werden läßt?
Unser Programm löst dieses Problem. Geben Sie die Werte ein, wobei auch die Null für die im März verkauften (oder eher nicht verkauften) Disketten nicht fehlen darf. Die Werte für a, b und c sind die Einnahmen pro Diskette, pro Farbband beziehungsweise pro Karton Papier.
Ein weiteres Beispiel können wir der Ernährungs-Wissenschaft entnehmen. Regelmäßig zur Sommerszeit sind in allen einschlägigen Frauenmagazinen die berühmten Diäten zu finden. Von Jahr zu Jahr werden sie schmackhafter,

	Wurst	Kart.p.	Kohlr.	Nougat	Sollwert/Tag
Energie (kJ)	1318	1318	85	2406	11000 kJ
Eiweiß (g)	13	4	2	9	160 g
Kohleh. (g)	0	80	2	55	260 g
Fett (g)	27	0	0	33	105 g

Auch Diätpläne können mathematisch errechnet werden

bekömmlicher und einfacher durchzuhalten.
Dabei muß man sich in einem bestimmten Zeitraum an einen Diätplan halten, der Menge und Art der erlaubten Speisen angibt. Daß dabei der individuelle Geschmack zu kurz kommt, muß hingenommen werden.
Oder auch nicht: Es wäre doch viel besser, einen persönlichen Diätplan zusammenzustellen. Einige Eckdaten werden jedoch benötigt:

- die Summe der Kilojoule (kJ), die am Tag verzehrt werden dürfen,
- die zur Auswahl stehenden Speisen und deren Eiweiß-, Kohlehydrat- und Fettwerte,
- die Summe in Gramm, die pro Tag an Eiweiß, Kohlehydraten und Fett benötigt wird.

Auch hier hilft unser Programm weiter. Nehmen wir als Beispiel eine Person, die sich mit 11.000 kJ pro Tag durch Fleischwurst, Kartoffelpuffer, Kohlrabi und Nougat-Schokolade ernähren will. Täglich benötigt sie jedoch mindestens 160 Gramm Eiweiß, 260 Gramm Kohlehydrate und 105 Gramm Fett.
Da sich alle Nährstoffwerte, die in Tabellen zu finden sind, auf eine Menge von 100 Gramm beziehen, muß der Diätplan in die in Abbildung 2 gezeigte Tabelle umgewandelt werden.
Sie können das Programm nun dazu benutzen, die Matrix zu berechnen. Als Ergebnis sollten Sie erhalten:

```
a = 3,4
b = 1,7
c = 52,7
d = 0,4
```

Da sich die Nährstoffwerte auf 100 Gramm beziehen, müssen Sie das Ergebnis noch mit 100 multiplizieren. Der Diätplan würde also lauten:

```
340 g Fleischwurst
170 g Kartoffelpuffer
5.800 g Kohlrabi
40 g Nougat
```

Da 5,8 Kilogramm Kohlrabi nicht jedermanns Sache sind, sollten Sie den Diätplan ein wenig ändern.
Weitere Einsatzmöglichkeiten für die Matrizenrechnung finden sich in der Wirtschaft (Produktionstechnik), Biologie (Populationsgenetik), Physik (Gleichstrom-Netzwerke) und Verkehrsplanung. Sie alle beruhen auf dem Prinzip verknüpfter Bedingungen, die sich als mathematische Gleichungen formulieren lassen.

Das Programm

Die Zeilen 8100 bis 8299 unseres Listings überprüfen, ob irgendeine Spalte oder Zeile keinen Wert oder null enthält. Dies entspräche einer nicht definierten Gleichung oder einem nicht vorhandenen Vektor. Das System ist in diesem Fall nicht lösbar.
Die Diagonale des Matrix-Feldes wird zum Lösen des Systems immer wieder multipliziert und dividiert. Es versteht sich also von selbst, daß dort keine Null stehen darf. Ist dies dennoch der Fall, so stellen die Zeilen 8300 bis 8399 die Matrix so um, daß die Werte der Diagonalen ungleich null sind.
Das Auflösen der Matrix nach dem Gauß'schen Verfahren besorgen die Zeilen 8500 bis 8699.

Durch äquivalente Umformungen und Addition der Gleichungen wird die Einheits-Matrix errechnet. Die Zeilen 8800 bis 8999 überprüfen, ob die Einheits-Matrix bereits vorliegt.
Der letzte Schritt ist die Division der Diagonalen, damit sie den Wert eins erreicht.
Die Meldung, das System sei nicht lösbar, ist nicht so zu deuten, daß Sie bei der Eingabe einen Fehler gemacht hätten. Vielmehr geht das Gleichungssystem nicht auf, so daß eventuell andere Lösungsmöglichkeiten gefunden werden müssen.

Andreas Goll/AE □

```
1 '********************************
                                    (26B2)
2 '* Matrizen aufloesen           *
                                    (26C8)
3 '*                 Version 2.5 *
                                    (2632)
4 '* von Andreas Goll             *
                                    (2697)
5 '* (c) 1988 Gsoft Inc.          *
                                    (26DA)
6 '*       The Art of Software    *
                                    (2626)
7 '********************************
                                    (26BE)
10 GOSUB 60000'Initialisierung      (1AE2)
20 GOSUB 1000'Matrize Eingabe       (1AA9)
30 GOSUB 8000'Matrize ausrechnen    (1DD8)
40 GOSUB 10000'Ergebnis ausgeben    (1CF2)
50 '                                (076F)
60 PRINT" (c) 1988 by Gsoft Inc."   (1FEC)
70 PRINT"              The Art of S
oftware"                            (2840)
80 END                              (0603)
1000 '*****************************
                                    (2356)
1001 '* Matrize eingeben           *
                                    (230B)
1002 '*****************************
                                    (235A)
1003 '                              (07E3)
1010 CLS#0:CLS#1                    (0C94)
1020 INPUT#1,"Groesse der Matrize:
 ",anz                              (2772)
1030 IF anz<2 OR anz>12 THEN 1020   (1C32)
1040 FOR n=1 TO anz                 (13D3)
1050 LOCATE 2,n*2:PRINT CHR$(24);:
PRINT USING"##";n:PRINT CHR$(24);   (2C4B)
1060 LOCATE n*5,1:PRINT CHR$(24);:
PRINT USING"##";n:PRINT CHR$(24);   (2C0F)
1070 LOCATE(anz+1)*5,n*2:PRINT"="   (1EE1)
1080 NEXT                           (0637)
1090 LOCATE(anz+1)*5+3,1:PRINT CHR
$(24)"Ergebnis"CHR$(24)             (2E62)
1100 FOR zeile=1 TO anz             (17FC)
1110 FOR spalte=1 TO anz+1          (1A0B)
1120 LOCATE spalte*5-(spalte=anz+1
)*3,zeile*2:INPUT"",w(zeile,spalte
)                                   (5094)
1130 NEXT                           (069B)
1140 NEXT                           (06AF)
1150 CLS#0:CLS#1                    (0CAC)
8000 '*****************************
                                    (2321)
8001 '* Matrize ausrechnen         *
                                    (2310)
8002 '*****************************
                                    (2325)
8003 '                              (07B0)
8010 GOSUB 8100'Pruefen, ob Loesun
g moeglich                          (2729)
8020 IF unmoeg THEN PRINT#1,"Loesu
ng nicht moeglich I":RETURN         (309E)
8030 GOSUB 8300'Moeglicherweise ta
uschen                              (23C3)
8040 GOSUB 8500'Matrize addieren    (1BC1)
8050 IF unmoeg THEN PRINT#1,"Loesu
ng nicht moeglich II":RETURN        (318B)
8060 GOSUB 8700'Diagonale dividier
en                                  (1FEF)
8070 RETURN                         (0651)
8100 '                              (0771)
8110 'Pruefen ob Loesung moeglich   (22E4)
8120 '                              (0799)
8130 unmoeg=0:null=0                (1A88)
8140 FOR spalte=1 TO anz            (1856)
8150 FOR zeile=1 TO anz             (172B)
8160 IF w(zeile,spalte)=0 THEN nul
l=null+1                            (32E7)
8170 IF w(spalte,zeile)=0 THEN nul
l1=null1+1                          (3470)
8180 NEXT                           (06C9)
8190 IF null=anz OR null1=anz THEN
 unmoeg=1:RETURN                    (327D)
8200 null=0:null1=0                 (193C)
8210 NEXT                           (0606)
8215 GOSUB 9000'Matrize anzeigen    (1B74)
8220 RETURN                         (067F)
8300 '                              (0703)
8310 'Moeglicherweise tauschen      (1FB5)
8320 '                              (072A)
8330 rflag=0                        (0F26)
8340 FOR spalte=1 TO anz            (18E6)
8350 IF w(spalte,spalte)<>0 THEN 8
410 ELSE rflag=1                    (31AD)
8360 IF spalte=anz THEN tausch=1 E
LSE tausch=spalte+1                 (3968)
8370 FOR tloop=1 TO anz+1           (19C0)
8380 zwischen=w(spalte,tloop)       (29F7)
8390 w(spalte,tloop)=w(tausch,tloo
p):w(tausch,tloop)=zwischen         (5B96)
8400 NEXT                           (0682)
8410 NEXT                           (0696)
8420 IF rflag THEN 8330             (1268)
```

```
8425 GOSUB 9000'Matrize anzeigen  <1B19>
8430 RETURN                       <0623>
8500 '                            <0794>
8510 ' Matrize addieren           <1890>
8520 '                            <07BC>
8530 FOR spalte=1 TO anz          <1863>
8540 FOR zeile=1 TO anz           <173A>
8550 IF zeile=anz AND spalte=anz T
HEN 8650                          <2AF6>
8560 IF zeile=spalte THEN zeile=ze
ile+1                             <2C30>
8570 IF w(zeile,spalte)=0 THEN 863
0                                 <24DC>
8580 div=w(zeile,spalte)/w(spalte,
spalte)                           <3EFD>
8590 FOR addier=1 TO anz+1        <1A58>
8600 w(spalte,addier)=w(spalte,add
ier)*div                          <3F9C>
8610 w(zeile,addier)=w(zeile,addie
r)-w(spalte,addier)               <5084>
8620 NEXT                         <063B>
8630 NEXT                         <064F>
8640 NEXT                         <0663>
8650 GOSUB 8800'Pruefen, ob Ergebn
is fertig                         <261F>
8660 IF nochmal=1 THEN 8500       <1639>
8665 GOSUB 9000'Matrize anzeigen  <1BFA>
8670 RETURN                       <0603>
8700 '                            <0724>
8710 ' Diagonale dividieren       <1C0E>
8720 '                            <074D>
8730 FOR zeile=1 TO anz           <17B6>
8740 w(zeile,anz+1)=w(zeile,anz+1)
/w(zeile,zeile)                   <4C57>
8750 w(zeile,zeile)=w(zeile,zeile)
/w(zeile,zeile)                   <4C32>
8760 NEXT                         <0655>
8765 GOSUB 9000'Matrize anzeigen  <1BC2>
8770 RETURN                       <06CD>
8800 '                            <07ED>
8810 'Pruefen, ob Loesung gefunden
                                  <233B>
8820 '                            <0715>
8830 nochmal=0                    <11E2>
8840 FOR spalte=1 TO anz          <18D0>
8850 FOR zeile=1 TO anz           <17A7>
8860 IF zeile=spalte THEN 8890    <1CF5>
8870 IF w(spalte,zeile)<>0 THEN no
chmal=1                           <2D65>
8880 GOTO 8900                    <091F>
8890 IF w(spalte,zeile)=0 THEN noc
hmal=0                            <2D94>
8900 NEXT                         <066C>
8910 NEXT                         <0680>
8920 RETURN                       <06F8>
9000 '*****************************
                                  <23F6>
9001 '* Matrize anzeigen         *
                                  <2358>
9002 '*****************************
                                  <23FA>
9003 '                            <0784>
9010 IF anzeige=0 THEN RETURN     <1475>
9020 LOCATE 1,1                   <095D>
9030 FOR n=1 TO anz               <135D>
9040 FOR nn=1 TO anz              <14F4>
9050 IF w(n,nn)THEN PRINT CHR$(24)
;                                 <1FA1>
9060 PRINT USING"####.##";w(n,nn);
                                  <223E>
9070 IF w(n,nn)THEN PRINT CHR$(24)
;                                 <1FC9>
9080 PRINT" "CHR$(64+nn)"+";      <1925>
9090 NEXT                         <06EA>
9100 PRINT CHR$(8)"=";            <0FAE>
9110 PRINT USING" ####.#";w(n,nn) <2181>
9120 NEXT                         <0625>
9130 RETURN                       <069D>
10000 '****************************
*                                 <23CA>
10001 '* Ergebnis ausgeben
*                                 <2309>
10002 '****************************
*                                 <23CE>
10003 '                           <0758>
10010 FOR zeile=1 TO anz          <17BB>
10020 PRINT CHR$(96+zeile)"="w(zei
le,anz+1)                         <2FB7>
10030 NEXT                        <0646>
10040 RETURN                      <06BE>
50000 '****************************
*                                 <23E5>
50010 '*--------------------------
*                                 <2305>
50020 '*    FEHLERMELDUNGEN
*                                 <23D1>
50030 '*       BEARBEITEN
*                                 <2347>
50040 '*--------------------------
*                                 <2341>
50050 '****************************
*                                 <2349>
50060 '                           <07E5>
50100 IF ERR=11 THEN unmoeg=1:RETU
RN                                <1914>
50130 PRINT"Fehler in Zeile "ERL:E
RROR ERR                          <1D12>
60000 '****************************
*                                 <232C>
60010 '*--------------------------
*                                 <234C>
60020 '*    TECHNISCHER START
*                                 <232C>
60030 '*--------------------------
*                                 <2374>
60040 '****************************
*                                 <237C>
60050 '                           <0718>
60060 MODE 2:DIM w(20,21):anzeige=
1                                 <21F6>
60070 WINDOW#0,1,80,1,24:WINDOW#1,
1,80,25,25                        <2168>
60080 ON ERROR GOTO 50000         <0B09>
60100 RETURN                      <0698>
```

Dateiverwaltung mit Turbo Pascal

Blitzschnelle Adreßverwaltung

Schlüsseldateien und Freispeicher-System

Die Verwaltung großer Datenmengen stellt den Programmierer oft vor ein Problem: Wie werden die Informationen am besten gespeichert und am schnellsten wiedergefunden? Unser Autor Steffen Klaer stellt eine Lösung vor.

Turbo Pascal kennt grundsätzlich zwei Arten von Dateien: sequentielle und Random-Access-Files. Sequentielle Dateien werden häufig zur Speicherung von Texten verwendet. Der Turbo-Editor etwa legt den Quelltext eines Programms in dieser Form ab.
In Turbo Pascal deklarieren Sie eine sequentielle Datei mit

```
VAR
  /Dateivariable/ : FILE OF
  /Komponententyp/
```

oder mit

```
VAR
  /Dateivariable/ : TEXT
```

Dabei stehen die Prozeduren ASSIGN, REWRITE, RESET, READ, WRITE, CLOSE, ERASE und RENAME sowie die Funktion EOF für die Bearbeitung zur Verfügung. Die möglichen Zustände einer sequentiellen Datei zeigt das Diagramm in Abbildung 1.
Sequentielle Files haben den Nachteil, daß jeder Datensatz gelesen werden muß, bevor der nächste erreichbar ist. Wenn Sie etwa den hundertsten Eintrag erfahren wollen, müssen Sie die 99 vorangehenden der Reihe nach abarbeiten.
Häufig benötigen Sie kein besseres System; wenn aber zum Beispiel einzelne Daten gelöscht werden sollen, kommen Sie nicht umhin, die ganze Datei in ein Array zu übertragen, die unerwünschten Teile zu löschen und das Ganze anschließend wieder abzuspeichern. Eine langsamere und umständlichere Methode ist kaum denkbar. Auf dem Schneider CPC kommt hinzu, daß manche Dateien nicht ohne Probleme ins RAM übertragen werden können. Der Speicherplatz reicht dafür nicht aus.
Eine Teillösung dieses Problems stellen Random-Access-Dateien dar, also Dateien mit wahlfreiem Zugriff. Sie können in einem solchen File jede Komponente einzeln adressieren. Die Voraussetzung dafür ist, daß alle Datensätze dasselbe Aussehen haben. Bei einer Adreßverwaltung etwa besteht jeder Eintrag aus den Elementen Name, Straße, Wohnort und Telefonnummer.
Random-Access-Files werden in Pascal ebenso wie sequentielle Dateien deklariert. Der Typ TEXT ist allerdings nicht zugelassen.
Der Unterschied zwischen den beiden Datei-Arten liegt in der Anwendung einiger weiterer Standard-Prozeduren auf das File:

```
SEEK (/Dateivariable/,n)
```

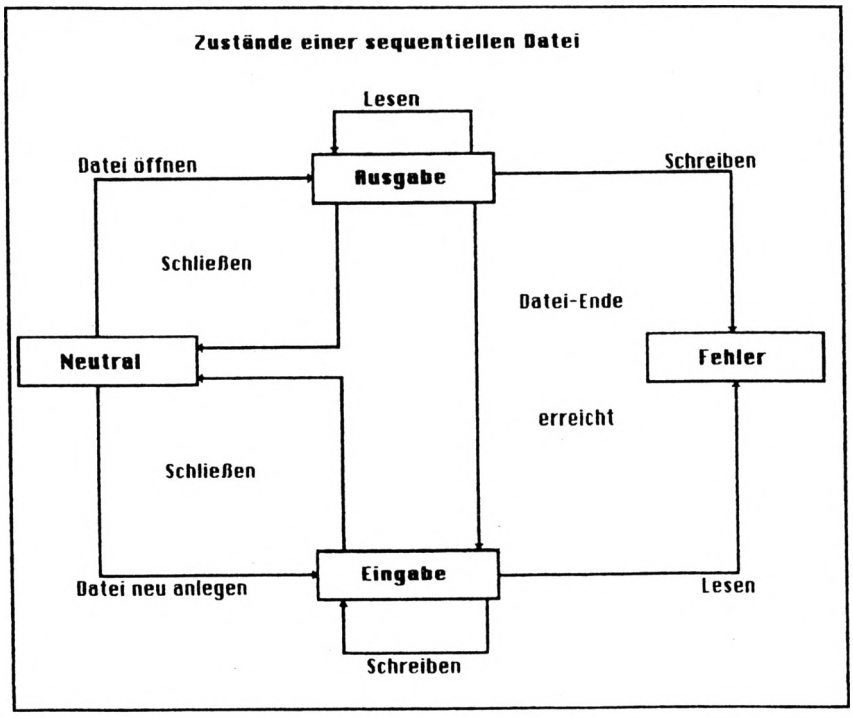
bewegt den Dateizeiger auf die n-te Komponente des Files. Die Numerierung beginnt bei null.

```
FILESIZE (/Dateivariable/)
```

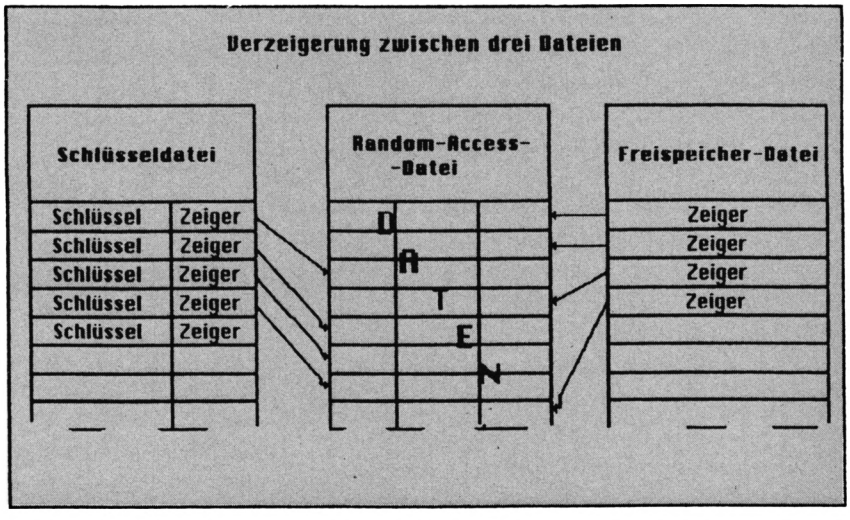
ermittelt die Anzahl der Komponenten in der Datei. Für Anwender von CP/M 2.2 ist erwähnenswert, daß diese Funktion bisweilen falsche Ergebnisse liefert. Dadurch wird der Vorteil einer Random-Access-Datei zunichte gemacht.

```
FILEPOS (/Dateivariable/)
```

gibt die momentane Position des Dateizeigers zurück.
Es ist unmöglich, an dieser Stelle auf alle Einzelheiten dieser Prozeduren einzugehen. Für das Verständnis des hier dargestellten Lösungsweges ist die Kurzbeschreibung ausreichend; wenn Sie sich eingehender mit der Materie beschäftigen wollen, so ist es empfehlenswert, das Listing der Adreßverwaltung komplett durchzuarbeiten.
Random-Access-Files sind eine feine Sache; aber dennoch ist das Problem der schnellen Suche damit nicht gelöst. Datensätze können nur anhand ihrer Nummern gefunden werden. Bei hundert Adressen können Sie sich kaum merken, an welcher Stelle etwa Frau Müller oder Herr Maier gespeichert sind.
Das wäre auch nicht der Sinn einer Dateiverwaltung. Ihr Gedächtnis soll schließlich entlastet werden. Für das Merken der Datei-Positionen ist also der Programmierer, beziehungsweise sein Programm, verantwortlich.
Eine gute Möglichkeit, große Datenmengen zu verwalten, bieten Schlüsselfiles mit Freispeicher-System. Dabei wird ein Feld mit Schlüsseln, zum Beispiel den Namen, angelegt, die mit einem Zeiger auf die zugehörigen Daten versehen werden. Diese Daten werden in einer Random-Access-Datei abgelegt.
Als System für die Verwaltung der Schlüssel bieten sich ein ARRAY OF RECORD oder eine lineare Liste mit Pointern an. Letztere Möglichkeit wurde für das Adressen-Programm gewählt. Im Record enthält eine Variable den Namen; eine zweite speichert den Zeiger auf die restlichen Daten.
Bis zu diesem Punkt ist die Methode einfach zu verstehen und anzuwenden. Was aber geschieht, wenn ein Schlüssel gelöscht werden soll? Mit dem Schlüssel geht auch der Zeiger auf den entsprechenden Datensatz verloren. Der Eintrag ist für das Programm nicht mehr zugänglich und damit gelöscht. In der Random-Access-Datei belegt er jedoch weiterhin Speicherplatz.
Sie könnten natürlich die einfachste Methode wählen und neue Datensätze nur am Ende anfügen. Dadurch erhöht sich im Laufe der Zeit jedoch die Anzahl der Datei-Leichen, die durch Löschen einzelner Datensätze entstanden sind. Ihre Datei wird immer größer, und irgendwann ist die Diskette voll.
Es wäre daher wünschenswert, die mit unzugänglichen Datensätzen belegten Positionen für neue Einträge nutzen zu können. Dies läßt sich durch eine Freispeicher-Liste realisieren. Sie enthält Zeiger, die auf die gelöschten Einträge der Random-Access-Datei weisen. Wenn neue Daten kommen, wird diese Liste überprüft. Die Positionen, auf die ein Zeiger weist, werden als erste überschrieben. Erst wenn die Liste leer ist, werden neue Daten am Ende angefügt.
Der Zusammenhang zwischen Schlüsseldatei, Freispeicher-Ver-



Das Zustandsdiagramm für ein sequentielles File



Zwei Zeiger-Listen ermöglichen schnellen Zugriff

waltung und Random-Access-Datei ist in Abbildung 2 grafisch dargestellt.
Die drei Datensysteme sind voneinander unabhängig. Daher wurden die Verwaltungsroutinen für Schlüsseldatei und Freispeicher in zwei Include-Files untergebracht. In Verbindung mit dem Deklarationsteil des Hauptprogramms eignen sie sich als Grundlage für eine Toolbox.
Auf ein Modul zur Verwaltung des Random-Access-Files wurde verzichtet. Die Verwaltung dieser Datei ist zu problemspezifisch, als daß es möglich wäre, allgemein gültige Routinen zu finden. Der Inhalt der Datensätze ist stark von der Anwendung abhängig. Unsere Adressenverwaltung verwendet die hier besprochenen Grundlagen. Die modulare Programmierung ermöglicht es Ihnen, Änderungen am Programm durchzuführen. In den Abbildungen 3 und 4 sind sämtliche Prozeduren aufgelistet, die für die Speicherverwaltung zuständig sind. Alle Routinen aus den Include-Dateien, die dort nicht genannt sind, können Sie zwar ebenfalls ändern; Sie sollten aber vorsichtig ans Werk gehen, da diese Prozeduren intern mehrfach verwendet werden. Eine Änderung an der falschen Stelle kann unvorhersehbare Ergebnisse nach sich ziehen.
Wenn Sie nur die Adreßverwaltung verwenden wollen, so tippen Sie zunächst die Listings EXIST.INC. FREE2.INC und KEY2.INC ab. Anschließend geben Sie ADR4.PAS ein und compilieren das Ganze. Obwohl eine Compilierung im Speicher möglich ist, sollten Sie ein COM-File erzeugen. Die dynamischen Pointer nehmen einigen Speicherplatz in Anspruch, sobald Sie mehrere Adressen eingegeben haben.
Durch die Menüführung ist das Programm einfach zu bedienen. Eine Erweiterung um Such- und Druck-Routinen liegt nahe und kann von Ihnen selbst realisiert werden.

In der vorliegenden Version ist die Adreßverwaltung nur unter Turbo Pascal und CP/M Plus lauffähig. Da der Autor sich weitgehend an die Standard-Notation von Pascal gehalten hat, ist eine Anpassung an andere Dialekte dieser Sprache leicht möglich. Der ASSIGN-Befehl zur Zuweisung einer Dateivariablen kann dabei eventuell entfallen. Die Prozedur CLRSCR zum Löschen des Bildschirms heißt möglicherweise anders; in HiSoft-Pascal wurde sie etwa als PAGE implementiert. Eine Anpassung an CP/M 2.2 können Sie vornehmen, indem Sie die FILESIZE-Funktion umdefinieren. Die entsprechende Prozedur muß die Datei als sequentielles File abarbeiten und die Anzahl der Datensätze, ausgehend vom ersten Record, ermitteln. Um die Kompatiblität zu wahren, müssen Sie der ersten Eintragung die Nummer Null geben. Dadurch wird die Verarbeitungsgeschwindigkeit natürlich reduziert, da Sie eine Random-Access-Datei wie ein sequentielles File behandeln.

Steffen Klaer/AE □

Prozeduren und Funktionen der Schlüssel-Verwaltung

```
ListEmpty
   stellt fest, ob die Liste leer ist

EnterListElement
   fügt ein Element alphabetisch in die Liste ein

ListKeyElement
   listet den angegebenen Schlüssel auf dem Bildschirm und liefert die Position
   der zugehörigen Daten

ReadKey
   liest die Schlüssel-Liste aus ihrer Datei

InList
   prüft, ob das angegebene Element schon vorhanden ist

DeleteListElement
   löscht ein Element aus der Liste

CreateKey
   initialisiert die Liste und die zugehörige Datei

WriteKey
   schreibt die Schlüssel-Liste in die Datei
```

Oben: Die Prozeduren der Schlüssel-Verwaltung

Unten: Die Freispeicher-Routinen des Programms

Prozeduren und Funktionen zur Verwaltung des Freispeichers

```
FreeEmpty
   prüft, ob der Freispeicher leer ist

PutElement
   legt ein Element im Freispeicher ab

GetElement
   holt ein Element vom Stapel

ReadFree
   liest den Freispeicher aus seiner Datei

CreateFree
   initialisiert den Freispeicher

WriteFree
   schreibt den Freispeicher auf Diskette
```

Listing von ADRESS.PAS

```
PROGRAM Address4 (DataFile,FreeFile,Input,KeyFile,Output);


  CONST
    DataFilename        = 'ADR.DTA';    { Filename }
    MaxEntries          = 150;
                          { Maximalzahl von Eintraegen }


  TYPE
    TypeDataFileType = Record
                         Street,
                         City     : String [20];
                         Phone    : String [15]
                       End;
    TypeKeyFileType  = Record
                         Name        : String [20];
                         DataPointer : Integer
                       End;
    TypeDataFile     = FILE OF TypeDataFileType;
    TypeFilename     = String [12];
    TypeFreeFile     = FILE OF Integer;
    TypeFreePointer  = ^FreeElement;
    FreeElement      = Record
                         FreeEntry    : Integer;
                         NextPointer,
                         BackPointer  : TypeFreePointer
                       End;
    TypeKeyFile      = FILE OF TypeKeyFileType;
    TypeKeyPointer   = ^KeyElement;
    KeyElement       = Record
                         Name         : String [20];
                         DataPointer  : Integer;
                         NextPointer,
                         BackPointer  : TypeKeyPointer
                       End;


  VAR
    CurrentPointer  : TypeFreePointer;
                      { Aktueller Pointer auf das
                        letzte Freispeicherelement }
    DataFile        : TypeDataFile;
                      { Datenfile mit den Teilnehmer-
                                      informationen }
    FreeFile        : TypeFreeFile; { Freispeicherfile }
    FreePointer     : TypeFreePointer;
                      { Erster Pointer auf den Frei-
                                          speicher }
    KeyFile         : TypeKeyFile; { Schluesselfile }
    KeyPointer      : TypeKeyPointer;
                      { Pointer auf die lineare
                                          Liste }
    NumberOfEntries : Integer; { Anzahl der Eintraege }


  {$I KEY2.INC }


  {$I FREE2.INC }


  PROCEDURE ListPerson
            (VAR DataFile    : TypeDataFile;
                 ListPointer : TypeKeyPointer;
                 Index       : Integer);

    Var
      DataPosition : Integer; { Position der Daten }
      PersonData   : TypeDataFileType; { Datenelement }

    Begin (* ListPerson *)
      ListKeyElement (DataPosition,ListPointer,Index);
      Seek (DataFile, DataPosition);
      Read (DataFile, PersonData);
      With PersonData Do
        Begin
          WriteLn (Output, '   ', Street);
          WriteLn (Output, '   ', City);
          WriteLn (Output, '   ', Phone)
        End
    End;  (* ListPerson *)


  PROCEDURE ClearSpace;

    Var
      LoopControl : Integer; { Schleifenzaehler }

    Begin (* ClearSpace *)
      For LoopControl := 8 to 12 Do
        Begin
          GotoXY (1,LoopControl);
          ClrEol
        End
    End;  (* ClearSpace *)


  PROCEDURE SetName (VAR DataFile : TypeDataFile);

    Begin (* SetNames *)
      Assign (DataFile, DataFilename)
    End;  (* SetNames *)


  {$I EXIST.INC }


  PROCEDURE Create (VAR DataFile : TypeDataFile);

    Begin (* Create *)
      Rewrite (DataFile);
      Close (DataFile)
    End;  (* Create *)


  PROCEDURE ReadInfo
            (VAR PresentPointer  : TypeFreePointer;
             VAR FreeFile        : TypeFreeFile;
             VAR StartPointer    : TypeFreePointer;
             VAR KeyFile         : TypeKeyFile;
             VAR ListPointer     : TypeKeyPointer;
             VAR NumberOfEntries : Integer);

    Begin (* ReadInfo *)
      ReadKey (KeyFile,ListPointer,NumberOfEntries);
      ReadFree (PresentPointer,FreeFile,
                StartPointer,NumberOfEntries)
    End;  (* ReadInfo *)


  PROCEDURE GetEntry
            (VAR PresentPointer  : TypeFreePointer;
             VAR DataFile        : TypeDataFile;
             VAR StartPointer    : TypeFreePointer;
             VAR ListPointer     : TypeKeyPointer;
             VAR NumberOfEntries : Integer);

    Var
      Correct      : Char;
                     { Testvariable fuer korrekte Eingabe }
      DataPosition : Integer;
      PersonData   : TypeDataFileType;
      PersonKey    : TypeKeyFileType;
                     { Schluesssel fuer einen Teilnehmer }

    Begin (* GetEntry *)
      Repeat
        ClrScr;
        LowVideo;
        Write (Output, 'PROGRAM');
        NormVideo;
        WriteLn (Output, ' Address4');
        GotoXY (1,5);
        WriteLn (Output, 'Person eingeben :');
        GotoXY (1,8);
        With PersonKey Do
          Begin
            Write (Output, 'Name    : ');
            ReadLn (Input, Name)
          End;
        WriteLn (Output);
        With PersonData Do
          Begin
            Write (Output, 'Strasse : ');
            ReadLn (Input, Street);
            Write (Output, 'Wohnort : ');
            ReadLn (Input, City);
            Write (Output, 'Telefon : ');
            ReadLn (Input, Phone)
          End;
        Repeat
          GotoXY (1,15);
          Write (Output, 'Ist die Eingabe korrekt (J/N) ? ');
          ReadLn (Input, Correct);
          Correct := UpCase (Correct)
        Until Correct IN ['J','N']
      Until Correct = 'J';
      If NumberOfEntries <= MaxEntries
        then
          Begin
            If FreeEmpty (StartPointer)
              then
```

```
185               DataPosition := FileSize (DataFile)
186             else
187               GetElement (PresentPointer,
188                           DataPosition,StartPointer);
189           EnterListElement (ListPointer,NumberOfEntries,
190                             DataPosition,PersonKey);
191           Reset (DataFile);
192           Seek (DataFile, DataPosition);
193           Write (DataFile, PersonData);
194           Close (DataFile)
195         End
196   End;  (* GetEntry *)
197
198
199 PROCEDURE KillEntry
200           (VAR PresentPointer  : TypeFreePointer;
201            VAR DataFile        : TypeDataFile;
202            VAR StartPointer    : TypeFreePointer;
203            VAR ListPointer     : TypeKeyPointer;
204            VAR NumberOfEntries : Integer);
205
206   Var
207     Correct        : Char;
208     DataPosition,
209     LoopControl    : Integer;
210     PersonData     : TypeDataFileType;
211     PersonKey      : TypeKeyFileType;
212
213   Begin (* KillEntry *)
214     Repeat
215       ClrScr;
216       LowVideo;
217       Write (Output, 'PROGRAM');
218       NormVideo;
219       WriteLn (Output, ' Address4');
220       GotoXY (1,5);
221       WriteLn (Output, 'Person loeschen :');
222       GotoXY (1,8);
223       Write (Output, 'Name : ');
224       With PersonKey Do
225         ReadLn (Input, Name);
226       Repeat
227         GotoXY (1,15);
228         Write (Output, 'Ist die Eingabe korrekt (J/N) ? ');
229         ReadLn (Input, Correct);
230         Correct := UpCase (Correct)
231       Until Correct IN ['J','N']
232     Until Correct = 'J';
233     If InList (ListPointer,PersonKey)
234       then
235         Begin
236           DeleteListElement (DataPosition,ListPointer,
237                              NumberOfEntries,PersonKey);
238           With PersonData Do
239             Begin
240               Street := '';
241               City := '';
242               Phone := ''
243             End;
244           Reset (DataFile);
245           Seek (DataFile, DataPosition);
246           Write (DataFile, PersonData);
247           Close (DataFile);
248           PutElement (PresentPointer,
249                       StartPointer,DataPosition)
250         End
251   End;  (* KillEntry *)
252
253
254 PROCEDURE ListEntries
255           (VAR DataFile        : TypeDataFile;
256                ListPointer     : TypeKeyPointer;
257                NumberOfEntries : Integer);
258
259   Var
260     LoopControl    : Integer;
261
262   Begin (* ListEntries *)
263     ClrScr;
264     LowVideo;
265     Write (Output, 'PROGRAM');
266     NormVideo;
267     WriteLn (Output, ' Address4');
268     GotoXY (1,5);
269     WriteLn (Output, 'Personen auflisten :');
270     For LoopControl := 0 to NumberOfEntries - 1 Do
271       Begin
272         GotoXY (1,8);
273         Reset (DataFile);
274         ListPerson (DataFile,ListPointer,LoopControl);
275         Delay (1000);
276         ClearSpace
277       End;
278     Close (DataFile);
279     GotoXY (1,23);
280     WriteLn (Output, 'Press any key to continue');
281     Repeat
282     Until KeyPressed
283   End;  (* ListEntries *)
284
285
286 PROCEDURE Initialize
287           (VAR PresentPointer  : TypeFreePointer;
288            VAR DataFile        : TypeDataFile;
289            VAR FreeFile        : TypeFreeFile;
290            VAR StartPointer    : TypeFreePointer;
291            VAR KeyFile         : TypeKeyFile;
292            VAR ListPointer     : TypeKeyPointer;
293            VAR NumberOfEntries : Integer);
294
295   Const
296     KeyFilename = 'KEY.DTA'; { Filename }
297
298   Begin (* Initialize *)
299     ClrScr;
300     WriteLn (Output, 'Reading..');
301     SetName (DataFile);
302     CreateKey (KeyFile,ListPointer,NumberOfEntries);
303     CreateFree (FreeFile,StartPointer);
304     If NOT FileExist (KeyFilename)
305       then
306         Create (DataFile)
307       else
308         ReadInfo (PresentPointer,FreeFile,
309                   StartPointer,KeyFile,
310                   ListPointer,NumberOfEntries)
311   End;  (* Initialize *)
312
313
314 PROCEDURE Menu
315           (VAR PresentPointer  : TypeFreePointer;
316            VAR DataFile        : TypeDataFile;
317            VAR StartPointer    : TypeFreePointer;
318            VAR ListPointer     : TypeKeyPointer;
319            VAR NumberOfEntries : Integer);
320
321   Var
322     Choice : Char; { Auswahl aus dem Menue }
323
324   Begin (* Menu *);
325     Repeat
326       ClrScr;
327       LowVideo;
328       Write (Output, 'PROGRAM');
329       NormVideo;
330       WriteLn (Output, ' Address4');
331       GotoXY (1,5);
332       WriteLn (Output, 'Menu :');
333       GotoXY (1,8);
334       WriteLn (Output, '1. Person eingeben');
335       GotoXY (1,10);
336       WriteLn (Output, '2. Person loeschen');
337       GotoXY (1,12);
338       WriteLn (Output, '3. Personen auflisten');
339       GotoXY (1,14);
340       WriteLn (Output, '4. Quit');
341       Repeat
342         GotoXY (1,17);
343         Write (Output, 'Bitte waehlen Sie (1..4) : ');
344         Read (Kbd, Choice)
345       Until Choice IN ['1'..'4'];
346       CASE Choice OF
347           '1' : GetEntry (PresentPointer,DataFile,
348                           StartPointer,ListPointer,
349                           NumberOfEntries);
350           '2' : KillEntry (PresentPointer,DataFile,
351                            StartPointer,ListPointer,
352                            NumberOfEntries);
353           '3' : ListEntries (DataFile,ListPointer,
354                              NumberOfEntries)
355       End
356     Until Choice = '4'
357   End;  (* Menu *)
358
359
360 PROCEDURE FinishUp
361           (VAR PresentPointer  : TypeFreePointer;
362            VAR FreeFile        : TypeFreeFile;
363            VAR StartPointer    : TypeFreePointer;
364            VAR KeyFile         : TypeKeyFile;
365            VAR ListPointer     : TypeKeyPointer;
366                NumberOfEntries : Integer);
367
368   Begin (* FinishUp *)
369     ClrScr;
370     WriteLn (Output, 'Writing..');
371     WriteKey (KeyFile,ListPointer);
372     WriteFree (PresentPointer,FreeFile,
```

```
373                 StartPointer,NumberOfEntries)
374    End;  (* FinishUp *)
375
376
377
378  BEGIN (* Address4 *)
379    Initialize (CurrentPointer,DataFile,
380                FreeFile,FreePointer,KeyFile,
381                KeyPointer,NumberOfEntries);
382    Menu (CurrentPointer,DataFile,FreePointer,
383          KeyPointer,NumberOfEntries);
384    FinishUp (CurrentPointer,FreeFile,
385              FreePointer,KeyFile,
386              KeyPointer,NumberOfEntries)
387  END.  (* Address4 *)
388
```

Listing von KEY2.INC

```
 1  PROCEDURE FillSegment
 2            ( SegmentPointer : TypeKeyPointer;
 3              PersonKey      : TypeKeyFileType);
 4
 5    Begin (* FillSegment *)
 6      With SegmentPointer^ Do
 7        Begin
 8          Name := PersonKey.Name;
 9          DataPointer := PersonKey.DataPointer;
10          NextPointer := Nil;
11          BackPointer := Nil
12        End;
13    End;  (* FillSegment *)
14
15
16  PROCEDURE AddAtEnd
17            ( Position    : Integer;
18              ListPointer : TypeKeyPointer;
19              PersonKey   : TypeKeyFileType);
20
21    Var
22      Temp1,
23      Temp2  : TypeKeyPointer; { Temporaere Pointer }
24
25    Begin (* AddAtEnd *)
26      Temp1 := ListPointer;
27      While Temp1^.NextPointer <> Nil Do
28        With Temp1^ Do
29          Temp1 := NextPointer;
30      New (Temp2);
31      FillSegment (Temp2,PersonKey);
32      With Temp2^ Do
33        BackPointer := Temp1;
34      With Temp1^ Do
35        NextPointer := Temp2;
36      Temp1 := Nil;
37      Temp2 := Nil
38    End;  (* AddAtEnd *)
39
40
41  FUNCTION ListEmpty
42           ( TestPointer : TypeKeyPointer) :
43                                    Boolean;
44
45    Begin (* ListEmpty *)
46      If TestPointer = Nil
47        then
48          ListEmpty := True
49        else
50          ListEmpty := False
51    End;  (* ListEmpty *)
52
53
54  PROCEDURE InsertIntoList
55            (VAR ListPointer : TypeKeyPointer;
56                 Position    : Integer;
57                 PersonKey   : TypeKeyFileType);
58
59    Var
60      Temp1,
61      Temp2  : TypeKeyPointer;
62
63    Begin (* InsertIntoList *)
64      If ListPointer^.Name >= PersonKey.Name
65        then
66          Begin
67            New (Temp1);
68            FillSegment (Temp1,PersonKey);
69            With Temp1^ Do
70              Begin
71                NextPointer := ListPointer;
72                NextPointer^.BackPointer := Temp1
73              End;
74            ListPointer := Temp1
75          End
76        else
77          Begin
78            Temp1 := ListPointer;
79            While ((Temp1^.Name <= PersonKey.Name)
80                  AND (Temp1 <> Nil)) Do
81              With Temp1^ Do
82                Temp1 := NextPointer;
83              If Temp1 <> Nil
84                then
85                  Begin
86                    New (Temp2);
87                    FillSegment (Temp2,PersonKey);
88                    With Temp2^ Do
89                      Begin
90                        BackPointer := Temp1^.BackPointer;
91                        NextPointer := Temp1
92                      End;
93                    With Temp1^ Do
94                      BackPointer := Temp2;
95                    With Temp2^.BackPointer^ Do
96                      NextPointer := Temp2;
97                    Temp1 := Nil;
98                    Temp2 := Nil
99                  End
100               else
101                 AddAtEnd
102                   (Position,ListPointer,PersonKey)
103         End
104   End;  (* InsertIntoList *)
105
106
107 PROCEDURE EnterListElement
108           (VAR ListPointer     : TypeKeyPointer;
109            VAR NumberOfEntries : Integer;
110                Position        : Integer;
111                PersonKey       : TypeKeyFileType);
112
113   Begin (* EnterListElement *)
114     With PersonKey Do
115       DataPointer := Position;
116     If ListEmpty (ListPointer)
117       then
118         Begin
119           New (ListPointer);
120           FillSegment (ListPointer,PersonKey);
121         End
122       else
123         InsertIntoList
124               (ListPointer,Position,PersonKey);
125     NumberOfEntries := NumberOfEntries + 1
126   End;  (* EnterListElement *)
127
128
129 PROCEDURE ListKeyElement
130           (VAR Position    : Integer;
131                ListPointer : TypeKeyPointer;
132                Index       : Integer);
133
134   Var
135     LoopControl : Integer;        { Schleifenzaehler }
136     Temp        : TypeKeyPointer;{ Temporaerer Pointer }
137
138   Begin (* ListKeyElement *)
139     Temp := ListPointer;
140     For LoopControl := 0 to Index - 1 Do
141       With Temp^ Do
142         Temp := NextPointer;
143     With Temp^ Do
144       Begin
145         WriteLn (Output, Index + 1, '. ', Name);
146         Position := DataPointer
147       End;
148     Temp := Nil
149   End;  (* ListKeyElement *)
150
151
152 PROCEDURE ReadKey
153           (VAR KeyFile         : TypeKeyFile;
154            VAR ListPointer     : TypeKeyPointer;
155            VAR NumberOfEntries : Integer);
156
157   Var
158     DataPosition : Integer; { Position der Daten }
159     FileRecord   : TypeKeyFileType;
160                    { Element der Schluesseldatei }
161
162   Begin (* ReadKey *)
163     Reset (KeyFile);
164     While NOT EOF (KeyFile) Do
165       Begin
166         Read (KeyFile, FileRecord);
```

```
167           With FileRecord Do
168             DataPosition := DataPointer;
169           EnterListElement (ListPointer,NumberOfEntries,
170                             DataPosition,FileRecord)
171         End;
172       Close (KeyFile)
173     End;  (* ReadKey *)
174
175
176   FUNCTION InList
177           (ListPointer : TypeKeyPointer;
178            PersonKey   : TypeKeyFileType) :
179                                 Boolean;
180
181     Var
182       Temp : TypeKeyPointer;
183
184     Begin (* InList *)
185       Temp := ListPointer;
186       While ((Temp^.Name <> PersonKey.Name)
187             AND (Temp^.NextPointer <> Nil)) Do
188         With Temp^ Do
189           Temp := NextPointer;
190       With Temp^ Do
191         If ((NextPointer <> Nil)
192             OR (Name = PersonKey.Name))
193           then
194             InList := True
195           else
196             InList := False
197     End;  (* InList *)
198
199
200   PROCEDURE DeleteListElement
201             (VAR Position         : Integer;
202              VAR ListPointer      : TypeKeyPointer;
203              VAR NumberOfEntries : Integer;
204                  PersonKey        : TypeKeyFileType);
205
206     Var
207       Temp : TypeKeyPointer;
208
209     Begin (* DeleteListElement *)
210       Temp := ListPointer;
211       While Temp^.Name <> PersonKey.Name Do
212         With Temp^ Do
213           Temp := NextPointer;
214       With Temp^ Do
215         Begin
216           Position := DataPointer;
217           If BackPointer = Nil
218             then
219               Begin
220                 NextPointer^.BackPointer := Nil;
221                 ListPointer := NextPointer;
222                 NextPointer := Nil
223               End
224             else
225               If NextPointer <> Nil
226                 then
227                   Begin
228                     NextPointer^.BackPointer
229                       := BackPointer;
230                     BackPointer^.NextPointer
231                       := NextPointer;
232                     NextPointer := Nil;
233                     BackPointer := Nil
234                   End
235                 else
236                   Begin
237                     BackPointer^.NextPointer
238                       := Nil;
239                     BackPointer := Nil
240                   End
241         End;
242       Dispose (Temp);
243       NumberOfEntries := NumberOfEntries - 1
244     End;  (* DeleteListElement *)
245
246
247   PROCEDURE CreateKey
248             (VAR KeyFile          : TypeKeyFile;
249              VAR ListPointer      : TypeKeyPointer;
250              VAR NumberOfEntries : Integer);
251
252     Const
253       KeyFilename = 'KEY.DTA'; { Name der Schluesseldatei }
254
255     Begin (* CreateKey *)
256       NumberOfEntries := 0;
257       ListPointer := Nil;
258       Assign (KeyFile, KeyFilename)
259     End;  (* CreateKey *)
```

```
260
261
262   PROCEDURE WriteKey
263             (VAR KeyFile      : TypeKeyFile;
264              VAR ListPointer : TypeKeyPointer);
265
266     Var
267       FileRecord : TypeKeyFileType;
268       Temp       : TypeKeyPointer;
269
270     Begin (* WriteKey *)
271       If NOT ListEmpty (ListPointer)
272         then
273           Begin
274             Rewrite (KeyFile);
275             With ListPointer^ Do
276               Temp := NextPointer;
277             While Temp <> Nil Do
278               Begin
279                 With ListPointer^ Do
280                   Begin
281                     FileRecord.Name := Name;
282                     FileRecord.DataPointer
283                       := DataPointer
284                   End;
285                 With Temp^ Do
286                   BackPointer := Nil;
287                 With ListPointer^ Do
288                   NextPointer := Nil;
289                 Dispose (ListPointer);
290                 ListPointer := Temp;
291                 Write (KeyFile, FileRecord);
292                 With ListPointer^ Do
293                   Temp := NextPointer
294               End;
295             With ListPointer^ Do
296               Begin
297                 FileRecord.Name := Name;
298                 FileRecord.DataPointer
299                   := DataPointer
300               End;
301             Write (KeyFile, FileRecord);
302             Dispose (ListPointer);
303             Close (KeyFile)
304           End
305     End;  (* WriteKey *)
306
```

Listing von FREE2.INC

```
1   FUNCTION FreeEmpty
2           (TestPointer : TypeFreePointer) :
3                                 Boolean;
4
5     Begin (* FreeEmpty *)
6       If TestPointer = Nil
7         then
8           FreeEmpty := True
9         else
10          FreeEmpty := False
11    End;  (* FreeEmpty *)
12
13
14  PROCEDURE ReverseFree
15            (VAR PresentPointer,
16                 StartPointer    : TypeFreePointer);
17
18    Var
19      FirstPointer,
20                  { Anfangspointer auf den   }
21                  { temporaeren Freispeicher }
22      TempPointer, { Temporaerer Pointer }
23      WorkPointer   : TypeFreePointer;
24                  { Aktueller Pointer auf das
25                    letzte Element des
26                    temporaeren Freispeichers }
27
28    Begin (* ReverseFree *)
29      If NOT FreeEmpty (StartPointer)
30        then
31          Begin
32            New (FirstPointer);
33            With FirstPointer^ Do
34              Begin
35                FreeEntry := PresentPointer^.FreeEntry;
36                NextPointer := Nil;
37                BackPointer := Nil
38              End;
39            WorkPointer := FirstPointer;
40            With PresentPointer^ Do
41              PresentPointer := BackPointer;
42            While NOT FreeEmpty (PresentPointer) Do
43              Begin
```

```
                New (TempPointer);
                With TempPointer^ Do
                  Begin
                    FreeEntry :=
                        PresentPointer^.FreeEntry;
                    NextPointer := Nil
                  End;
                With WorkPointer^ Do
                  NextPointer := TempPointer;
                With TempPointer^ Do
                  BackPointer := WorkPointer;
                WorkPointer := TempPointer;
                TempPointer := Nil;
                With PresentPointer^ Do
                  Begin
                    Dispose (NextPointer);
                    PresentPointer := BackPointer
                  End
              End;
            TempPointer := StartPointer;
            StartPointer := FirstPointer;
            Dispose (TempPointer);
            PresentPointer := WorkPointer;
            FirstPointer := Nil;
            WorkPointer := Nil
          End
    End;  (* ReverseFree *)


PROCEDURE PutElement
          (VAR PresentPointer : TypeFreePointer;
           VAR StartPointer   : TypeFreePointer;
               Value          : Integer);

    Var
      TempPointer : TypeFreePointer;

    Begin (* PutElement *)
      If FreeEmpty (StartPointer)
        then
          Begin
            New (StartPointer);
            With StartPointer^ Do
              Begin
                FreeEntry := Value;
                NextPointer := Nil;
                BackPointer := Nil
              End;
            PresentPointer := StartPointer;
          End
        else
          Begin
            New (TempPointer);
            With TempPointer^ Do
              Begin
                FreeEntry := Value;
                NextPointer := Nil
              End;
            With PresentPointer^ Do
              NextPointer := TempPointer;
            With TempPointer^ Do
              BackPointer := PresentPointer;
            PresentPointer := TempPointer;
            TempPointer := Nil
          End
    End;  (* PutElement *)


PROCEDURE GetElement
          (VAR PresentPointer  : TypeFreePointer;
           VAR Value           : Integer;
           VAR StartPointer    : TypeFreePointer);

    Begin (* GetElement *)
      ReverseFree (PresentPointer,StartPointer);
      With PresentPointer^ Do
        Begin
          Value := FreeEntry;
          PresentPointer := BackPointer
        End;
      If FreeEmpty (PresentPointer)
        then
          Begin
            Dispose (StartPointer);
            StartPointer := Nil
          End
        else
          Begin
            With PresentPointer^ Do
              Dispose (NextPointer);
            ReverseFree (PresentPointer,StartPointer)
          End
    End;  (* GetElement *)


PROCEDURE ReadFree
          (VAR PresentPointer  : TypeFreePointer;
           VAR FreeFile        : TypeFreeFile;
           VAR StartPointer    : TypeFreePointer;
           VAR NumberOfEntries : Integer);

    Var
      FileElement : Integer;
      { Element des Freispeichers }

    Begin (* ReadFree *)
      Reset (FreeFile);
      While NOT EOF (FreeFile) Do
        Begin
          Read (FreeFile, FileElement);
          PutElement
            (PresentPointer,StartPointer,FileElement)
        End;
      Close (FreeFile);
      GetElement
        (PresentPointer,FileElement,StartPointer);
      NumberOfEntries := FileElement
    End;  (* ReadFree *)


PROCEDURE CreateFree
          (VAR FreeFile     : TypeFreeFile;
           VAR StartPointer : TypeFreePointer);

    Const
      FreeFilename = 'FREE.DTA';
      { Filename des Freispeichers }

    Begin (* CreateFree *)
      StartPointer := Nil;
      Assign (FreeFile, FreeFilename)
    End;  (* CreateFree *)


PROCEDURE WriteFree
          (VAR PresentPointer  : TypeFreePointer;
           VAR FreeFile        : TypeFreeFile;
           VAR StartPointer    : TypeFreePointer;
               NumberOfEntries : Integer);

    Var
      FileElement : Integer;

    Begin (* WriteFree *)
      ReverseFree (PresentPointer,StartPointer);
      PutElement
        (PresentPointer,StartPointer,NumberOfEntries);
      ReverseFree (PresentPointer,StartPointer);
      Rewrite (FreeFile);
      While NOT FreeEmpty (PresentPointer) Do
        Begin
          GetElement
            (PresentPointer,FileElement,StartPointer);
          Write (FreeFile, FileElement)
        End;
      Close (FreeFile)
    End;  (* WriteFree *)
```

Listing von EXIST.INC

```
FUNCTION FileExist
         (NewFile : TypeFilename) :
                          Boolean;

    Var
      TestFile : File;

    Begin (* FileExist *)
      Assign (TestFile, NewFile);
      {$I-}
      Reset (TestFile);
      Close (TestFile);
      {$I+}
      If IOResult = 0
        then
          FileExist := True
        else
          FileExist := False
    End;  (* FileExist *)
```

2500 PROGRAMMIERSPRACHEN ZUR AUSWAHL

Wie in Babylon

Der vielschichtige, schnell wachsende Markt der Programmiersprachen und ihrer Dialekte ist auch für Insider kaum noch überschaubar. Das komplexe Thema ist nicht nur für den Profi-Programmierer, sondern auch für jene geeignet, die sich einen Überblick dieser babylonischen Sprachenverwirrung verschaffen wollen. Wir beenden unsere Serie mit neunzehn weiteren Programmiersprachen – von OCCAM bis V.I.P.

49. OCCAM

Algorithmische Sprache.
OCCAM entstand als Ergänzung zu dem Transputer, einem Microprozessor, der für Computer mit mehreren parallel arbeitenden CPUs (Zentraleinheiten) bestimmt ist.
Im Gegensatz zu Pascal ist die Möglichkeit, eine Aufgabe in parallellaufende Prozesse zu untergliedern, in OCCAM bereits integriert. Pascal mußte hierzu erst erweitert werden (Concurrent Pascal).
OCCAM ist auch für neuere Generationen von Transputern geeignet. Während bei den Transputern der ersten Entwicklungsphasen, mit herkömmlichen Prozessoren bestückt, ein Teil der Rechenleistung der Prozessoren zur Organisation und Kommunikation zwischen diesen gebraucht wurde, kommen bei den Weiterentwicklungen spezielle Prozessoren zum Einsatz, die hierfür besser geeignet sind.
Hundert herkömmliche CPUs konnten sinngemäß niemals die hundertfache Leistung einer einzelnen erbringen, trotz der Programmiersprache OCCAM, die beide Systemkonzepte gleichermaßen unterstützt. Die volle Leistungsfähigkeit kann OCCAM bei den neuen Systemen unter Beweis stellen.
Haupteinsatzgebiet von OCCAM (und auch das der Transputer) sind technisch-wissenschaftliche Applikationen wie die Konstruktion von Filterschaltungen oder schnelle Kurvenberechnung (Fast Fourier Transformation). Auch Echtzeitgrafiken, welche naturgemäß sehr rechenintensiv sind, können hiermit realisiert werden.

50. Pascal

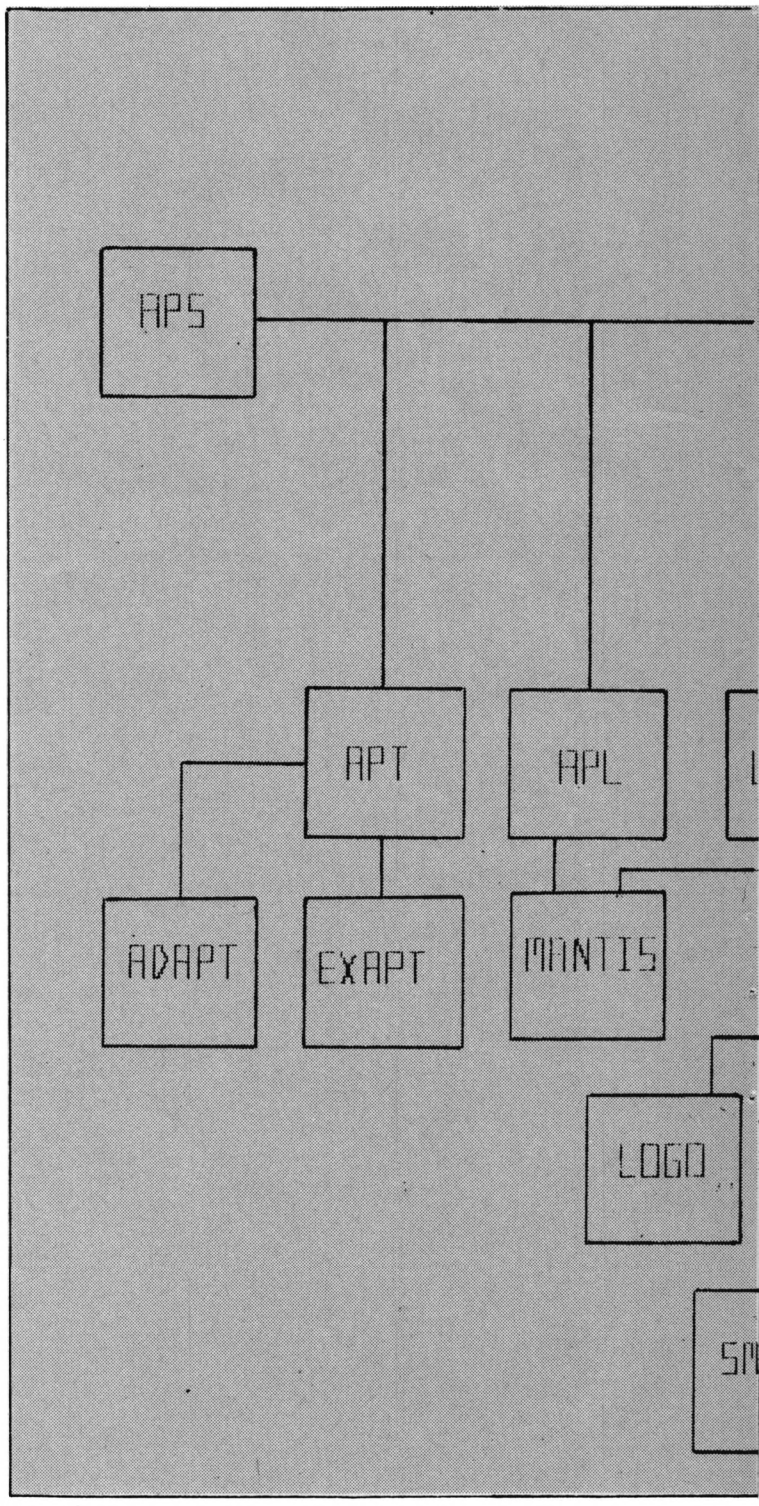
Benannt nach dem Wissenschaftler Blaise Pascal (1623 – 1663).
Pascal ist eine starke logisch strukturierte Programmiersprache, die hervorragende Hilfestellungen beim Programmieren umfangreicherer Probleme gibt.
Auf dem Markt ist Pascal seit etwa 1969.
Als Nachteil wäre zu nennen, daß Pascal für das Erstellen kleinerer Programme zu aufwendig ist.
Pascal ist eine der wenigen Sprachen, die Mengen (aus der Mengelehre) sinnvoll direkt verarbeiten kann, ohne den Programmaufwand wesentlich zu erhöhen.
Es gibt zwischenzeitlich mehrere Versionen von Pascal:

- Pascal (Jensen/Wirth-Standard),
- Pascal/Z,
- Pascal/MT+,
- UCSD-Pascal *(siehe unten)*,
- TINY Pascal,
- Concurrent Pascal (für Parallelverarbeitung),
- Pascal/M*,
- Pascal-XT (Extended Pascal),
- Pascal-SC (Pascal for Scientific Computation),
- TURBO-Pascal
- und andere.

Letztere Version dürfte wegen ihrer ausgezeichneten Verarbeitungsgeschwindigkeit die Verbreitung von Pascal wesentlich beeinflußt haben.
(Programmbeispiel siehe Tabelle 15.)

UCSD-Pascal System

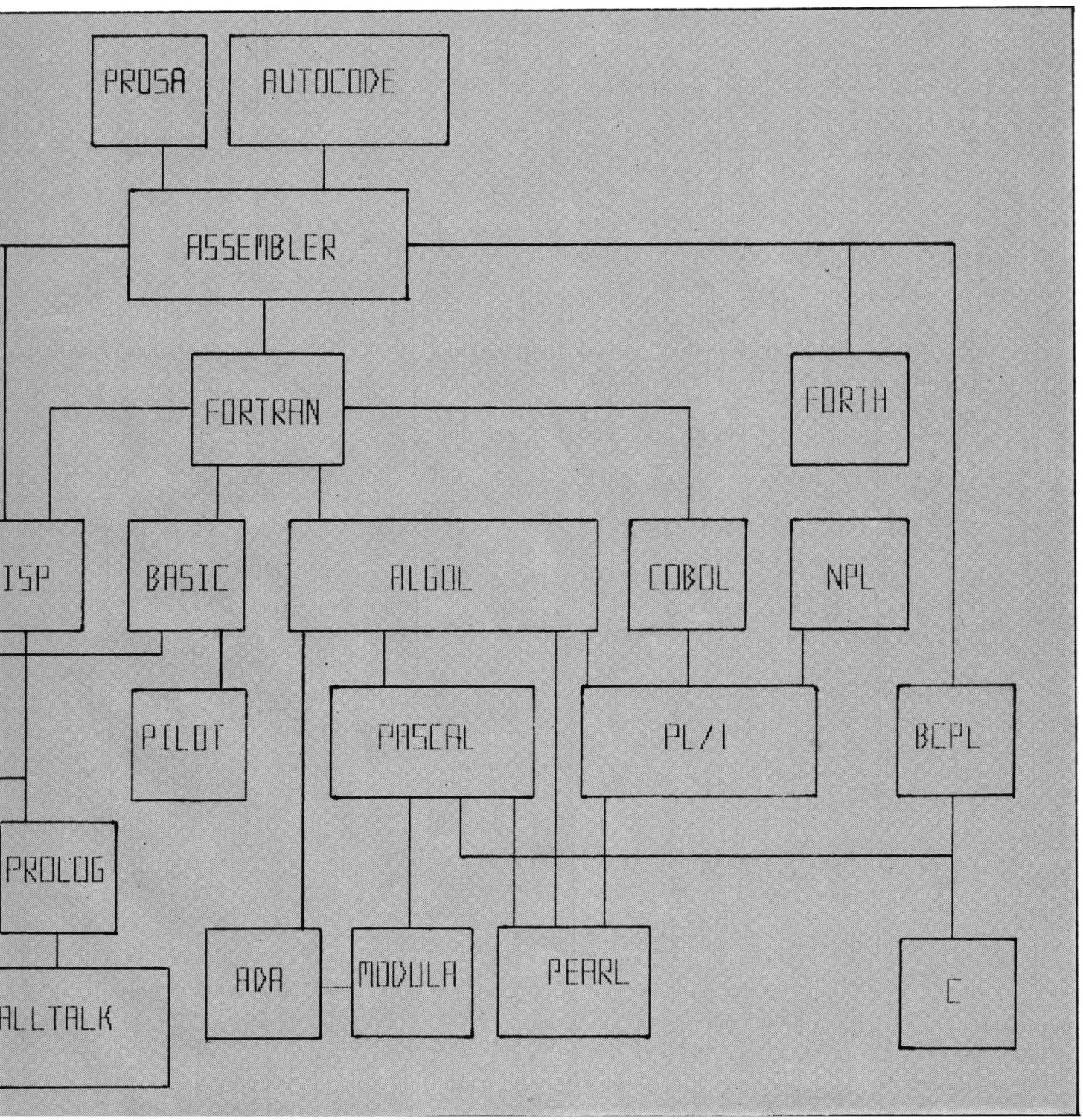
Das UCSD-Pascal wurde an der University of California in San Diego entwickelt, daher auch sein



Name. Mittlerweile gibt es UCSD-Pascal für fast alle Rechnertypen, so daß die Software in Pascal sehr portabel ist.
Der Standard des UCSD-Pascal wird hier einmal anhand seines Betriebssystems dargestellt, da es sich um mehr als einen einfachen Interpreter oder Compiler handelt.
UCSD-Pascal besteht aus mehreren Files (auf Diskette). Die Aufteilung in mehrere Files ist wegen des Umfangs einzelner Leistungsabschnitte notwendig. Bei Bedarf wird auf diese Files zugegriffen, daher müssen sie im Falle des Aufrufs durch den Anwender verfügbar sein; die Diskette muß sich im Laufwerk befinden. So ist beispielsweise das File mit dem Namen SYSTEM.APPLE notwendig, um das System zu booten (hier im Beispiel auf einem Apple-Computer).
Alle Systemfiles beginnen mit dem Namen SYSTEM:

SYSTEM.PASCAL
SYSTEM.CHARSET
SYSTEM.APPLE
SYSTEM.EDITOR
SYSTEM.COMPILER
SYSTEM.LIBRARY
und so weiter.

Die Namen lassen oft schon ahnen, welche Files sie bezeichnen. Dem Anwender ist es nicht möglich, Files anzulegen, die ebenfalls mit SYSTEM beginnen, damit sind die systemeigenen Files deutlich von den Anwenderfiles abgegrenzt.



Die Struktur des UCDS-Systems ist klar gegliedert in Systemeinheiten wie Editor, Compiler, Interpreter, Linker, Assembler und Filer. Die Systemkomponenten müssen also je nach Bedarf in den Rechner geladen werden. Zum Programmieren muß zunächst ein File erstellt werden, in welches mit Hilfe des Editors das Programm geschrieben werden kann. Zum Übersetzen muß dann der Compiler geladen werden.
Die Arbeit mit dem System findet also auf verschiedenen Ebenen statt. Das UCSD-System kennt drei verschiedene Filetypen zur Diskettenorganisation:

Text-Files,
Data-Filex,
Code-Files.

Der entsprechende Filetyp muß bei der Namensgebung des Files mitangegeben werden, beispielsweise NAME.TEXT oder INHALT.DATA. Textfiles sind solche Dateien, die mit dem Editor beschrieben werden können (etwa Programme in Pascal). Dieses File kann abgedruckt (Programmlisting) oder compiliert werden. Der Compiler bearbeitet den Inhalt des angegebenen Textfiles und speichert das Ergebnis der Übersetzung in einem Codefile ab. Dieses enthält also den ausführbaren Code. Datenfiles enthalten, wie ihr Name schon sagt, Daten, welche in einem Programm abgespeichert oder von diesem gelesen werden können.
Das Workfile ist das Systemfile (Text oder Code), auf das immer automatisch zugegriffen wird. Der Editor oder der Compiler öffnet – sofern nichts anderes angegeben wird – immer ein Input-File und ein Output-File. Befindet sich kein Workfile auf der Diskette und wurde auch kein anderes File angegeben, so erscheint eine Fehlermeldung.

Tabelle 15:

```
-------------------------------------------------------------
Programmbeispiel zu PASCAL
Listing
-------------------------------------------------------------
PROGRAMM SCHEIBEN (INPUT, OUTPUT);
VAR I: INTEGER
    LINKS, MITTE, RECHTS: CHAR;

PROCEDURE VERLAENGERE (I:INTEGER; LINKS, RECHTS, MITTE: CHAR);
BEGIN
   IF I <> 0 THEN
   BEGIN
      VERLAENGERE (I-1, LINKS, MITTE, RECHTS);
      WRITELN ('VON', LINKS, 'NACH', RECHTS);
      VERLAENGERE (I-1, MITTE, RECHTS, LINKS);
   END;
END;

BEGIN                              (* HAUPTPROGRAMM *)
   I:= 3                           (* = ANZAHL DER SCHEIBEN *)
   LINKS:= 'A';
   MITTE:= 'B';
   RECHTS:= 'C';
   VERLAENGERE (I, LINKS, RECHTS, MITTE);
END.
-------------------------------------------------------------
```

Tabelle 16:

```
-------------------------------------------------------------------------
Beispiel zu PEARL Echtzeitsprache
Listing
-------------------------------------------------------------------------
AFTER 5 SEC ALL 10 SEC              Der RechenprozeB RELAIS wird nach 5
DURING 100 MIN ACTIVATE             Sekunden, während 100 Minuten, im  7
RELAIS PRIORITZ 1                   Sekundenzyklus, mit Priorität 1 belegt.
-------------------------------------------------------------------------
```

Im Directory-Beispiel sind zwei solcher Workfiles:

SYSTEM.WRK.TEXT
und
SYSTEM.WRK.CODE

Das SYSTEM.WRK.TEXT-File dient dem Editor zugleich als Ein- und Ausgabefile, denn der Editor bearbeitet nur Texte, während der Compiler Texte in Codes umformt. Daher benutzt dieser zwar das gleiche Eingabefile, braucht aber zur Ausgabe ein Code-File: SYSTEM.WRK.CODE.

Eine Diskette darf immer nur zwei Workfiles enthalten (Text und Code).
Nach dem Booten (Laden) des USCD-Systems befindet sich der Rechner in der Kommandoebene, von der aus Direktbefehle gegeben und Dienstprogramme gestartet werden können. Der Rechner erwartet einen Befehl:

COMMAND: Edit, Run, File, Comp, Link, Xecute, Assem, Debug.

Weitere Befehlsmöglichkeiten erfährt man beim Drücken auf die Taste „?“. Es erscheint:

COMMAND: User restart, Initialize, Halt, Swap, Make exec.

Diese Befehle sind sogenannte Direktbefehle, also Teil des Betriebssystems, während die Befehle der oberen Zeile Aufrufe für Dienstprogramme sind. Diese müssen erst von der Diskette nachgeladen werden.
Anders als in BASIC bewirkt Run hier nicht lediglich einen Programmstart. Run ist selbst ein Dienstprogramm, welches den Compiler aufruft und startet. Dieser compiliert daraufhin das Workfile. Der Linker wird bei Bedarf automatisch dazugeladen. Das so übersetzte und gebundene Programm wird anschließend noch gestartet.
Run wird also nur für noch nicht compilierte Programme benötigt. Der Befehl, welcher dem BASIC-Run entspricht, wäre hier der „eXecute“-Befehl.
Run erleichtert das Austesten von Programmen, da bei der Fehlersuche nicht jedesmal der Compiler von Hand aufgerufen werden muß.
Neben dieser Möglichkeit kann der Pascal-Compiler auch direkt aufgerufen werden.
Der Compiler übersetzt den Inhalt des Workfiles (Text) in direkt ausführbaren Code und speichert diesen in dem Workfile SYSTEM.WRK.CODE ab. Ist eines der beiden Work-

files auf der Diskette nicht vorhanden, oder sollen andere Files benutzt werden, so kann dies optional angegeben werden. Erst compilierte Programme können mit „eXecute" gestartet werden.
Der Pascal-Linker ermöglicht die Einbindung von Assembler-Routinen in ein Pascal-Programm. Diese Routine muß im Pascalprogramm als „external" deklariert sein. External veranlaßt das System, vor Ausführung des Programmes die einzubindende Routine aufzusuchen und zu linken. Der Compiler meldet dies am Bildschirm. Beim Run-Befehl wird der Linker automatisch aufgerufen.
UCSD-Pascal erlaubt auch eine Art Batch-Betrieb: Ein Exec-File, welches mit „M" erzeugt wird, kann Systembefehle enthalten, die bei Start des Exec-Files ausgeführt werden. Dabei ist es gleichgültig, ob es sich um Direktbefehle oder Dienstprogramme handelt. Wie bei manchen anderen Betriebssystemen kann somit ein Batchfile erstellt werden, ein File, das Stapelverarbeitung zuläßt (Batch = Stapel).
Der Name rührt noch aus der Lochkartenzeit. Man hatte die Möglichkeit, Lochkarten mit verschiedenen Direktbefehlen aufeinanderzustapeln und diese gebündelt dem Computer einzugeben. Die Bearbeitung der Befehle erfolgt in der vorgegebenen Reihenfolge. Sprünge oder Verzweigungen sind nicht möglich.
Das UCSD-Pascal-System hat in seiner Struktur große Ähnlichkeit zu verbreiteten Betriebssystemen größerer Rechner. Ein markanterer Vorteil ist aber die Standardisierung für verschiedene Rechner und die Verbindung von Sprache und System zu einer Einheit. Erst dadurch ist es möglich, echte portable Software zu entwickeln. Zwar gibt es inzwischen viele Compiler für Pascal, aber das UCSD-System konnte sich behaupten.

51. Pearl

Process and **E**xperiment **A**utomation **R**ealtime **L**anguage.
Wie der Name dieser Programmiersprache schon sagt, ermöglicht sie das Realtime Processing, also die Echtzeit-Verarbeitung: Das Programm läuft in der gleichen Zeit wie ein Prozeß. Der Rechner arbeitet also parallel zu einem technischen Ablauf. Die Daten, die aus diesem technischen Ablauf (Prozeß) gewonnen

ECHTZEIT-VERARBEITUNG

werden, werden simultan verarbeitet. Aufgrund dieser Daten kann der Rechner in den Prozeß in Form einer Korrektur des Prozeß-Ablaufs eingreifen, oder für die Fortführung des Prozesses wichtige Entscheidungen treffen (Prozeßdatenverarbeitung).
Bei der Konzeption von Pearl wurde besonderer Wert darauf gelegt, daß Echtzeit-Sprachelemente in den Sprachkern von Pearl aufgenommen wurden – anders als bei den meisten übrigen höheren Programmiersprachen, die über solche spezifischen expliziten Sprachmittel nicht verfügen. Meist werden dort beispielsweise in Unterprogrammaufrufen Funktionen des Betriebssystems organisiert.
(Programmbeispiel siehe Tabelle 16.)
Die Zeichenketten-Verarbeitung in Pearl ist unzureichend, obwohl diese Sprache fast ausschließlich auf digitalen Großrechnern zum Einsatz kommt.
Hingegen sind die Strukturen für die Ein- und Ausgabe, sowie die Sprachmittel für Algorithmik ähnlich denen anderer höherer Programmiersprachen. Einflüsse von Algol 68 und Pascal sind augenfällig. Die Syntax ist weitgehendst an PL/1 angepaßt.
Versuche der Implementierung von Pearl auf kleineren Anlagen scheiterten lange an dem für damalige Verhältnisse großen Speicherplatzbedarf.
Inzwischen stehen auch bei kleinen Computern Speicherkapazitäten zur Verfügung, die eine Installation von Pearl zuließen. Jedoch nur für wenige Kleinrechner oder Microcomputer gibt es Implementierungen (zum Beispiel RTOS-Pearl).
Eines der Hauptmerkmale dieser „Prozeßrechnersprache" ist die Trennung in resistente und nichtresistente Prozeduren.
Pearl wurde gemeinsam von deutschen Wissenschaftlern, dem VDI/VDE, industriellen Anwendern, deutschen Prozeßrechnerherstellern und Softwareproduzenten entwickelt und wurde erstmals 1980 in der DIN 66.253, Teil 2: „Informationsverarbeitung; Programmiersprache Pearl, Full Pearl", festgelegt, allerdings in englischer Sprache mit der Absicht, den internationalen Anwenderkreis zu berücksichtigen.
(Programmbeispiel siehe Tabelle 17.)
Ähnlich anderer höherer Programmiersprachen (etwa Cobol) wird ein Pearl-Programm in verschiedene Divisions unterteilt. Diese Einheiten sind unabhängig voneinander compilierbar. Die getrennt übersetzten Programmteile (Module) werden mit sogenannten Global-Größen miteinander verbunden.
Ein fertiges Modul besteht im allgemeinen aus zwei Divisions: der System-Division, welche die Systemumgebung beschreibt und der Problem-Division, der Problemlösung einer bestimmten Aufgabe.
Im System-Teil eines Pearl-Programms werden die peripheren Geräte und deren Verbindungen beschrieben; insbesondere werden Eingangs- und Ausgangskanäle definiert. Es kann hier weiterhin festgelegt werden, ob die Übertragung auf diesen Kanälen (also von oder zu den peripheren Geräten) digital oder analog erfolgen soll. An dieser Stelle werden die Vorteile der Programmaufteilung in Problem- und System-Division am deutlichsten: Sollen bei bestehendem System einzelne Teile der Peripherie gegen verbesserte Gerätetypen ausgetauscht oder ergänzt werden, müssen lediglich einzelne Teile der System-Division geändert werden. Nämlich nur die dieses Gerät beschreibenden Programmzeilen, nicht jedoch alle Befehle im Programm, welche das entsprechende Gerät ansprechen, wie es in manchen anderen Programmiersprachen der Fall sein kann.

SYSTEM- UND PROBLEM-DIVISION

Bei der Übertragung des Pearl-Programms auf andere Rechnertypen oder Anlagenkonfigurationen kann die System-Division komplett ausgetauscht werden, die Problem-Division kann dabei unverändert bleiben.
Hierdurch ist der System-Teil von Pearl durchaus vergleichbar mit einer Job-Control-Language üblicher Compiler.
In der Problem-Division wird das eigentliche Problem formuliert, das mit Hilfe des Programms bewältigt werden soll. Der Problem-Teil besteht aus verschiedenen Tasks, die eine ähnliche Struktur haben können wie Haupt- und Unterprogramme anderer höherer Programmiersprachen. Diese Tasks aktivieren sich gegenseitig in vorgegebener Abfolge oder gleichzeitig, je nach Anforderung. Die erste Task eines Pearl-Programms wird von außen aktiviert (Programmstart).

Pearl erlaubt die Verwendung von sechs verschiedenen Datentypen:

FIXED
ganze Zahlen,
FLOAT
Gleitkommazahlen,
CHARACTER
Zeichenketten,
BIT
Bitmuster (Kette),
CLOCK
Zeitpunkte,
DURATION
Zeitdauer.

Im Programm verwendete Variablen und Konstanten müssen als einer der oben aufgeführten Datentypen vereinbart werden. Ähnlich wie in Pascal lassen sich in Pearl Felder und Zuordnungen von Variablen bilden. Felder können wieder Felder enthalten und getrennt angesprochen werden.
Hierdurch ist es in Pearl möglich, die Daten hierarchisch zu strukturieren, Gruppen innerhalb von Feldern verschiedenen Prioritätsstufen zuzuordnen.
Pearl-Prozeduren steuern die Übergabe von Parametern und Konstanten. Die Übergabe eines Parameters kann etwa von dessen Wert oder seiner Adresse (Identität) abhängig gemacht werden. Man unterscheidet:

- Initial-Prozedur: Der aktuelle Parameter wird einfach kopiert;
- Identical-Prozedur: Im Gültigkeitsbereich des Prozedurblocks (Modul oder Task) wird ein neuer Name für den Parameter vereinbart;
- Returns sind Funktionsprozeduren, die eine Größe zurückliefern;
- Inline bewirkt eine Direkteinfügung des entsprechenden Prozedur-Parameters.

In Pearl sind Sondervereinbarungen möglich. Dem Benutzer wird gestattet, sowohl neue, abstrakte Datentypen (neben den oben genannten) als auch neue Operationen zu definieren. Besonders interessant und effektiv ist diese Möglichkeit der Sonderfunktionsgestaltung im Zusammenhang mit den oben genannten Feldern.
Die E/A-Struktur von Pearl besteht aus einem Netzwerk von „Datenstationen" und „Kanalumsetzern".

Tabelle 17:

Programmbeispiel zu PEARL
Listing

```
MODULE;                                    /* EXAMPLE PRESSURE CONTROL */

      SYSTEM;
            MULTIPLEXOR                    CPU*5;
            DIGITALOUT(0)                  MULTIPLEXOR*0;
            ANALOGIN(0)                    MULTIPLEXOR*9;
            VALVE          :               DIGITALOUT(0)*0,2;
            LAMP           :               DIGITALOUT(0)*3,1;
            DISCHARGE      :               DIGITALOUT(0)*6,1;
            PRESSURESENSOR1:               ANALOGIN(0)*1;
            PRESSURESENSOR2:               ANALOGIN(0)*3;
            READY:                         INTERRUPTINPUT*0;
            ALARM:                         INTERRUPTINPUT*2;

      PROBLEM;
                        SPECIFY VALVE DATION OUT BASIC DIM() TFU,
                              (LAMP, DISCHARGE) DATION OUT BASIC,
                              (PRESSURESENSOR*1, PRESSURESENSOR*2)
                              DATION IN BASIC DIM() TFU,
                        DECLARE (PRESS1, PRESS2, MEANPRESS) FLOAT,
                              OPENVALVE BIT(2)  INITIAL('10'B),
                              CLOSEVALVE BIT(2) INITIAL('10'B),
                              (TURNON, TURNOFF)
                              BIT INITIAL ('1'B, '='B);

            START:          TASK GLOBAL;     /* ACTIVATED FROM OUTSIDE */

                        EVERY 1 SEC ACTIVATE PRESSURECONTROL;
                        WHEN ALARM ACTIVATE ALARMING;
                        END;                          /* OF TASK STRAT */
            PRESSURECONTROL: TASK PRIORITY 5;
                        TAKE PRESS 1 FROM PRESURESENSOR1;
                        TAKE PRESS 2 FROM PRESURESENSOR2;
                        MEANPRESS = (PRESS1 + PRESS2)/2;
                        IF MEANPRESS = .30
                            THEN SEND OPENVALVE TO VALVE;
                            ELSE IF MEANPRESS = .20 THEN SEND
                            CLOSEVALVE TO VALVE;
                              FIN;
                        FIN;
                        END;              /* OF TASK PRESURECONTROL */

            ALARMING:   TASK PRIORITY 3;
                        SEND TURNON TO LAMP;
                        SEND TURNON TO DISCHARGE;
                        WHILE MEANPRESS = .20
                              REPEAT
                              AFTER 2 MIN RESUME;
                        END;                                  /* LOOP */
                        SEND TURNOFF TO LAMP;
                        SEND TURNOFF TO DISCHARGE;
                        END;                        /* OF TASK ALARMING */
```

Datenstationen verkörpern allgemein alle Peripheriegeräte oder deren Kanäle. Jede Datenstation besteht aus maximal vier Kanälen: dem Datenkanal, dem Steuerkanal, dem Unterbrecherkanal und dem Signalkanal.
Die einzelnen Kanäle müssen im Programm spezifiziert werden.
Kanalumsetzer schaffen die Möglichkeit der Anpassung verschiedenartiger Peripheriegeräte auf einheitliche Formate zur Kommunikation mit der Zentraleinheit oder untereinander. Kanalumsetzer arbeiten wie Interfaces. Sie setzen die Daten innerhalb eines Kanals in einer prozedurartigen Routine um.
Zur Pearl-Syntax gehört eine Reihe von Sprachelementen, die es ermöglichen, beispielsweise parallel ablaufende Prozesse zu steuern und sowohl termin- und zeitgerechte Abläufe zu aktivieren.
Auf die vielschichtigen Befehlsstrukturen kann hier nicht eingegangen werden, aber im obigen Beispiel wird ziemlich deutlich, welche Realtime-Möglichkeiten Pearl besitzt.
Pearl ist für Anwender geschaffen worden. Gegenüber Systemimplementierungssprachen wie Concurrent Pascal, Modula oder Ada ist Pearl leichter erlernbar und bietet zudem große Vorteile.
Die Sprachelemente zur Formulierung algorithmischer Zusammenhänge und Abläufe entsprechen dem Standard moderner Programmiersprachen (so etwa Pascal, PL/1).
Pearl hat durch die Aufteilung in Divisions ein hohes Maß an Portabilität und Dokumentationswert. Strukturiertes Programmieren wird unterstützt. Pearl ist in vielen Anwendungsbereichen eine echte Alternative zur Assembler-Programmierung.

52. PILOT

Programmed Inquiry Learning Or Teaching.
PILOT ist eine Dialogsprache (siehe auch BASIC) und wurde von Prof. John Starkweather entwickelt.
PILOT hat nie den Bekanntheitsgrad von BASIC erreichen können. Gründe hierfür sind unter BASIC (im ersten Teil unserer Serie) nachzulesen.

Im Gegensatz zu BASIC ist PILOT aber eine Dialogsprache ohne Kompromisse. Sie wird oft als *die* Dialogsprache schlechthin bezeichnet. Heutigen Versionen, wie dem Utah-PILOT (auch für IBM-PC kompatible Rechner unter MS-DOS erhältlich), liegt der PILOT-73-Standard zugrunde, – ständig erweitert und verbessert. PILOT ist für Anfänger geschrieben und leicht erlernbar und kann daher auch ideal für Lernzwekke eingesetzt werden. Als Dialogsprache ist sie im besonderen Maße für interaktive Anwendungen geeignet. Hierzu werden großartige Hilfestellungen in der Syntax geboten: spezielle Dateneingänge, programmierte Instruktionen, Testroutinen und anderes. Interaktive Anwendungen liegen dann vor, wenn im besonderen Maße Wert auf den Dialog zwischen Mensch und Maschine gelegt wird (was oft problembedingt sein kann). PILOT hat ein recht einfaches Format und eine leicht verständliche Syntax.

Tabelle 18:

Programmbeispiel zu PL/I:
Listing

```
 1:     BLOCK1: PROCEDURE OPTIONS(MAIN);
 2:             DECLARE A DECIMAL FIXED(5,2),
 3:                     B BINARY FLOAT(24),
 4:                     C CHARACTER(20), VARYING;
 5:             A = 2.34;
 6:             B = 4.56E+4;
 7:             C = 'BEISPIEL';
 8:             PUT SKIP(2) LIST('BLOCK 1, ZEILE 9:');
 9:             PUT SKIP LIST('A =' , A,' B =' ,B,' C = ' ,C);
10:
11:         BLOCK2: BEGIN;
12:                 DECLARE A BINERY FLOAT(18) EXTERNAL,
13:                         X BINERY FIXED(15),
14:                 A = 2.875E+3;
15:                 B = 5.625E-3;
16:                 X = 275;
17:                 PUT SKIP(2) LIST('BLOCK 2, ZEILE 18:');
18:                 PUT SKIP LIST('A =' ,A,' B =' ,B,' C =' ,C,' X =' ,X,);
19:                 END BLOCK2;
20:
21:             PUT SKIP(2) LIST('BLOCK 1, ZEILE 22:');
22:             PUT SKIP LIST('A =' ,A,' B =' ,B,' C =' ,C);
23:
24:         BLOCK 3: BEGIN;
25:                 DECLARE A BINARY FLOAT(!() EXTERNAL,
26:                         C CHRACTER(!=) VARYING,
27:                         X BINARY FIXED(!%);
28:                 X = 386;
29:                 C = ' BEISPIEL 1';
30:                 PUT SKIP(2) LIST('BLOCK 3, ZEILE 31:');
31:                 PUT SKIP LIST('A =' ,A,' B =' ,B,' C =' ,C' X =' ,X,);
32:                 END BLOCK3;
33:
34:             PUT SKIP(2) LIST('BLOCK 1, ZEILE 35:');
35:             PUT SKIP LIST('A =' ,A,' B =' ,B,' C =' ,C);
36:
37:             END BLOCK1;
```

53. PLANKALKÜL

Eine der ersten universellen algorithmischen Sprachen überhaupt. Entwikkelt wurde sie schon 1945 von Dr. Ing. e.h. Konrad Zuse, dem Erbauer der ersten programmgesteuerten Rechenanlage. PLANKALKÜL war wegweisend für die Entwicklung aller symbolischen Programmiersprachen.

54. PL/1

Programming **L**anguage Nr. 1

PL/1 könnte man als Synthese aus Fortran, Algol und Cobol bezeichnen, was ihren Anwendungsbereich sowohl in den technisch-mathematischen als auch in den kaufmännischen Bereich legt (Mischsprache). Hier wurde versucht, die aus der Erfahrung mit diesen Sprachen erkannten Mängel zu vermeiden und die Vorteile in einer gemeinsamen Sprache zu vereinen.
Von Algol wurde die strenge Einteilung eines Blocks in Vereinbarungs- und Anweisungsteil, die Blockstruktur sowie die formatfreie Ein- und Ausgabe, die Ablaufstrukturen, Records, Pointer und das Prozedure-Konzept mit rekursivem Aufruf übernommen. Cobol lieferte die Dezimalarithmetik, Picture-Formatierung, Textverarbeitung, Datenstrukturierung und eine umfangreiche Dateiverwaltung. Aus Fortran kommen die impliziten Laufanweisungen, numerische Standardfunktionen, die Functions-Subroutine und das Prinzip der Common-Bereiche (siehe hierzu auch unter Fortran).
Der Nachteil: PL/1 ist in vollem Umfang nur auf größeren Rechenanlagen anwendbar, da der benötigte Speicherplatz wesentlich größer ist als der bei anderen Programmiersprachen.
Auch die Syntax ist umfangreicher, was das Erlernen von PL/1 erschwert.
Vorteile: PL/1 ist flexibler in den Anwendungen und problemloser bei der Verarbeitung großer Datenmengen innerhalb der Rechenanlage. PL/1 vereint die Vorteile verschiedener anderer Programmiersprachen in sich: auf der einen Seite gute Texthandhabung und Dialogfähigkeit, auf der anderen Seite die mathematischen Möglichkeiten der genannten Programmiersprachen.
Der modulare Aufbau von PL/1 gestattet es dem Anfänger, zunächst einen kleinen Teil der Syntax zu erlernen und sich dann schrittweise den gesamten Sprachumfang anzueignen. Vieles wird vom PL/1-Compiler übernommen, wie zum Beispiel die Speicherplatzorganisation, Datenumwandlungen, Formatierung der Ein- und Ausgabe und anderes.
Man spricht hierbei vom Default-Konzept. Der PL/1-Sprachumfang ist so umfassend, daß fast alle Anwendungsgebiete abgedeckt werden können. So muß nicht bei wechselnden Anforderungen auch die Programmiersprache gewechselt werden.
PL/1 unterstützt Mehrbenutzer-Betriebssysteme durch mehrstufige Passwortabfrage und Dateischutz bis zum Sperren.
Integriert sind außerdem

FLEXIBEL UND UMFASSEND

umfangreiche Möglichkeiten zur Behandlung von Ausnahme- und Fehlersituationen sowie Overlaytechniken.
PL/1 wurde zum ersten Mal 1965 vorgestellt. Seitdem wurde sie, wie viele andere Programmiersprachen auch, weiterentwickelt. Da dies aber nicht durch einen neutralen Ausschuß geschah (ANSI), gibt es heute mehrere PL/1-Dialekte. Erst 1979 wurde PL/1 in einer Norm vereinheitlicht:

- ISO 6160: „Programming Language PL/1" (international);
- DIN 66.255: „Informationsverarbeitung; Programmiersprache PL/1", Ausgabe 1980.

Jedoch in englischer Sprache, da man eine Abweichung bei der Übersetzung ins Deutsche befürchtete, was unter allen Umständen vermieden werden sollte, denn die Deutsche Industrie-Norm sollte fachlich der internationalen (ISO) Norm entsprechen.
PL/1 gehört heute zu den am meisten verbreiteten Programmiersprachen und ist auch für die Simultanverarbeitung geeignet. Simultanverarbeitung liegt

dann vor, wenn ein Programm gleichzeitig mit mehreren peripheren Geräten (Drucker, Dateien...) arbeitet. Hierbei entsteht weniger Wartezeit, da die Peripherie einer Anlage sehr viel langsamer ist als die Zentraleinheit. Steuereinheiten in den Kanälen ermöglichen den gleichzeitigen Einsatz mehrerer Geräte.
Ein weiterer Vorteil von PL/1 liegt in den vorhandenen Sprachbestandteilen zur Gliederung von Programmen.
In PL/1 werden Unterprogramme als Prozeduren geschrieben, die wiederum von anderen Prozeduren aufgerufen werden können. Komplizierte Aufgabenstellungen können so in logisch unterteilte Abschnitte (Module)

MODULTECHNIK

untergliedert werden. Dies verbesserte die Effizienz eines Programms erheblich, vor allem in Hinblick auf die Transparenz des Lösungswegs. Korrekturen und Änderungen (die Programmpflege) werden dadurch erheblich erleichtert.
Alle Prozeduren, die nach dem Start des Haupt-Programms bis zur Rückgabe der Ablaufsteuerung an das Betriebssystem aktiviert werden, bilden das Gesamtprogramm.
Dieses kann aus internen und externen Prozeduren bestehen. Das Hauptprogramm ist dabei immer eine externe Prozedur, die durch den Zusatz Options(Main) gekennzeichnet wird, alle anderen Prozeduren sind Unterprogramme.
Interne Prozeduren sind Bestandteile anderer Prozeduren (übergeordnete Prozedur) und werden mit dieser gemeinsam übersetzt und gebunden. Sie können nur mit der übergeordneten Prozedur verwendet werden.
Im Gegensatz hierzu werden externe Prozeduren separat übersetzt und bilden eigenständige Module. Diese können in Bibliotheken abgelegt und bei Bedarf an Hauptprogramme angebunden werden.
PL/1 gibt es in der Zwischenzeit auch für viele Klein- und Micro-Computersysteme unter CP/M oder MS-DOS.
Bekannt wurde noch PL/9900, eine Abwandlung von PL/1, entwickelt von Texas Instruments für deren Rechnerkonzept 9900. Inzwischen hat aber fast jeder Hardware-Hersteller eigene Implementationen von PL/1 für seine Maschinen im Angebot, sofern die Rechner für PL/1 geeignet sind.
Eine interessante Neuentwicklung stellte auch PL/M dar, eine spezielle, abgemagerte Version für Micro-Computer.
(Programmbeispiel siehe Tabelle 18.)

55. Prolog

PROgramming in **LOG**ic.
Prolog ist eine deklarative Programmiersprache, wie sie zur Lösung von Problemen aus dem Bereich der künstlichen Intelligenz (KI) eingesetzt werden.
Deklarative Sprachen beschreiben die vom Programm zu lösende Aufgabe, der Lösungsweg (Algorithmus) muß nicht definiert werden. Daher spricht man auch von logischer oder symbolischer Programmierung.
Die in Prolog bevorzugt verwendeten Datenstrukturen sind einfache und verkettete Bäume. Der Begriff des Datentyp im Sinne konventioneller Programmiersprachen existiert in der logischen Programmierung nicht.
Verschiedene Prolog-Compiler verlangen jedoch eine Deklarierung von Variablen für die Compilierung des Quellcodes.
Statt Datentypen werden in Prolog Domain-Typen verwendet:

integer	**ganze Zahlen**
real	**reelle Zahlen**
char	**Zeichen**
string	**Zeichenkette**
symbol	**Symbol**
file	**Datei-Name.**

Integerzahlen liegen zwischen −32.768 und +32.767.
Reelle Zahlen sind durch einen Dezimalpunkt gekennzeichnet, auch wenn keine Nachkommastellen auftreten.
Größere Zahlen werden in exponentieller Schreibweise dargestellt. Die Char-Variablen können einzelne Zeichen enthalten. Die Zeichen werden mit Hochkommas geschrieben. Zeichenketten (Strings) sind beliebige Zeichenkombinationen, auch sie werden durch Hochkommas abgegrenzt.
Symbol-Variablen beginnen mit Kleinbuchstaben oder sind Zeichenfolgen, die von Hochkommas eingeschlossen sind. Symbol und String werden in Prolog unterschiedlich verarbeitet.
File-Variablen ermöglichen die Definition symbolischer Dateinamen, welche als Argumente in Prädikaten verwendet werden können.
Jedes Programm darf aber nur einen Domaintyp enthalten, der zur Gruppe „File" gehört.
In der Deklarierung können mehrere Objekte des gleichen Domain-Typs zusammengefaßt werden. In Prolog ist es möglich, für die Definition einer Domain diese selbst zu verwenden. Dadurch entsteht ein rekursives Objekt.
Standard-Domaintypen müssen nicht deklariert werden.
Ein Prolog-Programm ist als Sammlung von Fakten und Regeln auch eine Datenbank. Durch Einfügungen und Löschungen während der Ausführung ist sie dynamisch.
Standard-Prolog erlaubt den Austausch von Fakten und Regeln. Manche Prolog-Dialekte wie Turbo-Prolog lassen nur den Austausch von Fakten zu.
Die Anordnung von Fakten und Regeln, sowie der Datensätze ist in Prolog sehr wichtig für die Ausführung des Programms. Je nach Anordnung kann ein Programm auf verschiedene Weisen interpretiert werden und es kann zu völlig unterschiedlichen Ergebnissen kommen.
An dieser Stelle sollte vielleicht einmal etwas Grundsätzliches zu den sogenannten KI-Sprachen gesagt werden.
Wie schon erwähnt, handelt es sich hier um deklarative Programmiersprachen. Im Gegensatz zu den herkömmlichen Sprachen wird nur ein Problem, nicht aber der komplette Lösungsweg beschrieben.
In algorithmische Sprachen wie Fortran, Cobol, BASIC, Pascal und andere müssen Lösungswege für alle möglichen Fälle bei der Programmierung definiert werden, was sehr aufwendig sein kann und damit auch als Qualitätsmerkmal eines Programms gilt.
Logische oder deklarative Programmiersprachen erlauben aber die Festlegung von Fakten (Facts) und Regeln (Rules), die für die eigentliche Lösung des Problems notwendig sind. Also nicht der Programmierer löst das gestellte Problem, sondern das Programm. Die Hauptarbeit des Programmierers besteht darin, die „Facts" und „Rules" korrekt und vollständig zu formulieren.

KÜNSTLICHE INTELLIGENZ

Bei Prolog ist das oben Gesagte leider nur zum Teil verwirklicht worden. Auch Prolog kann nicht ganz ohne explizite Anweisungen (Algorithmen) auskommen. Bei den konventionellen Programmiersprachen ist dieser Algorithmus das Programm selbst. Bei Prolog ist der benötigte Algorithmus inhärenter Bestandteil des Interpreters oder des Compilers. Aus diesem Grunde wurde schon mehrfach versucht, Prolog den deklarativen Charakter abzuerkennen. Sicher ist Prolog keine reine deklarative Programmiersprache, sondern eher eine Mischform,

wie viele höhere Programmiersprachen.
Um einen Eindruck zu gewinnen, wie ein Prolog-Programm aussieht, in Tabelle 19 ein Ausschnitt, geschrieben in Turbo-Prolog.
Es gibt viele Prolog-Implementierungen, von denen sich die meisten an den De-Facto-Standard halten.
Prolog-Programme werden mit einem gewöhnlichen Editor erstellt. Manche Implementationen erlauben die Syntax-Überprüfung schon während des Editierens.
Zudem werden sogenannte Software-Interfaces angeboten, die Schnittstellen zu anderen Programmiersprachen darstellen (etwa zu C).
Grundsätzlich zu unterscheiden ist zwischen Prolog-Interpreter und Compilern. Vor- und Nachteile dieser Übersetzungsmethoden wurden schon hinreichend besprochen. Sie gelten für alle in Frage kommenden Programmiersprachen.

Tabelle 19:

```
Programmbeispiel zu Turbo-PROLOG
Listing

domains
      position = integer*
      nummer = integer
      zugliste = nummer*

Predicates
      regel ( nummer, position, position)
      wende (integer, integer)
      endposition (position)

      postcheck (position)
      gültig (poditicn)

      löse (position, integer, zugliste)
      run

clauses
      regel(1,(x1,x2,x3,x4,x5,x6,x7,x8,x9),(y1,y2,y3,y4,y5,y6,y7,y8,y9,)
      if wende(x1,y1), wende(x2,y2), wende(x4,y4), wende(x5,y5),
      x3 = y3, x6 = y6, x7 = y7, x8 = y8, x9 = y9
      regel(2,(x1,x2,x3,x4,x5,x6,x7,x8,x9),(y1,y2,y3,y4,y5,y6,y7,y8,y9,)
      if wende(x1,y1), wende(x2,y2), wende(x3,y3),
      x4 = y4, x5 = y5, x6 = y6, x7 = y7, x8 = y8, x9 = y9
      regel(3,(x1,x2,x3,x4,x5,x6,x7,x8,x9),(y1,y2,y3,y4,y5,y6,y7,y8,y9,)
      if wende(x2,y2), wende(x3,y3), wende(x5,y5), wende(x6,y6),
      x1 = y1, x4 = y4, x7 = y7, x8 = y8, x9 = y9
      .
      .
      .
      wende (0,1).
      wende (1,0).

      endposition ((1,1,1,1,0,1,1,1,1)).

      poscheck (AP) if gültig (AP,9).
      poscheck (-) if
          beep, nl,
          write ("Eingabefehler."), nl,
          !,
          fail.

      gültig ((),0).
      gültig ((X|T),N) if
         X )= 0,
         X (= 1,
         N1 = N - 1
         gültig (T,N1)

      löse (A,-,Y) if
         endposition (A)
         !,
         Z = ().
         .
         .
         .
goal
       run.
```

56. Prosa
PROgrammiersprache mit **S**ymbolischen **A**dressen.
Prosa ist wie Autocode ein Vorläufer der heute gebräuchlichen Assembler-Sprachen. Wie diese ist Prosa maschinenorientiert. Prosa findet heute kaum noch Anwendung. Gebräuchlich war Prosa zwischen 1966 und 1965.

57. QBE
Query **B**y **E**xample.
Grafische Datenbank-Abfragesprache, von IBM Ende der 70er Jahre im San Jose Research Laboratory entwickelt.
QBE ist non-prozedural (siehe auch SQL).
Im Gegensatz zu SQL ist QBE allerdings nur für große Rechenanlagen tauglich.

58. QL
Query **L**anguage.
Abfragesprache (DL-Sprache).
Query Language hat eine sehr einfache Syntax und Struktur und ist daher leicht erlernbar.
Sie wird hauptsächlich zur Handhabung (Abfragung) von Datenbanksystemen benutzt (siehe auch SQL).
DL-Sprachen sind meist nicht-prozedural. Der Anwender teilt dem System mit, wie das zu lösende Problem aussieht und nicht, wie es zu lösen ist.

59. RPG
Report-**P**rogramm-**G**enerator, auch **LPG** (**L**isten-**P**rogramm-**G**enerator).
RPG und das heute gebräuchlichere RPG II, eine Weiterentwicklung, sind eigentlich keine echten höheren Programmiersprachen im eigentlichen Sinne, sondern, wie der Name schon sagt, Programm-Generatoren, also Systeme, die von Dateien eingelesene Daten-

AUSWERTUNG VON DATENSÄTZEN

sätze nach vorherbestimmten, gleichartigen Umformungen bearbeiten und meistens in Form einer Liste (daher LPG) ausgeben. RPG findet deshalb hauptsächlich in der kommerziellen Datenverarbeitung Anwendung, wo solche Probleme überwiegend auftreten.
Man spricht bei RPG von einer parametrischen Programmierung, da alle Anweisungen und Angaben in Form von Parametern beschrieben werden. Mit Hilfe dieser Parameter wird das Programm dann erstellt.
Erst die Entwicklung von RPG II konnte eine vollständige, gute und umfassende Bearbeitung aller, im kommerziellen Bereich auftretenden Probleme bewältigen. Dennoch kann RPG nicht zu den problemorientierten Programmiersprachen gezählt werden, obwohl einige Softwarehäuser dies gerne sähen. RPG enthält keine frei kombinierbaren Befehle. Und dies ist nun einmal eines der Kriterien für eine problemorientierte Programmiersprache.
RPG wurde ursprünglich von IBM entwickelt und von verschiedenen anderen Softwareherstellern weiterentwickelt. RPG II

wird meist in die Standard-Sprachumgebung des Rechners eingegliedert. Programme in RPG II können Module in anderen Sprachen aufrufen sowie diverse Laufzeit-Bibliotheksprogramme. Andererseits werden die Ergebnisse der RPG II-Programme (Listen) so ausgegeben, daß sie ohne weiteres auch von anderen Programmen in anderen Sprachen weiterverarbeitet werden können.

60. RPNL

Reverse **P**olish **N**otation **L**anguage.
RPNL ist eine höhere Programmiersprache, die sowohl mit Pascal als auch mit Forth große Ähnlichkeiten aufweist. Der erste Unterschied zu anderen Programmiersprachen ist die Verarbeitung der „Umgekehrten Polnischen Notation" (UPN) bei den Operationen (siehe hierzu auch EOL). RPNL unterstützt die strukturierte Programmierung und läßt hohe Ausführungsgeschwindigkeiten zu. RPNL ist compilerorientiert, obwohl das vom Compiler erzeugte Objektprogramm noch keinen direkten Maschinencode darstellt. Man spricht hierbei von einem „Zwischencode". Dieser Zwischencode besteht aus einer Zusammenstellung von Bibliotheksprogramm-Aufrufen, die allerdings sehr maschinennah sind, was sich positiv auf die Ausführungsgeschwindigkeit auswirkt.
In RPNL werden grundsätzlich nur zwei Datentypen unterschieden:
■ *Numerische Daten:*
INTEGER-TYP
(bis zur Größe 2 hoch 16 –1),
BOOLEAN-TYP
(True, False),
ARRAY-TYP
(Zusammenfassung verschiedener oder gleicher Typen).
■ *Alphanumerische Daten:*
STRING
(maximal 255 Zeichen lang).

Alle in einem RPNL-Programm verwendeten Variablen oder Konstanten müssen am Programmanfang definiert werden. Folgende Namen dürfen nicht als Variablen oder Konstanten benannt werden, da sie bereits vergeben sind:

■ **I, J, K**
als Laufindizes in Schleifen,
■ **TRUE, FALSE**
als logische Werte für „wahr" oder „falsch".

Tabelle 20:

Wahrheitstabelle der Boolschen Operatoren (Prädikate)

Operand 1	Prädikat	Operand 2	Ergebnis
TRUE	NOT	-	FALSE
FALSE	NOT	-	TRUE
TRUE	AND	TRUE	TRUE
TRUE	AND	FALSE	FALSE
FALSE	AND	TRUE	FALSE
FALSE	AND	FALSE	FALSE
TRUE	OR	TRUE	TRUE
TRUE	OR	FALSE	TRUE
FALSE	OR	TRUE	TRUE
FALSE	OR	FALSE	FALSE

Der Programmaufbau eines RPNL-Programmes ist grundsätzlich in drei Teile gegliedert:

1. Programmklammer mit Namen
Die erste Zeile eines RPNL-Programmes enthält immer die Anweisung:
PROGRAMM Programmname
In der letzten Zeile steht immer .
.
END

2. Deklarationsliste
Beispielsweise VAR für Variablenname,
CONST für Konstantenname,
Anzahl ARRAY Feldname.

Hier wird einerseits der Variablen/Konstanten- und Feldname deklariert, andererseits Speicherplatz zugewiesen und belegt: normalerweise für jede Größe 16 Bit, oder für die Größe des Feldes (Anzahl) * 16 Bit.

3. Anweisungsliste
Die Anweisungsliste enthält die Algorithmen, welche den Lösungsweg des Problems beschreiben.
Man unterscheidet bei RPNL drei verschiedene Anweisungstypen:
a) einfache Anweisungen,
b) E/A-Anweisungen für die Ein- und Ausgabe,
c) Steueranweisungen für Datenfluß und Ausführungssteuerung.
Einfache Anweisungen können aus einstelligen oder zweistelligen Operatoren bestehen. Einstellige Operatoren bearbeiten einen Operanden, zweistellige Operatoren benötigen zwei Operanden.

Einstellige Operatoren:
INC bildet das Inkrement des angegebenen Operanden (=Erhöhung um 1),
DEC das Dekrement wird gebildet (Verringerung um 1),
NOT logische Verneinung (Negation),
DUP Duplizierung eines Wertes (Kopie),
? Abruf eines Speicherplatzes, einer Variablen oder Konstanten.

Zweistellige Operatoren:
+ Addition,
– Subtraktion,
DIV Division (nur ganzzahlig), der Rest ist verloren,
***** Multiplikation,
MOD Division zweier Operanden, welche nur den errechneten Rest anzeigt (Modulo Division),
SWAP Vertauschung von zwei Operanden,
AND Logisches UND (Konjunktion),
OR Logisches ODER (Disjunktion).

Zur „Wahrheitstabelle der Boole'schen Operatoren" siehe *Tabelle 20.*

Außerdem stehen noch die logischen Vergleichsoperatoren gleich/größer/kleiner und deren Kombination zur Verfügung. Sie liefern für Integer-Operanden logische Wahrheitswerte als Ergebnis.

Ein- und Ausgabeanweisungen und RPNL-Prozeduren:
CR Carriage Return (Zeilenvorschub)
WRITE Ausgabe von Text (in Hochkommas),
PRINT Ausgabe einer Integerzahl,
PRINT(S) Ausgabe eines Strings,
READ liest eine Integerzahl ein,
READ(S) liest einen String ein,
STRING weist einem Text Speicherplatz zu.

Bedingungen und Schleifen
IF-THEN-ELSE
Vor dem Schlüsselwort IF steht ein Ausdruck, der die Bedingung darstellt (ein logischer Wert), da-

hinter die Anweisungsliste, die nur dann ausgeführt wird, wenn die Bedingung erfüllt ist (TRUE). Die Anweisungsliste hinter ELSE wird genau dann ausgeführt, wenn die Bedingung vor dem IF nicht erfüllt ist (FALSE).

REPEAT UNTIL LOOP
Hinter REPEAT folgt die Anweisungsliste, deren Anweisungen wiederholt werden sollen. Abgeschlossen wird die Schleife durch das Schlüsselwort LOOP. An einer geeigneten Stelle zwischen REPEAT und LOOP steht UNTIL, dem direkt ein Ausdruck (logischer Wert) vorangeht.
Abhängig von dem aktuellen Wert des Ausdruckes erfolgt die Fortsetzung des Programms mit der ersten Anweisung hinter UNTIL (bei FALSE) oder hinter LOOP (bei TRUE) und damit wird aus der Schleife herausgesprungen.

FOR LOOP
Vor dem Schlüsselwort FOR müssen zwei Operanden angegeben werden, die den Endwert und den Startwert der Laufvariablen festlegen. Zwischen FOR und LOOP steht die Anweisungsliste. Es können drei Schleifen ineinander verschachtelt werden. Die Variable I ist immer für die innere Schleife reserviert, J für die mittlere und K für die äußere. Bei nur einer Schleife oder einer Verschachtelung von zwei Schleifen gilt das oben Gesagte analog.
(Programmbeispiel siehe Tabelle 21.)

61. SIMULA

SIMULA ist mehr ein System als eine Sprache. Mit Hilfe des SIMULA-Systems lassen sich beispielsweise rechnerinterne Abläufe anhand von Modellen untersuchen. So wurden umfangreiche Untersuchungen von Betriebssystemen mit Hilfe von SIMULA durchgeführt.
SIMULA gibt es seit etwa 1965.
Der Vorteil gegenüber anderen Programmiersprachen liegt in der Struktur des Systems, welches sich besonders zu Simulationszwecken eignet (daher der Name SIMULA).
Es gibt verschiedene SIMULA-Systems, auf die aber wegen ihrer Spezialisierung hier nicht eingegangen werden kann.

Tabelle 21:

```
Programmbeispiel zu RPNL
Listing

PROGRAMM BEISPIEL
  5 ARRAY LISTE
  VAR ZAHL
  4  0  FOR
     REPEAT
     WRITE "EINGABE" CR
        ZAHL READ
        ZAHL ? I 2 * LISTE + ! =
     ZAHL ? 100 (= UNTIL
        ZAHL PRINT
        WRITE "ZU GROSS" CR
        LOOP
     LOOP
END
```

62. SIL 3

System Implementation Language 3.
Maschinenunabhängige System-Implementierungssprache (SIL).
Sie dient zur Software-Herstellung und zeichnet sich in erster Linie durch ihre Codeeffizienz aus.
Die Programmierung von SIL ist recht kompliziert, nicht zuletzt durch die fehlende Problemnähe und die speziellen Anforderungen an der Anwendersoftware.

63. Smalltalk 80

Smalltalk wurde Anfang der siebziger Jahre entworfen, 1980 fertiggestellt (daher Smalltalk 80) und 1981 zum ersten Mal der Öffentlichkeit vorgestellt. Im „Xerox Paolo Alto Research Center" entwickelt, ist Smalltalk zugleich Programmiersprache und Rechensystem. Großen Wert bei der Ausstattung dieses Systems wurde auf grafische Interaktionsmechanismen wie Windows gelegt. Menü- und Maustechniken werden ebenfalls unterstützt. In Smalltalk konnte also schon das realisiert werden, was später von Betriebssystemen (grafischen Benutzeroberflächen) wie GEM oder anderen übernommen wurde.
Das Ausgabemedium für Smalltalk ist immer ein Bitmap-Display, da jede Ausgabe grundsätzlich grafisch erfolgt. Dies geschieht mit Hilfe sogenannter „Forms". Cursor, Bildfläche und auch Schriftarten werden alle als Forms gehandhabt. Smalltalk enthält zwei Editoren zur Veränderung der Forms.
Auch echte Grafik ist in Smalltalk möglich: Liniengrafik und Splines, Kreise sowie andere geometrische Figuren.
Zur Simulation gibt es in Smalltalk das sogenannte Klassenkonzept, welches von der Programmiersprache SIMULA übernommen wurde. Die Klassen dienen den Wahrscheinlichkeitsverteilungen und ereignisgetriebenen Simulationen.
Smalltalk ist, wie schon erwähnt, nicht nur eine Sprache, sondern gleichzeitig ein System. Das System ist seinerseits in Smalltalk geschrieben und basiert auf einer virtuellen Maschine. Das System ist dem Benutzer zugänglich und ermöglicht so Eingriffe.
Für alle Anwenderprogramme wird ein einheitlicher Mechanismus für die Realisierung der interaktiven Schnittstelle zur Verfügung gestellt – ein ähnliches Vorgehen wie bei den Betriebssystemen mit grafischer Benutzeroberfläche. Dieser Mechanismus besteht aus den Bestandteilen Model, View und Controller. Das Modell wird von den Daten und Informationen gebildet. Die View beschreibt die Darstellungen und der Controller übernimmt die Steuerung, also er beschreibt, welche Auswirkungen Ereignisse (etwa Benutzereingaben) auf das System haben.

SPRACHE UND SYSTEM

Smalltalk ist objektorientiert. Jede Komponente von diesem System ist ein Objekt. Die Programme in Smalltalk beschreiben lediglich die Kommunikation dieser Objekte untereinander und natürlich die Objekte selbst.
Die Programmierung in Smalltalk besteht daher im Wesentlichen aus der Neubildung von Objekten und dem „Senden von Botschaften" zwischen diesen.
Die Objekte in Smalltalk 80 sind sehr abstrakt und haben kaum noch etwas mit den technischen Eigenschaften der Hardware des Rechners zu tun oder den klassischen Objekten in der Software, also Zahlen oder Zeichen. Smalltalk gibt es für ver-

schiedene Rechnersysteme, in der Regel sind dies aber keine Microcomputer. Lediglich für den Macintosh von Apple gibt es eine Implementation.

64. SQL

Structured Query Language.
Strukturierte Abfrage-Sprache, vorwiegend für Datenbanken entwickelt von IBM im San Jose Research Laboratory.
SQL ist seit Ende der 70er Jahre auf dem Markt und wurde seitdem in verschiedensten Datenbanksystemen installiert. Bekanntestes Beispiel ist die relationale Datenbank ORACLE (1979).
SQL ist non-prozedural, also muß nicht der Lösungsweg vom Programmierer beschrieben werden, sondern nur das Problem selbst, das Programm sucht nach Lösungen.
Datenstrukturen können in Form von Tabellen dynamisch definiert und manipuliert werden. Diese Möglichkeiten stehen dem Benutzer über ein spezielles Interface mit dem bezeichnenden Namen UFI = User Friendly Interface zur Verfügung, welches die Datenmanipulation im Dialogbetrieb zuläßt.
SQL besitzt erstaunliche Abfragemöglichkeiten, die jedoch je nach gewünschter Abfrage Zeit kosten können.
SQL bietet hierfür die Möglichkeit, Indizes zur Retrieval-Beschleunigung zu definieren. Die Indexdefinition kann auch dazu benutzt werden, um eine von SQL nicht abgedeckte, jedoch gewünschte Leistung sicherzustellen.
Neben SQL hat sich – ebenfalls von IBM entwikkelt – eine weitere Abfrage-Sprache etabliert: QBE (Query By Example) – mehr ein grafisches Abfragesystem. QBE, auch als Ergänzung zu SQL, ist nur auf mittleren und großen Rechenanlagen einsatzfähig.
Für Mini- und Microcomputer (so VAX von DEC und IBM, PC, XT, AT) gibt es SQL ebenfalls in Verbindung mit der Datenbank ORACLE.

65. SYMAP

Eigenentwicklung der DDR, aufgebaut aus verschiedenen Elementen von Fortran und Exapt.

66. V.I.P.

Visuelle Interaktive Programmiersprache.
Programme in V.I.P. werden über Flußdiagramme eingegeben. V.I.P. ist auf den Apple-Macintosh zugeschnitten und erlaubt Zugriffe auf die Macintosh-Toolbox. V.I.P. verfügt über einen visuellen Debugger und einen sehr schnellen Interpreter. Es gibt auch Transferprogramme, die den V.I.P.-Code in LS-Pascal übersetzen.

67. PL/PC

Programming Language for Personal Computers.
PL/PC ist eine relativ neue Universal-Programmiersprache, speziell geschaffen zum Einsatz auf PCs unter MS-DOS.
Wie BASIC ist PL/PC interaktiv, hat aber diesem gegenüber den Vorteil, strukturiert zu sein (wie MODULA-2).
Auch von APL sind Elemente bei der Schaffung dieser neuen Programmiersprache übernommen worden; so zum Beispiel die mächtigen Datenstrukturen und Manipulationsfunktionen.
Zum Sprachumfang gehören Editoren für Datenstrukturen, die wie ein Spreadsheet aufgebaut sind, und Text zur Deklaration von Subroutinen, Compiler für selbst definierte Subroutinen, Interpreter, Debugger, Grafikunterstützung für alle bekannten Bildschirme einschließlich Turtle-Grafik.
Für rechenintensive Anwendungen existiert eine Spezialversion, welche die Coprozessoren 8087 und 80287 unterstützt.
Variable können ihren Typ (Boolean, Byte, Integer, Long [32 Bit], Real und Complex, String, Charakter) dynamisch ändern.
Die meisten mathematischen Funktionen sind eingebaut.
PL/PC ist leicht lernbar und daher auch für Anfänger geeignet, hat aber gegenüber BASIC gewisse Vorteile.
Am Ende dieses Beitrages können wir nur hoffen, etwas Klarheit in dieses Sprachengewirr gebracht zu haben.

Ende der Serie

Oliver Rosenbaum □

BÖRSE

Schneider CPC 1512 MM/20HD u. Epson LX-800-Drucker (auch einzeln) zu verkaufen. Tel. 0921/44520 ab 18 Uhr

---

SUCHE für mich u. meinen CPC 6128 m. Drukker Star NL10 Heimarbeit. Tel. 06181/690864 ab 18 Uhr

---

Schneider CPC 464 zu verkaufen! Keyboard, Mon., Laufw. nur 650,– Sfr m. 2 Joysticks, Disketten u. 70 Zeitschr.! Interessiert? Dann anrufen unter CH-041/452507 od. schreiben an B. Zdziech, Kosthausstr. 8, CH-6010 Kriens

---

SUCHE Schrifttum (m. Prog.) f. Sinclair Spectrum od. Angabe v. Bezugsquelle. Zweck: Hilfe f. Computerfan in der DDR. J. Mertig, Buchenweg 8, 8209 Stephanskirchen

---

VERKAUFE CPC 464 u. Zubehör für 230,– DM; Drucker NLQ401 u. Traktor u. Kabel für 300,– DM; DDI-LW neu für 400,– DM; CPC 6128 für 700,– DM; AMX-Pagemaker für 80,– DM; Maus für CPC (neu) für 150,– DM; Div. orig. Spiele (Reisende, Sentinel etc.) je 15,– DM. Tel. 08334/1513

VERKAUFE CPC 464, 128 KB RAM, m. Vortex Doppelstation F1D (2x 704 KB) u. DOS-COPY (MS-DOS Diskmanager), CP/M plus u. CP/M 2.2 Wordstar u. d-Base II f. 950,– DM; Multiface II Kopiermodul m. Monitor, Freeze etc. f. 100,– DM. Tel. 06051/73745

ACHTUNG! ACHTUNG! Suche komplettes Bundesligaprog. m. Tabelle u. Speichern der Ergebnisse. Disk Schneider CPC 664. Thomas Mueller, Schlesienstr. 22, 8608 Lichteneiche. Tel. 0951/30572

SCHNELLE Zellsimulation bei max. 18 Bildern/Sek. Variable Gesetze; läuft auf allen CPCs. Faszinierende Effekte; 100% M-Code. 20,– DM an M. Ostermann, Sanderstr. 5r, 8700 Würzburg

---

VERKAUFE Schneider CPC 464 m. DDI1 u. Programmen f. DM 1.000,–. Tel. 06894/35919

SUCHE KONTAKT zu CPC-Usern. Infos u. Prog.-Austausch sehr willkommen. M. Kiontke, Weilerbachweg 9, 7320 Göppingen 6

Masken-Generator

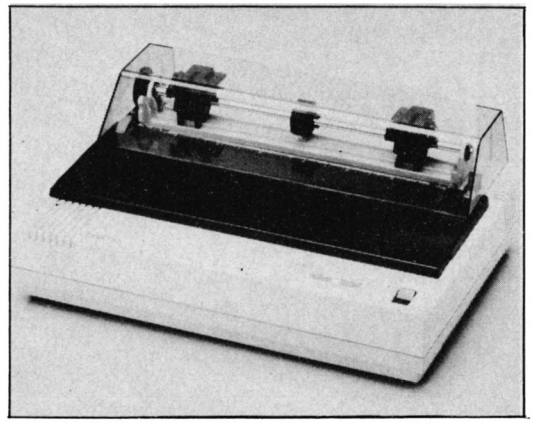
Einen Turbo-Pascal-Maskengenerator vertreibt die DATANET GmbH. Das Programm werde, so der Anbieter, durch die automatische Generierung des Quellcodes für die Dateneingabe Zeit und Fehler bei der Programmierung ersparen. Mit den erstellten Masken sind unter anderem Pop-Up-Windows, Pull-Down-Menüs und überlappende Windows möglich.
Fehleingaben werden vom Programm bemerkt und durch entsprechende Meldungen dargestellt. Zu jedem Feld der Maske beziehungsweise zur gesamten Maske lassen sich Hilfefenster erstellen. Der generierte Quelltext kann mit einem Editor nach Belieben editiert werden. Zum Maskengenerator werden drei Hilfsprogramme geliefert.

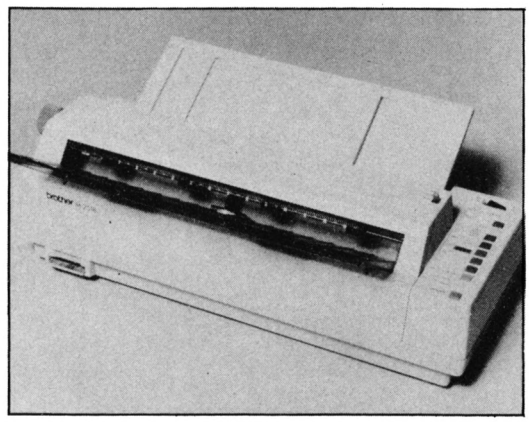
Zu beziehen bei:
DATANET Gmbh
Vogteistr. 19
5353 Mechernich
Tel. 0 24 43/66 26

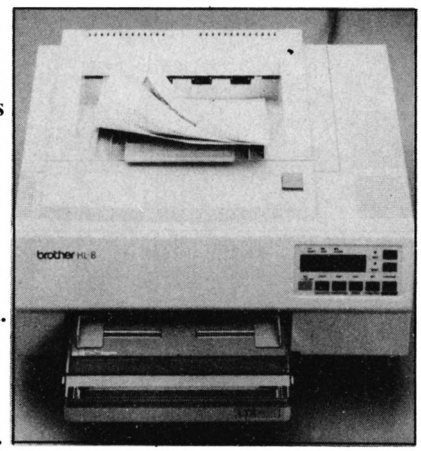
Sechs Drucker von Brother

Eine Palette von Druckern stellt Brother vor. Der M-1209 ist vor allem für semiprofessionelle und professionelle PC-Anwendungen konzipiert. Er ist mit den Schnittstellen Centronics Parallel und V24 ausgestattet und arbeitet mit einer Geschwindigkeit von 168 Zeichen pro Sekunde. Bei NQL-Schönschrift leistet er 35 Zeichen. Alle Zeichen können in doppelter Höhe und Breite ausgedruckt werden. Formulatraktor und Einzelblattzuführung gehören zur Standard-Ausrüstung. Der Preis beträgt 798 Mark.
Der 24-Nadel-Drucker M-1724L hat eine Geschwindigkeit von maximal 216 Zeichen pro Sekunde in Elite, in Schönschrift arbeitet er mit maximal 72 Zeichen. Schubtraktor und optionale Einzelblattzuführung SF-40 erlauben die Verarbeitung von Einzelblättern, Endlospapier und Formularsätzen bis zu einer Papierbreite von DIN A3 quer. Er ist mit den Schnittstellen Centronics Parallel und V.24 seriell ausgerüstet und emuliert Epson LQ-1000, IBM-Grafikdrucker, die Brother-HR-Serie und Diablo 630. Der M-1724L kostet 1995 Mark.
Mit 18 Nadeln ist der M-2518 ausgestattet. Er arbeitet im Datendruck mit 360 Zeichen pro Sekunde. NLQ wird mit 75 Zeichen in Zwei-Pass-Technik, LQ mit 50 Zeichen in Drei-Pass-Technik ausgedruckt. Schubtraktor und Papier-Parkposition sind eingebaut, die Zwei-Schacht-Einzelblattzuführung SF-300 ist optional. Der Flachbettdrucker emuliert die Epson-EX/FX-Serie und IBM Proprinter XL.
Der Farbmatrix-Printer M-3524-C druckt bis zu sieben Mischfarben. Seine Geschwindigkeit beträgt in HS Elite 432 Zeichen pro Sekunde, in Draft 324 und in LQ 108. Die Papierzuführung erfolgt in der Standard-Ausstattung über Schubtraktor und Gummiwalze. Der 24-Nadel-Drucker kostet 3500 Mark.
Der High-Speed-Nadeldrucker M-3524L arbeitet im Datendruck mit 324 Zeichen pro Sekunde und läßt sich bei größeren Datenmengen auf 432 Zeichen umstellen. Er verfügt über halbautomatische Papierzuführung für Einzelblätter und einen eingebauten Schubtraktor. Durch ein Vierfarbband läßt sich der M-3524 zum Farbdrucker umrüsten. Für universellen PC-Anschluß sorgen die Schnittstellen Centronics Parallel und V.24 seriell.
Der HL-8 „LaserAs" arbeitet mit einem Halbleiter-Laser und einer Druckgeschwindigkeit von acht Seiten pro Minute. Die Auflösung des Druckbildes beträgt 300 Punkte pro Inch. Die Papierzuführung erfolgt über eine Kassette mit 200 Blatt. Lieferbar sind zwei Modelle mit einer Speicherkapazität von 0,5 und einem MByte. Fünf Schriftarten stehen zur Verfügung. Der HL-8 ist ebenfalls mit den Schnittstellen Centronics Parallel und V.24 ausgestattet. Das 0,5-MByte-Modell kostet 6829 Mark, das Ein-MByte-Modell 7399 Mark.

Informationen bei:
Brother International GmbH
Im Rosengarten 14
6368 Bad Vilbel
Tel. 0 61 01/80 50







Oben:
Reichhaltige Ausstattung zu geringem Preis ist das Motto des M-1209.

Mitte:
Der Brother M-2518 ist als Flachbett-Drucker sogar für Postkarten und Karteikarten geeignet.

Unten:
Der Laser-Drucker HL-8 rundet die Palette nach oben ab.

Netzwerk von Torus

Mit Tapestry II hat das englische Software-Unternehmen Torus eine intelligente Netzwerk-Oberfläche mit einem integrierten Office-Automation-System entwickelt. Das Programm ist in C geschrieben und unterstützt alle wichtigen Standards wie etwa Netbios Interface und SMB.
Ein weiterer Schwerpunkt liegt bei der Netzwerk-Konfiguration, die auf Domains basiert. Eine Domain umfaßt bis zu 100 PCs. Da Tapestry II auch Internetworking unterstützt, besteht die Möglichkeit, mehrere LANs zu einem WAN zu verbinden. Die deutsche Version wird im Lauf dieses Sommers erscheinen.

Information bei
ALSO-ABC Trading GmbH
Am Stadtrand 35-37
2000 Hamburg 70
Tel. 0 40/6 94 46 01

PageMaker Version 3.0

Die dritte Version von PageMaker wird im Lauf dieses Sommers auf den Markt kommen. Sie enthält Funktionen für die effektive Erstellung von Büchern, Angeboten und anderen umfangreichen Dokumenten.
Ein automatischer Spalten- und Seitenumbruch sorgt für die schnelle Plazierung langer Dateien. Layout- und Formatierungswünsche werden berücksichtigt. Mit PageMaker 3.0 kann der Anwender eigene Schrift-Makros erstellen. Durch die Farb-Unterstützung lassen sich diverse Einfärbungen ohne Umwege erreichen. Der automatische Textumlauf lokkert Publikationen auf.
PageMaker 3.0 bietet jetzt auch zusätzliche Schriften sowohl für den HP LaserJet Plus als auch für andere Serie-II-Drucker.

Informationen bei
ALSO-ABC Trading GmbH
Am Stadtrand 35-37
2000 Hamburg 70
Tel. 0 40/6 94 46 01

MAXXUM-Software

Vom MAXXUM-Software-Verlag, eigenständiger Geschäftsbereich von ALSO ABC, kommen mehrere neue Produkte. SuperProjekt PRIMUS ist ein Projektplanungs-Programm in drei Arbeitsschritten: In der Orientierungsphase werden die Grundrisse des Projekts entwickelt, in der Realisierungsphase wird der aktuelle Stand überprüft und der Arbeitsplan an wechselnde Bedingungen angepaßt. Im dritten Schritt übernimmt das Programm die Auswertung und Anpassung von Projektplänen.
F&A PRIMUS verbindet Textverarbeitung und Datenbank innerhalb eines Programmpakets. Dadurch können Serienbriefe oder Formulare mit Adressen oder Datei-Angaben verbunden und ausgedruckt werden.
Das Kalkulations-Programm SuperCalc PRIMUS bietet Arbeitsbereiche von der Budgetplanung über Verkaufs- und Gewinnvoraussagen bis zu „Was-wäre-wenn"-Analysen. Auf Knopfdruck erstellt das Programm auch präsentationsbereite Grafiken aus den Daten des Arbeitsblattes.
SNAP ist ein System-Verwaltungs-Programm mit über 200 Funktionen. Es verfügt über den kompletten ASCII-Zeichensatz und unterschiedliche Editierfunktionen.
SNAP kostet 248 Mark, die anderen drei Programme je 398 Mark.

Informationen bei
ALSO-MAXXUM
Am Stadtrand 35-37
2000 Hamburg 70
Tel. 0 40/6 94 46 03

Freeway mit Btx

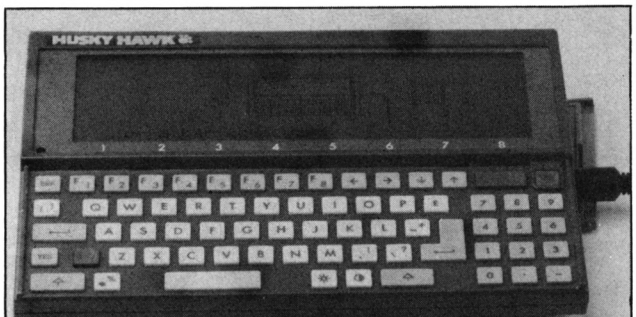
Freeway Advanced ist ein Telekommunikations-Programm mit Btx für Hayes-Kompatible und Post-Modeme mit Hintergrund-Verarbeitung. Hauptmenü und Submenüs werden in Fenstertechnik angezeigt. Der Autopilot speichert und wiederholt die vorgegebene Einwählprozedur und Datenbankabfrage. Das Programm verwaltet maximal 108 Wählmakros mit den zugehörigen Parametern.
Freeway Advanced wird ab sofort mit einer integrierten Btx-Emulation geliefert, die den Einstieg in das Telexnetz der Post ermöglicht.
Im Telefonverzeichnis werden für jeden der 108 Einträge Systemparameter und Wählsequenzen festgehalten. Durch die automatische Wahlwiederholung werden nicht zustande gekommene Verbindungen wieder angewählt.
Der Seiteneditor ermöglicht die Erstellung und Korrektur von Texten. Eingegeben werden kann menügesteuert in Freeway oder mit dem Crosstalk-Befehlssatz. Hardwarevoraussetzungen sind IBM PC/XT/AT, PS/2 und Kompatible, PC-DOS Version 2.0 und höher, mindestens 300 KByte freier RAM-Speicher sowie V.24-Schnittstelle mit externem Modem. Freeway Advanced kostet 598 Mark.

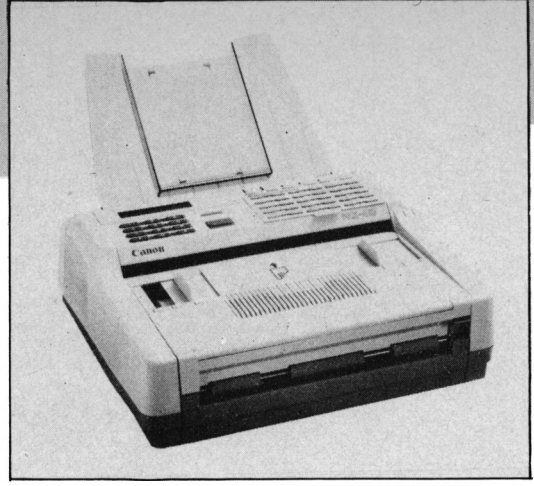
Zu beziehen von
Nord Computer
& Software GmbH
Emil-Kurz-Str. 1
8045 Ismaning/München
Tel. 0 89/96 75 72, 96 54 42, 96 45 60

HAWK 8/16 mit MS-DOS 3.21

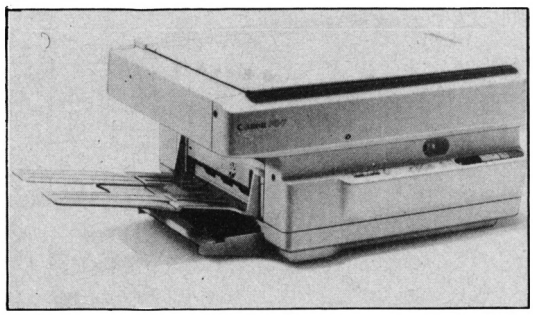
Mit dem HAWK 8/16 des englischen Herstellers HUSKY Computers ist erstmals ein 16-Bit-Hand-Held-Computer mit MS-DOS-3.21-Betriebssystem vorgestellt worden. Er bietet MS-DOS-3.21-Kompatibilität, Optokoppler, Kommunikations-Software und zwei serielle RS232C/V.24-Schnittstellen. Am Gerät, das mit dem 80C88-Prozessor läuft, können netzunabhängige Peripheriegeräte wie Printer und Modem in einer Sidebox befestigt werden.
Als Kommunikations-Basis und Ladegerät für das HUSKY-HAWK-System ist HOMEBASE konzipiert.
Ein Optokoppler macht herkömmliche Steckverbindungen überflüssig. Zusätzlich sind Interfaces für den Drucker- und Diskettenlaufwerk-Betrieb eingebaut. HOMEBASE besitzt einen eigenen Prozessor und optionale Speichererweiterungs-Möglichkeiten.

Informationen bei
Boston Computer
Handelsgesellschaft mbH
Mobile Informationssysteme
Anzinger Str. 1
8000 München 80
Tel. 0 89/49 10 73 ■





Neue Fotokopierer und Telefaxgeräte von Canon. Der Fotokopierer hat einen Zoombereich von 70 bis 122 Prozent.

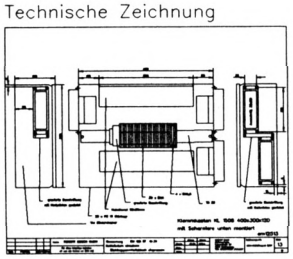


Programm für Handwerker

Mit HANDWERK-CAD 3000 können Handwerker technische Zeichnungen entwickeln und ausdrukken. Das Programm wurde, um eine einfache Handhabung zu gewährleisten, zusammen mit einer Münchner Elektro-Installations-Firma entwickelt. Es fragt nötige Angaben selbst ab. Schalter, Steckdosen und ähnliches sind in der Symbol-Bibliothek gespeichert. HANDWERK-CAD 3000 setzt das Programm „Pictures by PC" voraus. Beide Programme kosten zusammen 6950 Mark.

Informationen bei:
Heinz W. Nothofer
EDV-Beratung
Plazerstr. 2
8000 München 70
Tel. 0 89/14 12 0 01

Technische Zeichnung



System lernt lesen

Ein von der französischen Firma Inovatic entwickeltes Software-Produkt ermöglicht es dem Benutzer, dem System eine Schriftart in drei bis acht Minuten beizubringen.
Mit READSTAR-III werden der Software durch Algorithmen Objekte angeboten, die dann entweder als Buchstaben, Linien oder Bilder erkannt werden.

Information und Vertrieb:
Advantage Computer
Company GmbH
Bayernthalgürtel 15
5000 Köln 51
Tel. 02 21/38 50 18

Drei Kopierer von Canon

Drei neue Farbkopierer hat Canon entwickelt: den Tischkopierer PC 7, den Telekopierer Canon Fax 410 und den Farb-Laser-Kopierer CLC.
Modell PC 7 erreicht einen Zoom-Bereich von 70 bis 122 Prozent. Er verfügt über ein feststehendes Vorlagenglas und leistet Vergrößerungen in DIN-Schritten von A4 bis A0. Die Universal-Kassette enthält 100 Blatt, bis zu 99 Kopien können vorgewählt werden.
Der Fax 410 überträgt in zwölf Sekunden eine Seite an jedes andere kompatible Canon-Modell. Die automatische Erkennung von Halbton und Strich ermöglicht Telekopien vom Original in einem Arbeitsgang. Durch die automatische zeitversetzte Übertragung wird jede Mitteilung termingerecht durchgeführt.
Der CLC verfügt über digitale Bildverarbeitung. Durch diese Technologie lassen sich auch feinste Linien und Zeichen originalgetreu wiedergeben. Der CLC vergrößert bis zu einem Format von doppelt DIN A0. Das Kopieren mit vier Tonerfarben erlaubt sowohl vollfarbige Kopien und Farbabstimmungen als auch Umfärben in andere Farben.
Schlagzeilen hatte der Farb-Laser-Kopierer gemacht, als die mit ihm gefertigte Kopie eines sogenannten fälschungssicheren Personalausweises nicht auf den ersten Blick als solche zu erkennen war. Zum Ausschluß von Mißbrauch kann der CLC nur mit einem Schlüssel gestartet werden.

Hilfe bei dBASE III +

Ein neues Computer-Lernprogramm von Individual Software verspricht dem Benutzer Hilfe bei der Verwendung von dBASE III+. Das „Training für dBASE III+" bietet interaktive Lektionen für die praktische Erfahrung mit Befehlen unter Verwendung des Assistenten und des Punkt-Prompt. Das Lernprogramm zeigt, wie Datenbank Dateien entweder mit Hilfe des Assistenten oder durch direkte Arbeit vom dBASE-Punkt-Prompt erstellt werden. Es beginnt mit einer Einführung in Datenbank-Konzepte und die Terminologie und will damit auch Nicht-Programmierern ermöglichen, schnell mit dBASE III+ arbeiten zu können.
Das Programm ist passend für IBM PC/AT/XT und führende PC-Kompatible. Minimum sind 256 KByte Benutzerspeicher, DOS ab Version 2.0, ein Diskettenlaufwerk und ein Farb- oder Monochrombildschirm.
Der Preis beträgt 275 Mark.

Weitere Informationen bei:
TECPRO GmbH
Augsburger Str. 6
8890 Aichach

Manual für DOS

Ein Online-Referenzmanual für DOS liegt mit dem Individual-Software-Produkt SmartGuide For DOS vor. Das speicherresidente Programm bietet Hilfe bei der Eingabe und Ausführung von DOS-Befehlen, erklärt schwierige Themen und hält eingebaute Dienstleistungen bereit. SmartGuide For DOS behandelt Themen von Syntax über grundlegende Befehle bis hin zu einem Zeileneditor. Alle Befehle von DOS 2.1 bis 3.3 werden abgedeckt.
Das Programm kostet 99 Mark. Es ist für IBM PC/XT/AT sowie PS/2 und Kompatible erhältlich. Mindestanforderung sind 256 KByte Benutzerspeicher, DOS ab Version 2.0, ein Diskettenlaufwerk und ein 80-Zeilen-Bildschirm.

Weitere Informationen bei:
TECPRO GmbH
Augsburger Str. 6
8890 Aichach ■

Speicherstellen und CALLs

Die BASIC-Trickkiste

Nützliche Routinen

Mit einigen Kenntnissen über den internen Aufbau der CPCs können Sie BASIC noch besser nutzen. Wir haben für Sie das Innenleben dieses Computers untersucht.

Wer kennt das nicht: Die Aliens rotieren wild, die Bomben fliegen tief, der Raum leuchtet magisch im Licht der Laserkanonen. Und da – ein Schuß, ein Schrei, der Highscore ist im Eimer.
Da kommt mancher auf die Idee, das BASIC-Programm nach den Variablen für die Leben oder die Laser-Energie abzusuchen. Voller Spannung drücken Sie die ESC-Taste. Und siehe da, es funktioniert. Das Programm ist unterbrochen.
Aber anstatt eines freundlichen „Ready" erscheint nur ein Pixel-Muster, dessen Länge ungefähr der gewohnten Meldung entsprechen könnte. Außerdem ist auf dem Bildschirm kaum etwas zu erkennen, da die INKs in allen Regenbogenfarben schillern.

Neu-Initialisierungen mit BASIC

Das Programm hat die Symboltabelle, die INKs und womöglich noch Belegung und Wiederholfrequenz der Tasten verändert. In solchen Fällen ist es meist sehr umständlich, den Computer wieder in seinen Standard-Zustand zu versetzen, ohne das gerade geladene Programm zu löschen.
Doch da kann Abhilfe geschaffen werden. Es gibt nämlich in der Sprungtabelle des CPC fünf Routinen, die Sie mit einem einfachen CALL-Befehl aufrufen können. Sie heißen:

```
KM  INITIALISE – &BB00
TXT INITIALISE – &BB4E
SCR INITIALISE – &BBFF
GRA INITIALISE – &BBBA
SOUND RESET    – &BCA7
```

Was diese Routinen bewirken, soll im folgenden erklärt werden.
Die Routine KM INITIALISE ist eines der Firmware-Programme, die zusammen den Keyboard Manager bilden – daher die Bezeichnung KM. Ihr Aufruf bewirkt, daß der Tastatur-Puffer geleert und neu aufgebaut wird. Das ist von Vorteil, wenn Sie in einem Programm die Tastatur abfragen wollen: Es wird vermieden, daß sich Zeichen im Puffer befinden, die Sie irgendwann während des Programmlaufs getippt haben.
Zusätzlich werden durch diese Routine die gesamte Tastenbelegung wieder auf die Standard-Werte gesetzt und die Reaktionszeiten des Tastatur-Treibers zurückgesetzt. Die internen Flags für SHIFT, CAPS LOCK und SHIFT LOCK werden auf „aus" gesetzt.

Textausgaben

Die Routine TXT INITIALISE bewirkt, daß die vom Anwender definierte Symboltabelle zurückgesetzt wird. Alle Zeichenmatrizen werden also wieder aus dem ROM gelesen.
Alle Windows werden auf volle Bildschirmgröße gesetzt, als akti-

Bit Nr.	7	6	5	4	3	2	1	0
MODE 0	A1	B1	A3	B3	A2	B2	A4	B4
MODE 1	A1	B1	C1	D1	A2	B2	C2	D2
MODE 2	A1	B1	C1	D1	E1	F1	G1	H1

Nach diesem Schema können Sie eine Linienmaske erstellen

Die Speicherstellen und Firmware-Einsprünge, die in dieser Folge vorgestellt wurden

POKEs:

&B1C8 (&B7C3) : MODE-Nummer
&B28F (&B72F) : Maske für Text-PEN
&B290 (&B730) : Maske für Text-PAPER
&B338 (&B6A3) : Maske für Grafik-PEN
&B339 (&B6A4) : Maske für Grafik-PAPER
(Werte in Klammern gelten für CPC 664/6128)

CALLs:

&BB00 : Tastatur initialisieren
&BB4E : Text-VDU (Video Display Unit) initialisieren
&BBFF : Bildschirm-Paket initialisieren
&BBBA : Grafik-VDU initialisieren
&BCA7 : Sound-Routinen initialisieren

Die Reset-Routinen der CPC-Firmware

KM INITIALISE (&BB00)
Tastaturpuffer löschen
Tastenbelegung zurücksetzen
Geschwindigkeiten zurücksetzen
CAPS LOCK, SHIFT und SHIFT LOCK löschen

TXT INITIALISE (&BB4E)
Symbole zurücksetzen
Windows zurücksetzen
Window #0 wählen
PAPER 0 und PEN 1 setzen
Opaque-Modus einschalten
Cursor nach oben links bewegen

SCR INITIALISE (&BBFF)
Farben zurücksetzen
Blinkperioden zurücksetzen
MODE 1 einschalten
Bildschirm löschen
*** Achtung! Nur in MODE 1 verwenden! ***

GRA INITIALISE (&BBBA)
ORIGIN 0,0 ausführen
Grafik-PAPER auf 0 setzen
Grafik-PEN auf 1 setzen
Grafik-Fenster zurücksetzen

SOUND RESET (&BCA7)
Tonwarteschlange löschen
Laufende Töne abschalten

ves Window wird Fenster #0 ausgewählt. Als PAPER wird 0, als PEN 1 gesetzt; der Cursor befindet sich nach dem Aufruf dieser Routine in der linken oberen Ecke des Bildschirms. Der Schirm wird jedoch nicht gelöscht.
Die letzte Tätigkeit von TXT INITIALISE ist das Rücksetzen des Textmodus auf den normalen Opaque-Mode. Ein vorher ausgeführter Befehl

PRINT CHR$(22);CHR$(1);

wird dadurch rückgängig gemacht.

Das Bildschirm-Paket

SCR INITIALISE entstammt dem Paket der Bildschirm-Routinen (Screen). Ein

CALL &BBFF

veranlaßt, daß alle INKs und Blinkperioden auf ihre Standard-Werte zurückgesetzt werden. Der Bildschirm wird dabei mit Farbe Null gelöscht.
Sie sollten diese Routine nur in MODE 1 aufrufen, da sie die interne Speicherstelle für den Bildschirm-MODE auf 1 setzt. In den MODEs 2 und 0 hat dies unvorhersehbare Folgen für die Bildschirm-Darstellung.

Auch für Grafik

Eine Routine wie SCR INITIALISE gibt es auch für die Grafik. Sie heißt GRA INITIALISE und erledigt folgende Aufgaben:
Der Grafik-Ursprung wird auf die Bildschirm-Koordinaten (0;0) zurückgesetzt. Ähnlich wie beim Text-Reset werden Grafik-PAPER und -PEN auf 0 beziehungsweise 1 gesetzt.
Der Grafik-Cursor steht nach dem Aufruf dieser Routine an der Stelle (0;0), das Grafik-Fenster beansprucht den ganzen Bildschirm. Ein bereits bestehendes Bild wird nicht gelöscht.

Die Routine SOUND RESET hat nur wenige Funktionen. Sie löscht alle Tonwarteschlangen und schaltet jeden laufenden Ton ab. Der CPC schweigt.
Eine Kurzübersicht über alle Reset-Routinen finden Sie in Abbildung 1. Sie sollten sich die Liste griffbereit neben Ihren CPC legen. Wer weiß, wann Sie sie benötigen.

Der Trick mit dem MODE

Der CPC arbeitet in drei verschiedenen MODEs, in denen er 20, 40 oder 80 Zeichen pro Zeile darstellen kann. Mancher will aber in seinen selbstgeschriebenen Programmen auch eine größere oder kleinere Schrift darstellen, als es der gerade gewählte MODE erlaubt.
Hier können Sie mit einigen Tricks ein wenig manipulieren. Im Betriebssystem-RAM gibt es eine Speicherstelle, mit der der CPC den Bildschirm-Modus verwaltet.

Sie befindet sich beim CPC 464 an der Adresse &B1C8, bei den anderen CPCs an der Adresse &B7C3.

Modus-Änderung

Wenn Sie in diese Speicherstelle einen anderen MODE hineinPOKEn, geschieht folgendes:
Der CPC benutzt ab sofort die Bitmasken für den neuen MODE. Alle Zeichen werden also in der entsprechenden Größe dargestellt. Doch gibt es hierbei gewisse Probleme. Wenn Sie nämlich einen Modus wählen, der weniger Zeichen als der ursprüngliche darstellen kann, werden die Zeichen in der Mitte aufgeteilt. Zwischen den beiden Hälften entsteht eine häßliche Lücke. Versuchen Sie's doch mal auf Ihrem CPC.
Schalten Sie hingegen in einen MODE, der mehr Zeichen darstellen kann als der zuvor eingestellte, so werden nur die ersten vier Bits eines jeden Buchstabens dargestellt. Sie sehen also nur die linke Hälfte des Zeichens.
Beim Umschalten in einen höheren MODE müssen Sie zudem beachten, daß nur noch der halbe Bildschirm-Bereich genutzt wird. Beim Schalten in einen niedrigeren Modus paßt nur noch die Hälfte der Zeichen in eine Zeile – obwohl hier die ganze Breite genutzt wird. Sorgen Sie also dafür, daß die auszugebenden Strings die richtige Länge haben.
Bei all diesen Nachteilen könnte man denken, daß sich ein Experimentieren mit diesen Möglichkeiten kaum lohnt. Dennoch können Sie mit der MODE-Umschaltung schöne Effekte erzeugen, vor allem, wenn Sie sie mit anderen Tricks, wie etwa den nun folgenden, kombinieren.
Der CPC kann – je nach Modus – mehrere Farben darstellen, die Sie jedem Zeichen einzeln zuordnen können. Die Frage liegt nahe, ob dies nicht auch für einzelne Pixel eines Zeichens möglich ist.
Im RAM des CPC gibt es tatsäch-

```
1 'Maskenwahl                                  <110F>
2 '                                            <070F>
10 CLS                                         <063E>
20 WINDOW 1,80,1,25:WINDOW#1,1,80,
6,25                                           <1DD5>
30 FOR i=1 TO 5                                <0E85>
40 CLS#1                                       <084C>
50 FOR o=(i-1)*51 TO i*51                      <1EA6>
51 POKE &B6A3,255'(fuer CPC 464: P
OKE &B338,...)                                 <2C63>
52 MOVE 30,(o-(i-1)*51)*6                      <1DFC>
53 DRAWR-10,0                                  <0BCF>
54 MOVER 10,0                                  <0AAD>
55 POKE &B6A3,z MOD 256'(fuer CPC
464: POKE &B338,...)                           <3200>
56 DRAWR 640,0                                 <0B6B>
57 z=z+1                                       <1026>
60 NEXT o                                      <0A00>
70 GOSUB 130                                   <096C>
80 NEXT i                                      <0A10>
90 RUN                                         <06DF>
100 '                                          <07D3>
110 ' Unterprogramm 1                          <17A9>
120 '                                          <07FB>
130 y=25:GOTO 200                              <1100>
140 a$=INKEY$:IF a$=""THEN 140                 <1908>
150 IF ASC(a$)=240 THEN IF y>50 TH
EN y=0 ELSE y=y+1                              <2E2D>
160 IF ASC(a$)=241 THEN IF y<1 THE
N y=51 ELSE y=y-1                              <2E90>
170 IF ASC(a$)=224 THEN GOSUB 280              <1691>
180 IF ASC(a$)=32 THEN RETURN                  <13B5>
190 IF a$<>CHR$(240)AND a$<>CHR$(2
41)THEN 140                                    <218C>
200 MOVE 0,y2*6:DRAWR 15,0,0:DRAWR
-5,4:DRAWR 5,-4:DRAWR-5,-4                     <2A20>
210 MOVE 0,y*6:DRAWR 15,0,1:DRAWR-
5,4:DRAWR 5,-4:DRAWR-5,-4                      <2942>
220 LOCATE 1,1:PRINT"Masken-Code:"
;((i-1)*52+y)MOD 256:PRINT CHR$(10
)CHR$(18)                                      <3E24>
230 y2=y                                       <0FCB>
240 GOTO 140                                   <0932>
250 '                                          <07FE>
260 ' Unterprogramm 2                          <17D9>
270 '                                          <0728>
280 mc=((i-1)*52+y)MOD 256                     <21A1>
290 LOCATE 1,1:PRINT"Masken-Code:"
;mc;" ":LOCATE 1,3:PRINT"Masken-Ke
y: &X";BIN$(mc,8);                             <472E>
300 RETURN                                     <0680>
```

lich zwei Speicherstellen, mit deren Hilfe Sie Zeichen zumindest in Längsrichtung mit einer Farbmaske versehen können. Für die Maskierung des PENs sind dies die Adressen &B28F für den 464 und &B72F für den 664 und 6128; für PAPER gelten die Werte &B290 (CPC 464) beziehungsweise &B730 (CPC 664 und 6128).
Der Ausdruck Maskieren bedeutet, daß ein Zeichen – ein Bitmuster also – mit einem anderen Bitmuster verknüpft wird. Über das Zeichen wird sozusagen eine Schlitzmaske mit verschiedenfarbigen Schlitzen gelegt. Am besten experimentieren Sie ein wenig und sehen sich die Ergebnisse an. Eine solche Maske können Sie sowohl für die dargestellten Zeichen (PEN) als auch für deren Hintergrund (PAPER) verwenden.
Da Sie mit höchstens acht Bits maskieren können – mehr paßt eben nicht in ein Byte – können Sie nur in MODE 2 das ganze Zeichen erfassen. Dafür haben Sie hier nur die Auswahl zwischen weiß und schwarz oder was immer Sie als Vorder- und Hintergrundfarben eingestellt haben.
In MODE 1 können Sie bereits vier Farben verwenden. Für eine Farbe benötigen Sie dann aber schon zwei Bits, und die Maske reicht nur noch für die ersten vier Reihen des Zeichens. Deshalb verdoppelt sie der CPC in Querrichtung und maskiert die erste und die zweite Hälfte gleich.
Für den MODE 0 gilt fast dasselbe, mit der Ausnahme, daß Sie für eine Farbe vier Bits benötigen. Die Maske ist daher nur noch für zwei Reihen ausreichend, so daß sie der CPC vervierfacht und jedes Viertel des Zeichens mit ihr verknüpft.
Ein Acht-Bit-Muster, wie Sie es für die Maske verwenden müssen, ist im Dezimal-System eine Zahl zwischen Null und 255. Sie können mit einem POKE-Befehl eine solche Zahl in eine der oben genannten Adressen schreiben.
Auf dem CPC können Sie auch die Binärschreibweise verwenden, wenn Sie die Zeichenfolge &X voranstellen. In MODE 2 ist es so relativ einfach, ein Zeichen zu maskieren. Bei dem Befehl

```
POKE &B28F,&X01110101
```

bedeutet jede Null, daß die Spalte mit der Farbe Schwarz, jede Eins, daß sie mit der Farbe Weiß aufgefüllt wird.
In MODE 1 wird das Ganze schon etwas schwieriger. Um die vier Farben darzustellen, benötigen Sie pro maskiertem Pixel zwei Bits.
Übertragen Sie dazu die Nummer der gewünschen Farbe ins Binär-System und zerlegen Sie sie in die zwei einzelnen Bits. Wenn etwa zuerst ein Streifen der Farbe 3, dann einer mit Farbe 2, einer mit Farbe 1 und schließlich einer mit Farbe 0 erscheinen soll, müssen Sie so vorgehen:

```
Farbmaske: 3210
3 –> &X11 –> Bit A1=1,
             Bit A2=1
2 –> &X10 –> Bit B1=1,
             Bit B2=0
1 –> &X01 –> Bit C1=0,
             Bit C2=1
0 –> &X00 –> Bit D1=0,
             Bit D2=0
```

Suchen Sie anschließend einfach in unserer Abbildung 2 die Zeile für MODE 1 und setzen Sie die Maske zu einer Acht-Bit-Zahl zusammen. Sie erhalten:

```
&X11001010
```

oder dezimal 202. Ob Sie die Binärschreibweise direkt verwenden oder Ihren CPC zuerst ins Dezimal-System umrechnen lassen, ist Ihre Sache.
POKEn Sie doch die Zahl 202 in die Speicheradresse für die Hintergrund-Maske. Die Ready-Meldung sieht gleich viel schöner aus.
Die Vorgehensweise für MODE 0 ist dieselbe. Sie haben hier lediglich mit Vier-Bit-Werten zu tun. Doch das dürfte Ihnen keine Schwierigkeiten bereiten.
Dieses Beispiel zeigt, daß man dem CPC doch mehr Farben entlocken kann, als auf Anhieb möglich scheint. Doch der Trick funktioniert nicht nur im Text-Modus. Jeder, der auf seinem CPC-Bildschirm einmal mehrere Kurven gleichzeitig darstellen wollte, kennt das Problem: Wählt man wegen der besseren Auflösung den MODE 2 oder zur besseren farblichen Unterscheidung der Funktionen den MODE 1?
Die Lösung ist einfach: Maskieren Sie die Grafik-Linien, und Sie können auch im hochauflösenden Modus Farbunterschiede darstellen. Wenn Sie nämlich eine Punktmaske benutzen, die nur jeden zweiten Pixel darstellt, so erscheint die Linie in MODE 2 dunkler als eine, bei der jeder Punkt dargestellt ist. Die Lücken, die eigentlich vorhanden sein müßten, sind aufgrund der hohen Pixel-Dichte in Modus 2 nicht zu erkennen.
Beim Zeichnen von maskierten Grafiken müssen Sie darauf achten, daß Sie nur den DRAW-Befehl verwenden. Der PLOT-Befehl berücksichtigt die Maske nicht, die Sie in die Speicherstellen &B338 (CPC 464) oder &B6A3 (CPC 664 und 6128) gePOKEd haben.
Unser Listing zeigt ein kleines Programm, mit dem Sie sich alle verfügbaren Linienmasken ansehen können. Aktivieren Sie den gewünschten Modus und starten Sie das Programm. Mit den Cursortasten können Sie den Pfeil steuern. Der Masken-Code, der jeweils gePOKEd werden muß, wird links oben angezeigt. Ein Druck auf COPY liefert Ihnen zusätzlich die Binär-Darstellung.
In Abbildung 3 finden Sie eine Übersicht der POKEs und CALLs, die wir in dieser Folge der CPC-Trickkiste vorgestellt haben.

Jochen Weilandt/AE □

Neue BASIC-Dimensionen

Der Archimedes zeigt, was Sache ist.

CPC-Besitzer sind ein komfortables BASIC gewohnt. Doch Superrechner Archimedes beweist, daß diese Sprache noch anspruchsvoller werden kann. Durch die Geschwindigkeit des schnellen RISC-Prozessors ist sein BASIC sogar noch den compilierten Sprachen manch anderer Computer überlegen.

Die Leistungen des Archimedes gelten als so enorm, daß die Gerüchteküche humorvolle bis abenteuerliche Meinungen hervorbrachte. Zum einen hieß es, der BASIC-Interpreter kenne keinen CLS-Befehl, sondern müsse über komplizierte VDU-Befehle angesprochen werden, andere Berichte gipfelten in der Vermutung, hier sei ein billiger Zeileneditor am Werk. Um diese Unklarheiten zu beseitigen, haben wir das BBC-BASIC V einem ausführlichen Test unterzogen.

Die Falschmeldungen rühren wahrscheinlich daher, daß die BBC-BASIC-Version V die neue Dimension einer Programmiersprache ist. Die wesentlichen Merkmale sind: leichte Erlernbarkeit bei umfangreicher Befehlsauswahl, alle Möglichkeiten zur strukturierten Programmierung, Bibliotheksfunktionen und dazu eine unglaublich schnelle Ausführungszeit. Für das erstaunliche Tempo der Abarbeitung sind die Entwickler des BASIC allerdings nur zu einem Teil verantwortlich. Das meiste Lob muß dem RISC-Prozessor zukommen, dessen Geschwindigkeit manchen Programmierer dazu brachte, Warteschleifen in das Listing einzubauen.

Funktionstasten am Keyboard: Das „Dirigentenpult" mit der Belegung der Funktionstasten

Die Auswahl dieser Programmiersprache und die Idee, den Interpreter in einem ROM unterzubringen, war eigentlich schon überholt, bevor Acorn zeigen wollte, wie es gemacht wird. Neuere Rechner der Konkurrenz, wie der Amiga, der Atari ST oder die MS-DOS-Maschinen, liefern BASIC auf einer Diskette mit, die erst einzuladen ist, bevor ein Programm gestartet werden kann.

Das Desinteresse bei Commodore und Atari an BASIC ist durchaus erklärbar. Je besser und umfangreicher der Befehlssatz dieser Interpreter wurde, desto langsamer wurde auch die Programmausführung. Sie machte einfach keinen Spaß, da halfen auch

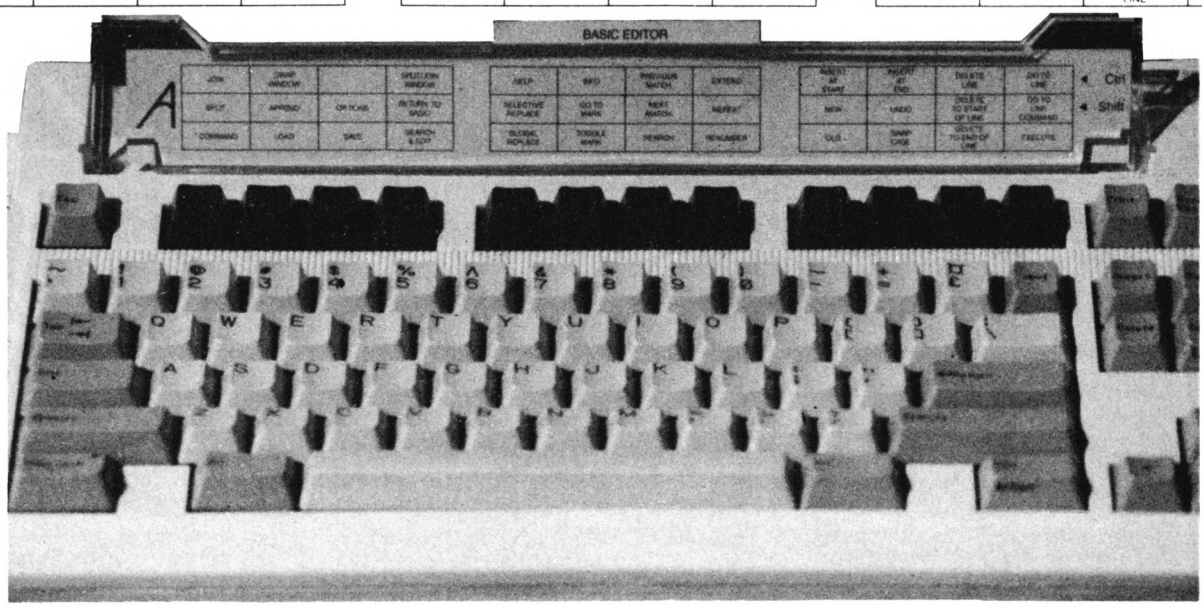
BASIC EDITOR

A

JOIN	SWAP WINDOW		SPLIT/JOIN WINDOW
SPLIT	APPEND	OPTIONS	RETURN TO BASIC
* COMMAND	LOAD	SAVE	SEARCH & EDIT

HELP	INFO	PREVIOUS MATCH	EXTEND
SELECTIVE REPLACE	GO TO MARK	NEXT MATCH	REPEAT
GLOBAL REPLACE	TOGGLE MARK	SEARCH	RENUMBER

INSERT AT START	INSERT AT END	DELETE LINE	GO TO LINE	◄ Ctrl
NEW	UNDO	DELETE TO START OF LINE	GO TO LINE COMMAND	◄ Shift
OLD	SWAP CASE	DELETE TO END OF LINE	EXECUTE	



die relativ schnellen Prozessoren der 68000er-Serie nichts. Die Anwender zogen Pascal oder C dem lahmen Interpreter vor und nahmen dafür Zusatzkosten und erhöhten Lernaufwand in Kauf.
Erst mit RISC konnten wieder hohe Geschwindigkeiten erreicht werden. Es gibt keinen Grund mehr, auf die Vorteile zu verzichten, die BASIC zweifelsohne hat. Erst recht, wenn es noch um einige interessante Optionen erweitert wurde.

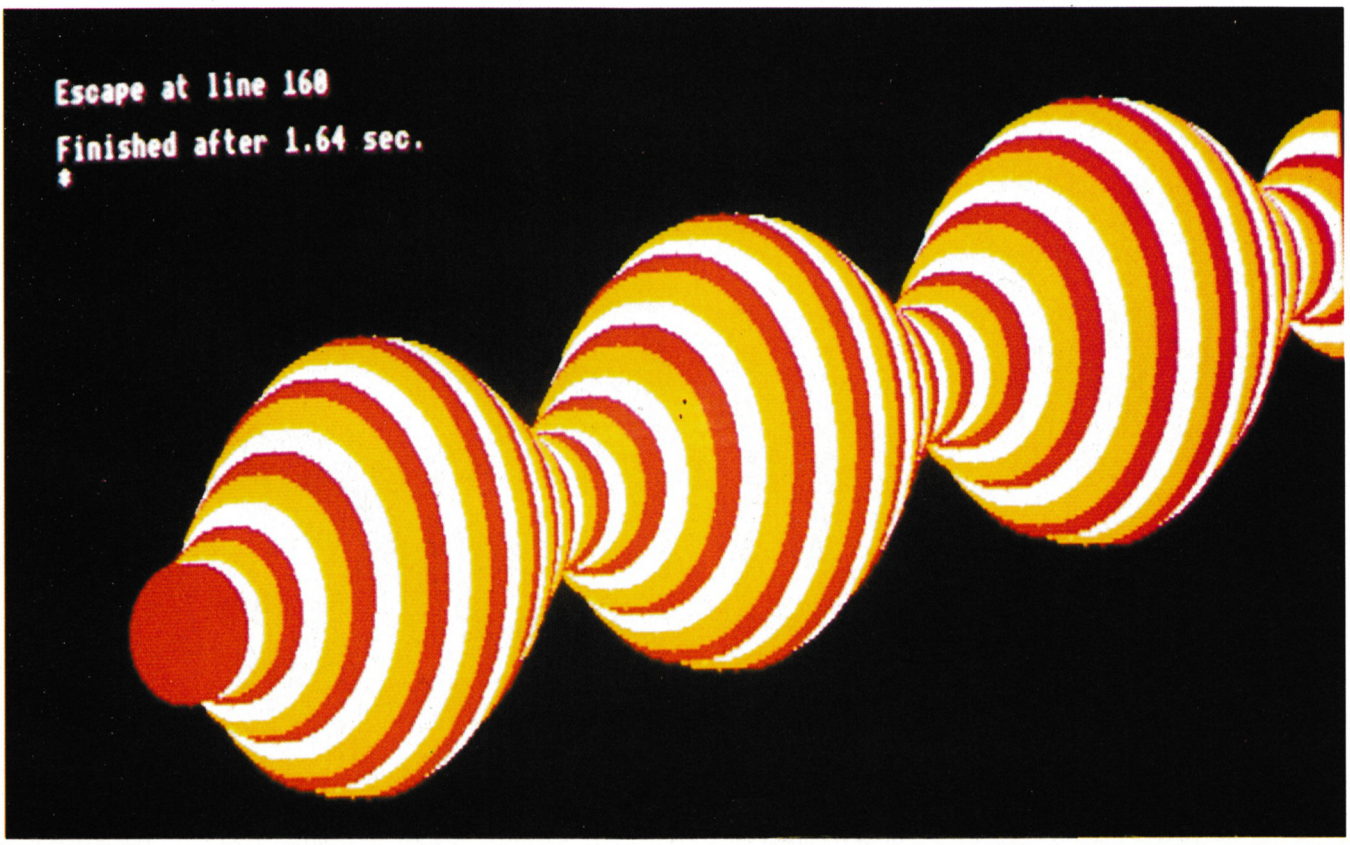
Spärlicher Direktmodus

Beim ersten Versuch, mit Archimedes-BASIC zu arbeiten, dürften oberflächliche Benutzer allerdings enttäuscht sein. Unbeeinflußt von „neumodischen" Entwicklungen, etwa Windows oder Menüzeilen, ist der Bildschirm eintönig leer, der Rechner wartet mit einem Prompt auf die Befehlseingabe. Im Direktmodus verhält sich BBC-BASIC V etwas antiquiert, etwa vergleichbar dem Locomotive-BASIC der CPC-Modelle von Amstrad.
Die Kargheit ist durchaus berechtigt. Der Archimedes macht einen großen Unterschied zwischen Direkt- und Editiermodus. Der Direktmodus dient bestenfalls dazu, einige Formeln, die nicht zu einem Programm gehören, abarbeiten zu lassen, oder, falls man es im Desktop versäumt hatte, ein Programm zu starten. Zwar kann auch hier im vollen Befehlsumfang programmiert werden, doch ist die Arbeit recht umständlich. Zeilennummern müssen eingegeben werden und zum Editieren steht nur die Copy-Cursor-Methode zur Verfügung. Weil dies so umständlich ist, wird wohl jeder den Editiermodus aufrufen, der nach der Eingabe EDIT zur Verfügung steht.

Editieren mit Dirigentenpult

Hier kann der Archimedes zeigen, was er kann. Ein vorhandenes Programm wird sofort aufgelistet und der Anwender verfügt über einen Bildschirmeditor. Mit den Cursortasten kann jede Stelle des Listings angesteuert und Ergänzungen oder Korrekturen vorgenommen werden. Unbeeinflußt von den in einem vorhergehenden Programmlauf initialisierten Farben oder Bildschirmmodi, folgt die Darstellungsform im Editiermodus immer in der Zeichenbreite, die der Anwender für den Editiermodus eingestellt hat. Es kann also nicht mehr passieren, daß irgendeine Software die Farben für Hinter- und Vordergrund schwarz eingestellt und daran nichts mehr geändert werden kann, weil die Zeichenausgabe Schwarz auf Schwarz erfolgt und der Programmierer sein Listing nicht mehr sehen kann.
Der Editor verhält sich wie ein Textverarbeitungsprogramm, dessen Optionen über die Funktionstasten gesteuert werden. Die Maus hat beim BBC-BASIC-Interpreter keinerlei Funktion, es gibt auch keine Pulldown-Menüs oder Fenster. Damit es nicht zu Mißverständnissen kommt: Maussteuerung, Fenster und Menüs können programmiert werden und sind in BASIC-Programmen aufrufbar, der Editor kommt allerdings ohne sie aus.
Zur komfortablen Bedienung sind die zwölf Funktionstasten vorgesehen, die, mit einer Ausnahme, dreifach belegt wurden. Damit der Programmierer nicht den Überblick verliert, gibt es an der Tastatur des Archimedes eine Plexiglashülle, in die ein Belegungsplan eingeschoben ist. Die Gedächtnisstütze aus Plastik läßt sich aufklappen wie ein Dirigentenpult und kann auch bei anderen Programmen genutzt werden, indem das eingeschobene Papier ausgewechselt wird.
Insgesamt sind 35 Befehle zur vereinfachten Bedienung des Editors aufge-

Schade, daß sich Geschwindigkeit nicht fotografieren läßt, ...
... aber Grafikleistung ist beweisbar.

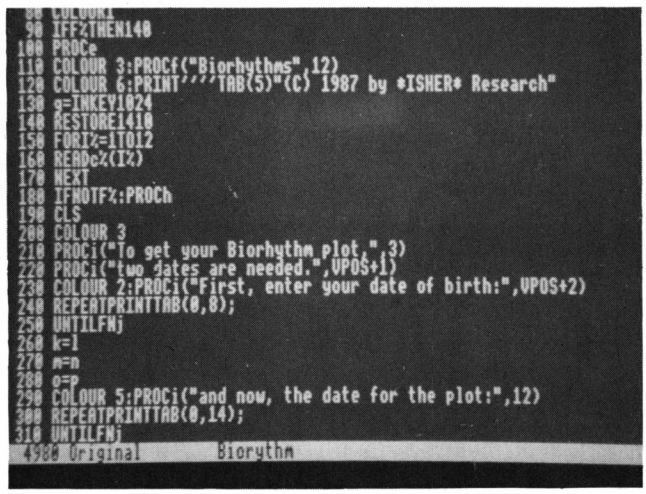


zeichnet. Sie alle aufzuzählen, hat wenig Sinn. Tatsache ist, daß vergleichbare Bequemlichkeit auch nicht beim Amiga, dem bisherigen Spitzenkönner der BASIC-Dialekte geboten werden.
Einige Beispiele der Funktionstasten sollen dies belegen:

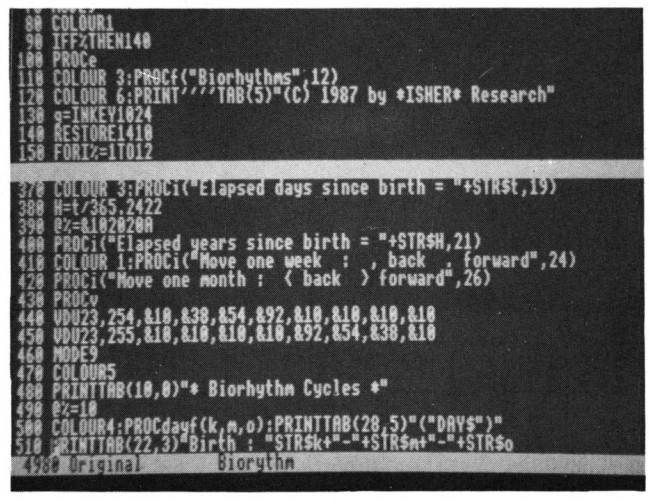
- **Automatische Zeilennumerierung,**
- **Textstellen im Listing suchen,**
- **Suchen und Ersetzen,**
- **Zeilen markieren und suchen,**
- **Statistisches Info,**
- **Help-Funktionen.**

Auch Windows sind möglich, wenngleich sie nicht die üblichen Funktionen haben. Beim Archimedes wird damit der Bildschirm des BASIC-Editors in zwei Bilschirmbereiche aufgeteilt, in denen dasselbe Listing bearbeitet wird. An der Größe oder der Lage ist nichts zu ändern, die Windows haben auch einen ganz anderen Zweck, als den der Optik. Die veränderungen, die man in einem Fenster vornimmt, wirken auf ein- und dasselbe Programm. Der Sinn ist, daß man an zwei verschiedenen Stellen des Programmes arbeiten kann, etwa in einer Menüroutine und in einem Zielbereich. Die Optionen der Funktionstasten gelten im gerade aktivierten Fenster, bearbeiten jedoch, etwa beim Suchen, das komplette Listing.
Der BBC-BASIC-Editor bietet weniger für das Auge, ist vielmehr auf eine funktionelle Bequemlichkeit ausgerichtet. Nach einiger Zeit der Eingewöhnung kann leicht auf die Maus verzichtet werden, die beim Programmieren ohnehin noch nie die Geltung besaß, die sie in einem Desktop hat.
Über 150 Befehle umfaßt das BBC-BASIC im Klartext. Dazu kommen noch einige VDU-Befehle. Diese Gruppe wirkt direkt auf den Video-Controller und kann deshalb grafische Befehle wesentlich schneller ausführen. Allerdings ist dazu die Benutzung des Handbuchs notwendig, da sonst niemand weiß, was sich hinter den Parametern verbirgt. Auf diese Weise lassen sich Windows definieren und Zeichen neu gestalten. Wem es beliebt, der kann mit einem VDU-Befehl auch den Bildschirm löschen, überflüssigerweise, denn selbstverständlich enthält der normale Sprachschatz auch den Befehl CLS.
Auf grafische Befehle wurde bei Acorn besonders großen Wert gelegt. Viele Optionen, die mit VDU-Parametern gesteuert werden können, finden sich deshalb auch unter BASIC: FILL und CIRCLE sind selbstverständlich, wobei es für die Ellipsenform noch einen Extra-Befehl (ELIPSE) gibt, um die CIRCLE-Routine nicht allzusehr mit Parametern zu belasten und damit die Geschwindigkeit zu mindern.
Um gezeichnete Objekte zu füllen, stehen nicht nur die Grundfarben zur Verfügung, sondern auch eine ganze Reihe vorgefertigter Muster, sogenannte Patterns. Auch diese Grafiken können nach den Wünschen des Anwenders frei gestaltet werden.
Besonderes Augenmerk sollte jeder den Kreis- und Kurvenbefehlen widmen. Durch die entsprechenden Parameter kann auch ein Sektor gezeichnet werden, oder man skizziert eine Kurvenlinie auf dem Bildschirm. Dies erspart große Berechnungen, wie sie zum Beispiel bei einem Kreisdiagramm vorkommen. Die Programmierung einer Geschäftsgrafik dürfte beim Archimedes zu den leichtesten Aufgaben zählen.
Die größte Überraschung sind sicherlich die zwingend notwendigen Zeilennummern. In Zeiten, in denen sich BASIC immer mehr zu einer Sprache entwickelt, die mit Labels und Subroutinen arbeitet, erscheint diese Methode auf den ersten Blick antiquiert. Aber ein Blick ins Handbuch oder das Studium bereits vorhandener Programme, macht die Befürchtungen hinfällig.
Zeilennummern müssen sein, der Archimedes akzeptiert keine anderen Programmierungsarten. Aber es können Prozeduren entwickelt werden, dessen Namen als Sprungziel verwendet werden. BBC-BASIC ist, wenn man es so nennen will, ein vollwertiges Label-BASIC. An die Prozeduren können auch Parameter oder Variablen übergeben werden, innerhalb der Routine bestehen lokale Variablen und für das ganze Listing gelten globale Deklarationen. Wem dies alles bekannt vorkommt und wer dabei an Pascal denkt, der hat durchaus Recht. Bis auf die Zeilennummern und die Tatsache, daß der Quellcode nicht compiliert werden muß, ähnelt BBC-BASIC einem stark erweiterten und gleichzeitig wesentlich vereinfachten Pascal.
Dies schließt natürlich auch den Ge-

Listing im Editor
Der Editiermodus mit einem...



...und mit zwei Fenstern



brauch von Bibliotheken ein. Unterprogramme können von Diskette eingeladen und abgearbeitet werden, nicht nur durch den CHAIN-MERGE-Befehl, sondern durch eine echte Bibliotheksverwaltung, wie sie zum Beispiel bei C üblich ist.
Wozu überhaupt noch Zeilennummern, mag man da fragen. Strukturierte Programmierung ist nicht jedermanns Sache und es ist auch nicht einzusehen, warum ein sehr kurzes Listing unbedingt in Prozeduren verzweigen soll, deren Labels extra aufgerufen werden müssen. Für solche Fälle steht immer noch der GOTO-Befehl zur Verfügung, mit dem eine Zeilennummer angesprungen werden kann. Längere Listings werden dadurch zwar unübersichtlich, doch bei kleineren Programmen ist dies der einfachste Weg der Programmierung. Dies als Tip für die Anwender.
Die Entwickler hatten jedoch einen ganz anderen Effekt im Hinterkopf, der sich positiv auf die Geschwindigkeit des Interpreters auswirkt. Durch die strenge Numerierung ist der Archimedes in der Lage, auch von einem BASIC-Listing eine Sprungtabelle anzufertigen. Stark vereinfacht ist dies eine Minimal-Compilierung, die bei einem RUN-Start durchgeführt wird. Nebenbei gesagt ist diese Methode so neu nicht. Der CPC von Amstrad arbeitet bereits damit.

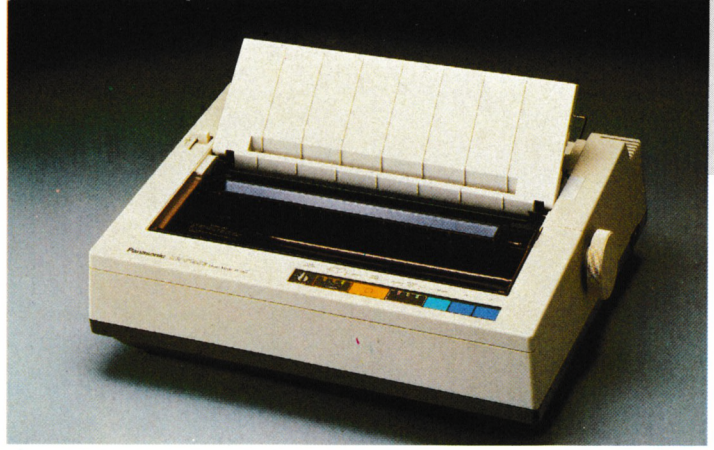
Vielfältige Soundprogrammierung

Der Archimedes besitzt nicht nur ausgezeichnete Grafik-Fähigkeiten, er beherrscht auch den „guten Ton". Für den Anwender ist die BASIC-Programmierung einfach. Mit nur drei Befehlen, SOUND, BEAT und TEMPO, lassen sich sämtliche Kanäle ansprechen und jede Note samt Tonhülle festlegen.
Durch die mitgelieferte Welcome-Diskette, die auch ein Notenprogramm enthält, erübrigen sich komplizierte Berechnungen für Musikstücke.
Ebenso verhält es sich mit den Sprites, die am besten durch den mitgelieferten Sprite-Editor erstellt werden und dann in das Programm zugeladen werden.
Schleifenprogrammierung heißt beim Archimedes nicht nur IF...THEN, ein Befehl, der ohnehin schon um ENDIF erweitert wurde. Viele Befehle aus der Gruppe der Programmsteuerung sind selbstverständlich auch bei anderen Rechnern implementiert. Die Menge derartiger Befehle dürfte jedoch nie derartig umfangreich gewesen sein. Klare Programmstrukturen sind für den Programmierer, der einige Vorüberlegungen investiert, eine Leichtigkeit.
Da gibt es WHILE...WEND-, oder WHILE-ENDWHILE-Schleifen. WHEN gehört zu der Gruppe der Bedingungen, ebenso wie CASE...OF, OTHERWISE oder ENDCASE. Die Möglichkeiten, die sich damit offenbaren, ähneln jener der regulären Sprache. Entsprechend leicht ist die Programmierung.
BBC-BASIC wartet nicht nur mit Superlativen auf. Daß an der Sprache schon seit mehreren Jahren gearbeitet wurde, obwohl bereits ein Grundstock durch den Home-Computer Proton vorhanden war, zeigt sich auch an vielen kleinen Verbesserungen, die sich Anwender insgeheim schon immer wünschten, während die Systemmentwickler Wichtigeres im Kopf hatten.
Der LIST-Befehl gibt jederzeit, auch ohne den Editor, einen gesuchten Text aus. Durch LIST IFPRINT werden zum Beispiel alle Zeilen ausgegeben, in denen das Wort PRINT vorkommt. Auch die Optik eines aufgelisteten Programmes läßt sich vorausbestimmen. Nach LISTO sorgen einige Parameter für die Anzahl der Leerzeichen nach der Zeilennummer, werden Zeilen am Doppelpunkt getrennt oder es wird gleich zur Zeile eine Fehlernummer ausgegeben.
Auch der simple INKEY-Befehl ließ sich noch verbessern. Durch INKEY zahl wartet der Archimedes soviel Zeiteinheiten der Variablen „zahl" zu je einer Zehntel-Sekunde, bis er im Programm weiterfährt.
Die Liste der Beispiele ließe sich endlos ausdehnen. Ohne die VDU- und die FX-Anweisungen verfügt BBC-BASIC V über 150 Befehle. Dabei wird der Anwender einiges neu lernen müssen, etwa den Bereich der Bibliotheksaufrufe, der Prozedurgestaltung und vielleicht auch der Programmsteuerung. Anderes ist bekannt und funktioniert einfach nur besser oder schneller.
Auf jeden Fall handelt es sich um einen Interpreter der einwandfrei und ohne Fehlerquellen arbeitet und um eine Syntax, mit der sich (fast) alles machen läßt.

Fazit

BASIC bleibt BASIC. Compilierte Sprachen laufen auch auf dem Archimedes schneller. Soviel ist sicher. Doch bei einem RISC-Prozessor sind diese Unterschiede belanglos. Da tatsächlich in bestimmte Routinen Warteschleifen eingebaut werden müssen, um den BASIC-Interpreter etwas zu bremsen, ist es müßig und unnötig, in Pascal, C oder einer anderen Sprache zu programmieren. Die Vorteile, die BASIC gegenüber diesen - recht schwer erlernbaren und umständlich aufgebauten - Listings hat, zählen weiterhin. Noch ist BASIC die Sprache, welche die meisten Struktkurfehler verzeiht, noch ist BASIC die Sprache, die der menschlichen am meisten ähnelt, und noch ist BASIC deshalb die Computer-Sprache, die am leichtesten gelernt werden kann. Nun ist sie auch eine „schnelle" Sprache geworden. GS





Dauertest bestanden

Panasonic 1083

Mit günstigem Preis und hohem Qualitätsanspruch haben sich die Panasonic-Drucker einen festen Platz am heißumkämpften Markt im Umfeld der Computer erobert. Ein Sprintstar der Panasonic-Reihe in konventioneller Neun-Nadel-Technologie ist das Modell KX-P 1083.

Seit sich die Preise für 24-Nadel- und Laserdrucker nach unten bewegen, scheinen die Tage der Neun-Nadel-Drucker gezählt zu sein. Doch der Eindruck täuscht. Trotz sinkender Nachfrage werden im Handel zur Zeit immer noch große Stückzahlen konventioneller Drucker abgesetzt. Bei vielen Käufern mag der Preis den Ausschlag geben. Aber auch die Tatsache, daß die konventionelle Drucktechnologie ausgereift ist und sich im praktischen Dauerbetrieb bestens bewährt hat, ist ein zugkräftiges Argument, das den Nachteil des nicht ganz perfekten NLQ-Schriftbildes für viele wett macht.

Der Panasonic 1083 ist ein Matrixdrukker der gehobenen Leistungsklasse, wie seine technischen Daten beweisen. Grundsolide zeigt sich bei näherem Hinsehen der mechanische und elektronische Aufbau des Gerätes, das in einem formschönen, stabilen Gehäuse sitzt.

Der großzügig dimensionierte Druckknopf mit neun Anschlagnadeln wird auf zwei kräftigen Stahlachsen geführt. Große Kühlrippen ermöglichen problemlosen Dauerbetrieb. Die Metall-Lagerbuchsen des Kopfschlittens versprechen eine hohe Lebensdauer der Führungseinheit. Die Zahnräder sind überwiegend aus Hartplastik gefertigt, die nach einem halben Jahr Dauerbetrieb keinerlei Abnutzungserscheinungen zeigten.

Der Drucker insgesamt ist ein Schwergewicht von rund zehn Kilogramm. Das spricht für ein solides Inventar und einen vibrationsarmen Betrieb, erfordert andererseits aber auch einen stabilen Unterbau, der den Betriebsbelastungen auf die Dauer gewachsen ist.

Der Panasonic verarbeitet dank Gummi- und Stachelwalze sowohl Endlos- als auch Einzelblattpapier. Die Papierzuführung erfolgt von hinten in den Schubtraktor, über die Walze bis zur Abrißkante. Diese Anordnung ermöglicht ein einfaches Papiereinlegen ohne jegliche Fummelei und eine vollständige Ausnutzung des Papiers bis zur letzten Zeile.

Gut gelöst ist auch die zweiteilige, mit einem Handgriff abnehmbare Drukkerabdeckung, die einerseits eine Geräuschdämpfung bewirkt, andererseits eine gute Zugänglichkeit zu den DIP-Schaltern (Bodenplatte) und der Papierführung ermöglicht.

Der durchsichtige Deckel an der Vorderseite ist mit einem Sicherheitsschalter kombiniert. Ein Anheben des Deckels stoppt den Druckbetrieb. Unfälle oder Beschädigungen des Drukkers während des Betriebs sind damit ausgeschlossen. Das hintere Plastikboard weist eine Zuführung für Einzelblätter auf, die dort praktisch nur hineingestellt werden. Wer es komfortabler mag, kann stattdessen auch einen automatischen Einzelblatteinzug einsetzen, der gegen Aufpreis erhältlich ist.

Die Bedienungselemente des 1083 sind übersichtlich und logisch an den richtigen Stellen plaziert. An der rechten Seite findet man den Netzschalter, links oben den Umschalter für den Traktor (Endlospapier) und die Walze (Einzelblatt). Die Funktionsschalter und die notwendigen Kontroll-Lampen sind gut sichtbar an der Vorderfront angeordnet. Die Druckschalter

```
Schriftprobenauswahl  PANASONIC  KX-P 1083

Druckertest PANASONIC KX-P 1083 Schriftart : DRAFT-10
Druckertest PANASONIC KX-P 1083 Schriftart : DRAFT-17
Druckertest PANASONIC KX-P 1083 Schriftart : DRAFT-PS
Druckertest PANASONIC KX-P 1083 Schriftart : COURIER-10
Druckertest PANASONIC KX-P 1083 Schriftart : COURIER-12
Druckertest PANASONIC KX-P 1083 Schriftart : COURIER-PS
Druckertest PANASONIC KX-P 1083 Schriftart : BOLD-PS
Druckertest PANASONIC KX-P 1083 :ELITE
Druckertest PANASONIC KX-P 1083 :ELITE-NLQ
Druckertest PANASONIC KX-P 1083 :KURSIV
Druckertest PANASONIC KX-P 1083 Schriftart  : SUPERSCRIPT
```

für den Papiervorschub (FF und LF) sind üblicher Standard. Nur wenige andere Drucker verfügen über eine Einstellmöglichkeit der Papierlänge (drei bis zwölf Zoll) und eine Funktionseinheit zum Anwählen verschiedener NLQ-Schriftarten.
Beides hat der Panasonic zu bieten. Nicht weniger als vier NLQ-Schrifttypen in Courier und Bold stehen im ROM zur Verfügung, unabhängig von der gerade verwendeten Software. Selbstverständlich läßt sich daneben eine breite Palette an Schriften als Software (etwa Proportionaldruck) abrufen.
Völlig unproblematisch ist beim KX-P 1083 der Farbbandwechsel. Die handtellergroße Kassette ist mit einem Handgriff ausgetauscht. Das besonders reißfeste Gewebeband und ein eingebauter Zusatzfarbtank sind je nach Beanspruchung (etwa bei häufigem Grafikdruck) für rund drei Millionen Zeichen gut. Das mindert die Betriebskosten.
Serienmäßig verfügt der Panasonic über eine parallele Schnittstelle, ist optional aber auch mit einem seriellen Interface lieferbar. Die Steuercodes sind vollständig kompatibel mit dem Epson RX-80 oder IBM-Proprinter, je nach Stellung der DIP-Schalter. Anpassungsprobleme an bestehende Hardware im Homecomputer-, PC-XT- oder AT-Bereich sind kaum zu erwarten.
Gleiches gilt für gekaufte Software jeglicher Art. Adressierbare, hochauflösende Grafik funktioniert ebenso problemlos wie mathematische, technische oder länderspezifische Sonderzeichen. Fünf(!) separat ladbare Zeichensätze, 32 internationale Zeichen (11 Länder), IBM-Sonderzeichen und IBM-Blockgrafikzeichen weisen den 1083 als Alleskönner im Matrixdrucker-Bereich aus. Im Textmodus wird druckwegoptimiert und bidirektional geschrieben; im Schnelldruck immerhin mit 240 Zeichen/sec. Deutlich langsamer, aber doch schneller als mancher Konkurrent, geht es mit 48 Zeichen/sec. im Korrespondenzdruck.
Zusammen mit dem druckereigenen Textspeicher von sechs KByte (wahlweise auch 32 KByte) treten in der praktischen Arbeit kaum Wartezeiten auf. Lobenswerterweise läßt sich beim Panasonic auch die Papierdicke in fünf Stufen einstellen. Zwei Durchschläge bewältigt das Gerät ohne Beanstandungen.

Fazit

Im praktischen Betrieb überzeugt der Panasonic 1083 selbst kritische Anwender. Die mechanische Stabilität, seine universelle Verwendung, die einfache und praktische Handhabung, sein vorbildliches Handbuch und nicht zuletzt ein (für einen Neun-Nadel-Drucker) ausgezeichnetes Schriftbild heben ihn von der Masse seiner Konkurrenten ab. Lediglich das Betriebsgeräusch kann mit 66 dB(A) nicht zufriedenstellen, auch dann nicht, wenn die Druckgeschwindigkeit (je schneller, desto lauter) mit in Betracht gezogen wird. Beim heutigen Stand der Technik müßte eine wirksamere Geräuschdämpfung möglich sein. Trotzdem kann dieser Drukker, gemessen an Preis und Leistung, nur empfohlen werden. ■

PANASONIC 1083

Stromversorgung	220 V/50 Hz Wechselstrom
Leistungsaufnahme	1,5 A
Druckarten	Datendruck, Korrespondenzdruck (Courier, Bold-PS), Grafikdruck
Zeichensatz	96 ASCII normal und kursiv, 32 Länderzeichen (11 Länder), 135 IBM-Spezialzeichen
Zeichenaufbau	Einzelpunktdurchmesser 0,3 mm Pica-Schrift 9 x 9 Matrix Korrespondenz-Schrift 18 x 18 Matrix
Punktdichte	bis 1/240 horizontal bis 1/144 vertikal
Zeichen/Zeile (Zeichen/Zoll)	Pica 80 (10) Elite 96 (12) Semi-schmal 120 (15) Schmaldruck 137 (17) Pica-Breit 40 (5) Elite-Breit 48 (6) Semi-Schmalbreit 60 (7, 5) Schmaldruck breit 68 (8, 5)
Druckgeschwindigkeit (Herstellerangabe)	Pica-Datendruck 240 Z/sec. Elite-Datendruck 240 Z/sec. Korrespondenzdruck 48 Z/sec.
Druckrichtung	bidirektional; undirektional bei Grafik
Zeit für Zeilenvorschub	100 ms
Papierbreite	Endlos-Lochrand 102–254 mm Einzelblatt 102–229 mm Höhe 127–363 mm
Durchschläge	ein Original, zwei Durchschläge
Papierdicke	maximal bis 0,32 mm
Lebensdauer Druckknopf	100 Millionen Zeichen
Farbband	Kassette mit Endlosband bis zu drei Millionen Zeichen
Abmessungen	427 x 350 x 137 (B x T x H)
Gewicht	9,9 kg
Preis	etwa 598 bis 898 Mark

CPC 464 m. Grünmonitor, Vortex-Erw. 512 KB, Vortex Doppellaufwerk FD1, Drucker NLQ 401, viele Top-Progr., Literatur (Bestzustand); VB 900,– DM. Tel. 02433/1350

TAUSCHE Software auf Tape, z.B. Flying Shark, Tanatos, Turbo Esprit, Deep Strike, Combat Lynx u.v.m. Suche Erbschaft u. Elite, Spindizzy (Tapes). Thomas Meyer, Oldenburgerstr. 25, 4594 Garrel

VERKAUFE 1030 neue Fragen für das Spiel KNOW. 3-Zoll-Disk für 30,– DM. Info gg. Rückporto. Henning Schnitger, Langenweg 144, 2900 Oldenburg, Tel. 0441/301492

TAUSCHE auf Disk (3-Zoll, 5 1/4 Zoll) oder Kassette Spiele- und Anwendersoftw. Eingeschickte Programme bekommt ihr neu bespielt zurück. Thomas Süllau, Föttingerzeile 37, 1000 Berlin 48

SEARCHING for fast guys with new stuff! Robbert Jan Smit, Gr. jul. van stolberglaan 28, NL - 2263 VA Leidschendam, Tel. 070/278151

Gratisliste für alle CPC-Versionen bei Friedrich Neuper 8473 Pfreimd Postfach 72

SUCHE Tauschpartner f. Software aller Art (CPC 464, 3-Zoll, Vortex 5 1/4 Zoll). Stefan Zeiger, Seligenstädter Weg 24, 8756 Kahl, Tel. 06188/2425 ab 14 h.

VERKAUFE CPC Public-Domain-Software (Spiele u. Anwenderprogr.), engl. u. dt. Für Info/Liste bitte –,80 DM Rückporto. K. Bartram, Haardtwaldstr. 1–7, 6000 Frankfurt 71, Tel. 069/6702230

CPC 464 (gr.), SP64, DDI-1 FS1, DMP 2000, Softw., Bücher, Magazine, Lightpen, Joycard etc. f. ca. 1000,– DM. Tel. 040/2194405 ab 20 h.

VERKAUFE Schneider-Grünmonitor GT 65, VB 200,– DM (ungebraucht). Tel. Augsburg 709638

VERKAUFE Small-C Entwicklungssystem: C-Compiler, 8080/Z80-Macroassembler, Linker/Loader, Bibliotheksverwalter, Editor/Texttools CPC 464/664 (benötigen Speichererweiterungen, min. 56 KB): Orig.-Handb. u. 3 Disketten f. 90,– DM, bei Nachnahme 94,– DM. Klaus Lindner, Weilerstr. 7, 5303 Bornheim 4

TAUSCHE Games CPC 6128 auf 3 Zoll, z.B. Bubble Bobble, Western G. u. viele News. Verkaufe Trivial Pursuit. Antwort bei Rückumschlag. Thomas Biegun, Kirchplatz 4, 6685 Eppelborn

VERKAUFE CPC 6128 grün, 1 Jahr alt, kaum gebraucht, m. 107 Spielen auf Disk, Software, Computerzeitschriften, Joysticks, Recorderkabel u. Kassettenspiele etc. f. 650,– DM. Oliver Borchert, Tel. 0581/74499

LICHTGRIFFEL nur DM 49,–. Vers. gegen Scheck/Nachnahme. Info gratis Computer angeben. Anschluß an jeden Computer möglich. Standardversion für Schneider lieferbar. Firma Schißlbauer Postfach 1171T 8458 Sulzbach Tel. 09661/6592 oder 0941/999915 bis 21 Uhr

VERKAUFE div. Spielesoftware auf. Kass. (orig.). Liste geg. frank. Rückumschlag. Bomb-Jack-Disk, 20,– DM. Klaus Lindner, Weilerstr. 7, 5303 Bornheim 4

TAUSCHE Games CPC 6128 auf 5 1/4 Zoll, z.B. Bubble Bobble, Western G. u. viele News. Verkaufe von orig. Games auf 3 Zoll. Antwort bei Rückumschlag. Helmut Gross, Pastor-Vogt-Str. 23b, 6685 Eppelborn

CPC-COMPUTERCLUB „BHG" sucht immer noch neue Mitglieder. Es gibt eine Clubzeitung, Clubsoftware, Tips etc. Info anfordern mit –,80 DM Rückporto bei BHG, Postfach 30, 8115 Ohlstadt

SUCHE zuverlässige Tauschpartner f. CPC 464/664/6128 (3-Zoll-Disk). Richard Höfter, Herbststr. 2, 8011 Baldham

KAUFE Software aller Art für Tape (Anwendungen, Adventure, auch Selbstgeschriebenes). Listen an Lars Stracke, Schlehdornstraße 20, 5910 Kreuztal 4

SUCHE billige Orig.-Software f. CPC 6128 auf 3 Zoll. Christian Wagner, Altstadt Str. 2, 8741 Sulzfeld 1

SUCHE Tauschpartner f. CPC 464-Progr. auf Tape od. 5 1/4-Zoll-Disk. Markus Klesen, Mozartstr. 14, 6686 Eppelborn Dirmingen

ZWEITFLOPPY FD1 f. 270,– DM. Suche zuverlässige Tauschpartner auf 3-Zoll-Disk. Walter Baum, Markgrafenstr. 87, 5000 Köln 80, Tel. 0221/624618

VERKAUFE Disk mit neuester Software für 22,– DM. K. Schwettmann, Steubenstr. 6, 4990 Lübbecke

VERKAUFE Schneider CPC International Hefte (ab 3/85 bis 5/88) u. alle Sonderhefte; Normalheft 2,– DM, Sonderheft 6,50 DM. Tel. 07971/4637

SUCHE Erfahrungsaustausch u. Tauschpartner, CPC 6128, keine Spiele, Gerd Rothfuchs, An der Weid 28, 6799 Etschberg, Tel. 06381/3438

VERKAUFE Drucker: Star stx-80 f. nur 200,– DM; 7- od. 8-Bit-Interface über DIP-Schalter wählbar, grafikfähig usw. Tel. 06403/5478

ZEUGNIS CPC Notenverwaltung f. Lehrer; Info gg. Freiumschlag. H. Kranz. Winterbacherstr. 9, 7060 Schorndorf

SUCHE für CPC 464 dringend Action-Spiele. G. Hoffmann, Enggasse 28, 5540 Prüm

PUBLIC-DOMAIN-SOFTWARE für CPC 464/6128 u. Joyce. Antwort bei 80 Pf. Rückporto. W. Krimkowski, Försterstr. 31, 5000 Köln 30

SUPER-LOHN-EINKOMMENSTEUER Jahresausgleich 87 neu m. Datenspeicher, Kundenverw., Formularadr., Analyse; ab 70,–; jährl. Aktu. 10,–; Info gg. RP; PC-Demo-Disk 10,–. MIET-, WOHNGELD-BERECHNUNG Mit allen Kreisen d. BRD! Jedes Prg, ab 70,–. H-I-Software H. Ilchen, Niederfelderstr. 44, 8072 Manching, Tel. 08459/1669

VERKAUFE CPC 6128, 60 3-Zoll-Disks, Schneider-Joy, Competition Pro, Stereo-Adapter, Staub-Schutzhauben f. 999,– DM (auch einzeln). Tel. 09009/1578

VERKAUFE CPC 464, GT64, dk'tronics 64 KB Memory, Vortex F1D Doppelfloppy (2LW a 708 KB) m. VDOS 2.0, Joystick, Spiele, Wordstar, dBase II, CP/M 2.2 u. CP/M plus, DOSCOPY zur Bearbeitung von DOS-Disketten (nur komplett) gegen Gebot; Multiface II Kopiermodul u. Gamefreezer m. Monitor (anschl. an CPC 464) f. 100,– DM. Tel. 06051/73745

SUCHE Tauschpartner in Wuppertal. Listen an Stephan Kaldewey, Nordenbachstr. 12, 5600 Wuppertal 21. Suche Kontakt zu CPC-Usern im Raum Wuppertal. Tel. 0202/4670949

CPC 6128, Grünmon., 100 Prog., Data Recorder m. Anschluß, 22 Disks u. 5 Kass., 20 Zeitschr., 2 Data-Becker-Bücher, 2 Joysticks. NP 1.200,– DM, VP 700,– DM. Tel. 06765/7230

PC International, Jahrgänge 1986/87 komplett, 1–4/88, einige CPC-Hefte. Komplettpreis inkl. Porto 90,– DM. Gerd Rothfuchs, An der Weid 28, 6799 Etschberg, Tel. 06381/3438

VERKAUFE CPC 6128 DM 700,–; CPC 464 DM 230,–; NLQ 401, Traktor, Kabel, DM 300,–; DMP 2000 u. Kabel DM 370,–; CPC-Maus DM 150,–. Software: Mastercopy DM 30,–; Supercopy DM 35,–; AMX-Pagemaker DM 80,–; Reisende im Wind DM 30,–; The Sentinel DM 20,–. Tel. 08334/1513

***EINMALIG*SUPER* KASSENBUCH- und Girokontobuchführung m. UMSATZ-STEUER-VORANMELDG.-Ausdruck Schneider CPC 464, 664, 6128 ADRESSVERWALTUNG m. Einzel-Serien-Etikettendruck Karteiverwaltung Anhang: normal – Rechnungsschreiben Serienbrief pro Prg. DM 70,– + MwSt., Porto u. Versand S-B Software Bastian, Zusenhofener 15, 7602 Oberkirch 2 Tel. 07805/3941**

Meinen Stammkunden danke ich auf diesem Wege für Ihr reges Interesse

SUCHE Spiele auf 3-Zoll u. 5 1/4 Zoll (F1X), Pegasus u.a. Listen an mich. Schicke meine Liste an Euch. Ansonsten Kontakt zu anderen 6128-Usern im Raum Hameln gesucht. Roger Salzenberg, Kuhle 1, 3258 Aerzen 16, Tel. 05154/3562. Anwendersoftw. ebenfalls ges.

THOMSON-CLUB sucht Thomson-User. Bitte melden bei Stefan Sedlaczek, Ginsterweg 1a, 5063 Overath 8, Tel. 02204/73320

VERKAUFE dk'tronics Speech-Synthesizer 30,– DM; CPC-Mouse-Pack 100,– DM; tausche Top-Anwender-SW. Tel. 02461/7266

IMPRESSUM

CPC-SPECIAL erscheint zweimonatlich in der CA Verlags GmbH, einer Gesellschaft der AKTUELL-Gruppe, Heßstraße 90, D-8000 München 40, Tel. 089/1298011, Telex: 5214428 cav d.

VERANTWORTLICH FÜR DEN INHALT:
Alwin Ertl

REDAKTION UND STÄNDIGE MITARBEITER:
Dipl.-Ing. Manfred Eibisch, David Krasucki, Torsten Seibt, Gert Seidel, Gerhard Wertenbach

HERSTELLER:
Anton Kult

MONTAGE:
Walter Esterl

GESCHÄFTSFÜHRER:
Werner E. Seibt

VERANTWORTLICH FÜR ANZEIGEN:
Brigitte Kostic

ANZEIGEN-VERWALTUNG:
ADV-Mediendienste beide: Postfach 101124, 8900 Augsburg 1, Tel.: (0821) 7904-0, Telekopierer: (0821) 7904-243, Telex: adv 533502, Teletex: 821887

Es gilt Preisliste Nr. 9 vom 1.6.1988. Media-Unterlagen bitte anfordern.

ANSCHRIFT FÜR ALLE VERANTWORTLICHEN:
Postfach 1161, 8044 Unterschleißheim, Tel.: 089/1298011, Telex: 5214428 cav-d

©1988 by CAV-GmbH, Heßstraße 90, 8000 München 40. Für unaufgefordert eingesandte Manuskripte und Listings keine Haftung. Bei Einsendung von Texten, Fotos und Programmträgern erteilt der Autor dem Verlag die Genehmigung für den Abdruck und die Aufnahme in den Softbox-Service zu den Honorarsätzen des Verlages. Das Copyright und das Recht der wirtschaftlichen Verwertung gehen auf den Verlag über. Alle in dieser Zeitschrift veröffentlichten Beiträge sind urheberrechtlich geschützt. Jede Verwertung ist untersagt. Namentlich gezeichnete Beiträge unserer Mitarbeiter stellen nicht unbedingt die Meinung der Redaktion dar.

LAYOUT:
Sonja Anderle & Bernhard Schnitzler

VERTRIEB:
Verlagsunion Wiesbaden

Printed in Germany by ADV, Aindlingerstr. 17–19, 8900 Augsburg 1

COMPUTERN LEICHT GEMACHT

Das PC-Magazin



NEU

Jetzt an ausgewählten Kiosken und im Bahnhofs-Buchhandel