

Mit  
Inhaltsverzeichnis  
'85-'87  
auf Papier und Diskette

DAS ELEKTRONISCHE MAGAZIN 12/87

# INPUT 64

Infos · News · Programme · Unterhaltung · Tips **DM 19,80**

Unverbindliche Preisempfehlung

öS 170,- sfr 19,80

Ordnung für die Daten

## UniDat

Entwicklungshilfe  
für BASIC

## Programmer's Little Helper

Domino einmal anders

## Triangles

Serien:

64er Tips

Compiler-Know-how

Einer gegen Alle

Optik: VisHypnotica

Drucken: Graudruck

Test: Btx-Decoder

Spiel: Fighting Hardware

Über 140 KByte Software  
Ohne Abtippen

# Hinweise zur Bedienung

INPUT 64 ist nicht nur einfach eine Programmsammlung auf Diskette, sondern ein Elektronisches Magazin. Es enthält ein eigenes Betriebssystem mit Schnelllader und komfortabler Programmauswahl. Die Bedienung ist kinderleicht.

Bitte entfernen Sie vor dem Laden eventuell vorhandene Steckmodule, und schalten Sie den Rechner einmal kurz aus und wieder ein. Geben Sie nun zum Laden der Diskette

## LOAD "INPUT\*",8,1 und RETURN

ein. Alles Weitere geschieht von selbst.

Es wird nun zunächst ein Schnelllader initialisiert. Besitzen Sie ein exotisches Laufwerk oder ist Ihre Floppy bereits mit einem hardwaremäßigen Beschleuniger ausgerüstet, kann es zu Konflikten mit unserem SuperDisk kommen. In diesem Falle sollten Sie versuchen, die Diskette mit

## LOAD "LADER\*",8,1 und RETURN

zu laden.

Nach der Titelgrafik springt das Programm in das Inhaltsverzeichnis des Magazins. Hier können Sie mit der Leertaste weiter und mit SHIFT und Leertaste zurückblättern. Mit RETURN wird das angezeigte Programm ausgewählt und geladen.

Das Betriebssystem von INPUT 64 stellt neben dem Inhaltsverzeichnis noch weitere Funktionen zur Verfügung. Diese werden mit der CTRL-Taste und einem Buchstaben aufgerufen. Sie brauchen sich eigentlich nur CTRL und H zu merken, denn mit dieser Tastenkombination erscheint eine Hilfsseite auf dem Bildschirm, die alle weiteren System-Befehle enthält. Nicht immer sind alle Optionen möglich. Befehle, die zur Zeit gesperrt sind, werden auf der Hilfsseite dunkel angezeigt. Hier nun die Befehle im einzelnen:

### CTRL und Q

Diese Tastenkombination hat nur während der Titelgrafik eine Bedeutung. Mit ihr wird

das Titelbild abgekürzt, und Sie landen sofort im Inhaltsverzeichnis.

### CTRL und H

Haben wir schon erwähnt – damit wird die Hilfsseite ein- und ausgeschaltet.

### CTRL und I

Sie verlassen das gerade laufende Programm und kehren ins Inhaltsverzeichnis zurück.

### CTRL und F

Ändert die Farbe des Bildschirmhintergrundes. Diese Option funktioniert immer, wenn ein Programm läuft oder Sie sich im Inhaltsverzeichnis befinden, aber nicht auf der Hilfsseite.

### CTRL und R

Wie CTRL-F, wirkt auf die Rahmenfarbe.

### CTRL und B

Sie erhalten einen Ausdruck der Textseite eines laufenden Programmes auf einem angeschlossenen Drucker. Diese Hardcopy-Routine ist angepaßt für Commodore-Drucker und kompatible Geräte. Das Programm wählt automatisch die richtige Geräteadresse (4, 5 oder 6) aus. Sie können diese Routine mit der ←-Taste abbrechen.

### CTRL und S

Programme, die auch außerhalb von INPUT 64 laufen, können Sie mit diesem Befehl auf eine eigene Diskette überspielen. Wenn Sie diesen Befehl aktivieren, bekommen Sie unten auf der Hilfsseite angezeigt, wie viele Blocks das File auf der Diskette belegen wird. Geben Sie nun den Namen ein, unter dem das Programm auf Ihre Diskette geschrieben werden soll. In der Regel handelt es sich um Programme, die Sie ganz normal laden und mit RUN starten können. Ausnahmen sind in den jeweiligen Programmbeschreibungen erläutert.

### CTRL und D

Gibt das Directory der eingelegten Diskette

aus. Die Ausgabe kann mit der Leertaste angehalten und mit RETURN wieder fortgesetzt werden. Ein Abbruch ist mit der ←-Taste möglich. Wenn das Directory vollständig ausgegeben ist, gelangen Sie mit der RETURN-Taste zurück ins unterbrochene Programm beziehungsweise auf die Hilfsseite.

### CTRL und @

Disk-Befehle senden, zum Beispiel Formattieren einer neuen Diskette oder Umbenennen eines Files. Für den zu sendenden Befehls-String gilt die übliche Syntax, natürlich ohne ein- und ausführende Hochkomma. CTRL-@ und RETURN gibt den Zustand des Fehlerkanals der Floppy auf dem Bildschirm aus. Weiter im Programm oder zurück auf die Hilfsseite führt ein beliebiger Tastendruck.

### CTRL und A

Sucht auf der Diskette nach einem INPUT 64-Inhaltsverzeichnis. Mit diesem Befehl ist es möglich, ohne den Rechner auszuschalten, Programme von anderen INPUT 64-Disketten zu laden. Das funktioniert aber nur bei den Ausgaben ab 4/86.

## Bei Ladeproblemen

Bei nicht normgerecht justiertem Schreib-/Lesekopf oder bei bestimmten Serien wenig verbreiteter Laufwerke (1570) kann es vorkommen, daß das ins INPUT-Betriebssystem eingebaute Schnelladeverfahren nicht funktioniert. Eine mögliche Fehlerursache ist ein zu geringer Abstand zwischen Floppy und Monitor/Fernseher. Das Magazin läßt sich auch im Normalverfahren laden, eventuell lohnt sich der Versuch.

### LOAD "LADER",8,1

Sollte auch dies nicht zum Erfolg führen, senden Sie bitte die Diskette mit einem kurzen Vermerk über die Art des Fehlers und die verwendete Gerätekonstellation an den Verlag (Adresse siehe Impressum).

## Liebe(r) 64er-Besitzer(in)!

Mit dieser Ausgabe erhalten Sie zwei Mal einen „nach zweckmäßigen Kriterien geordneten, zur Aufbewahrung geeigneten Bestand an sachlich zusammengehörenden Belegen oder anderen Dokumenten“ (Datei [lat.] *die*, aus Duden, „Das Fremdwörterbuch: Notwendig für das Verstehen und den Gebrauch fremder Wörter“, Bibliographisches Institut, Mannheim/Wien/Zürich, 1982): Das Jahresarhaltsverzeichnis aller INPUT 64-Ausgaben der Jahre 1985 bis 1987.

Eine Dokumentation des gesamten Software-Bestands, zusammen 361 Programme und Artikel – eine Anzahl, die schon nach zweckmäßigen Kriterien geordnet sein will, wollen Sie bei der Suche nach einem bestimmten Programm nicht die Übersicht verlieren. Doch warum das ganze Verzeichnis gleich zweimal? Reicht da nicht das vertraute Gesamtverzeichnis schwarz auf weiß zum Nachschlagen? Oder ist es zeitgemäßer und eher dem Charakter von INPUT 64 entsprechend, Ihnen statt dessen nur eine Disketten-Datei zur Verfügung zu stellen nebst einer Dateiverarbeitung, mit der Sie auf den Datenbestand zugreifen können?

So manchem von Ihnen mag es ebenso gegangen sein: Auf der Suche nach einem bestimmten Thema fällt einem plötzlich ein: „Das habe ich doch vor ein paar Monaten irgendwo gesehen . . .“, und die Sucherei geht los. Spätestens hier lernt man die Vor- und Nachteile der verschiedenen Medien Papier und elektronische Datenträger kennen.

Hoffen Sie das Gesuchte in einem Magazin zu finden, greifen Sie wahrscheinlich zuerst zu den Dezemberausgaben mit den Jahresarhaltsverzeichnissen. Diese paar Seiten bieten schnell einen einfachen Überblick. Selbst eingefeischte Computer-Anwender dürften sich fein bedanken, wenn sie erst nach Ladeprozeduren und Bildschirmarbeit feststellen können, ob im letzten Jahr beispielsweise etwas zu Lisp in INPUT 64 veröffentlicht wurde. Papier ist geduldig – ruhiger als ein Bildschirm und kann (noch) erheblich mehr Informationen auf einer DIN-A4-Seite aufnehmen als Ihr Bildschirm

mit 40\*25 Zeichen. Außerdem bietet die bequeme Haltung im heimischen Lieblingssessel doch mehr an Attraktivität als die zwingende Nähe zu Bildschirm und Tastatur.

Suchen Sie jedoch nach einem Stichwort oder einer anderen Zusammenstellung als im Jahresarhaltsverzeichnis verwendet, wird es um einiges unbequemer. Mal angenommen, Sie wollen entscheiden, ob sich die Nachbestellung einer früheren Ausgabe lohnt. Lisp erschien in den Ausgaben 4–6/86. Was für Programme sind denn in diesen Ausgaben noch enthalten? Spätestens hier entsteht der Wunsch nach einer vernünftigen Dateiverwaltung.

Genau hierbei ist die EDV dem Medium Druck und Papier überlegen. Sie geben das Suchkriterium ein, um im Beispiel zu bleiben: die Ausgabennummer, schauen sich die ausgewählten Informationen an und können das Drucken Ihrer hauseigenen Anlage überlassen.

Deutlich wird aber dabei, daß der Sinn des Ganzen von den Datenmengen abhängt – für 10 Telefonnummern lohnt sich die ganze Mühe kaum, ein Verzeichnis von 500 Kochrezepten wäre dagegen sogar von allgemeinem Interesse. Deshalb unser Angebot: Haben Sie Daten, die auch für andere von Nutzen sein können, helfen wir Ihnen gerne als Vermittler.

Ihre INPUT 64-Redaktion

*Ralph Hülsenbusch*  
R. Hülsenbusch

### Letzter Drucker

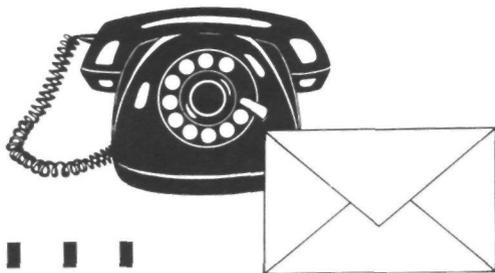
Der mit 3000 DM dotierte **INPUT 64-Adventure-Wettbewerb** läuft noch bis zum 15. Januar 1988. Das sind, ab Erscheinen dieser Ausgabe, immerhin 40 Tage – für engagierte Programmierer eine halbe Ewigkeit. Ausführliche Teilnahmebedingungen sind bei der Redaktion zu erfahren.



### INHALT

<b>Leser fragen</b>	2
<b>Test</b> Btx-Decoder von Commodore	3
<b>Buchkritik</b> Das große Btx-Buch	5
<b>UniDat</b> Indexsequentielle Dateiverwaltung	6
<b>64er Tips</b> Interrupts des Video-Chips	12
<b>Inhaltsverzeichnis '85-'87</b>	15
<b>Datei-Generator</b> Tips zur INPUT-Datei	18
<b>Fighting Hardware</b> Spiel für zwei Partner	19
<b>Programmer's Little Helper</b> BASIC-Programmierhilfe	20
<b>Compiler-Know-how</b> Tips zum Speedcompiler/Teil 2	23
<b>Studio</b> VisHypnotica	26
<b>Einer gegen Alle</b> Spiel um Dechiffrierung	27
<b>Graudruck</b> Multicolor-Hardcopy ohne Farbdrucker	28
<b>Triangles</b> Dominospiel auf Dreiecks-Basis	29
<b>Vorschau</b>	31
<b>Impressum</b>	32

# Leser fragen . . .



## Ideen-Beitrag

... ich freue mich, daß Sie in der ID-Werkstatt der Oktoberausgabe die drei kleinen Programme von mir veröffentlicht haben. Nun werden aus dem Modul die drei Files nicht ganz richtig abgespeichert: Es fehlen jeweils zwei Null-Bytes am Ende, und das Programm ist um eine Speicherstelle zu „kurz“. Sicher sind Sie selber schon darauf gekommen, der Einfachheit halber aber nachfolgend ein „Rezept“:

Programm (vom eigenen Datenträger, d. Red.) laden, nicht starten, dann im Direktmodus eingeben:

```
l=peek(45)+peek(46)*256
poke l-1,0
poke l,0
poke 45,peek(45)+1
clr
```

Dann kann das Programm wieder abgespeichert werden. ( P. Casulleras, Tecknau, Schweiz)

Als Erklärung zu dieser „Zubereitungs-methode“: Im ersten Befehl wird das Programmende in die Variable 'l' übernommen. In das letzte und in das darauf folgende Byte wird jeweils eine Null geschrieben und anschließend das Programmende um ein Byte erhöht. Der Befehl „CLR“ restauriert dann die Programmzeiger. (d. Red.)

## Urheberrechte zum SPEEDCOMPILER

Gratulation zu Ihrem fantastischen Tool SPEEDCOMPILER! Ein wirklich professionelles Werkzeug zum Preis einer Zeitschrift, das ist schon unglaublich.

Ich habe – gerade deswegen – eine für mich wichtige Frage. Wer besitzt das Urheber-

recht an mit dem SPEEDCOMPILER behandelten BASIC-Programmen? Darf ich von mir geschriebene und mit dem SPEEDCOMPILER compilierte BASIC-Programme frei verkaufen, oder müßte ich Lizenz-Gebühren an Sie zahlen?

(S. de la Motte, München)

Diese Frage können wir mit einem zufriedenstellenden „Ja“ beantworten. Im Heft, Ausgabe 10/87, wurde bereits auf Seite 11 unten rechts gesagt: „Der Interpreter darf beliebig kopiert und weitergegeben werden. Damit sind kompilierte Programme sogar kommerziell nutzbar“, sozusagen die schriftliche Freigabeerklärung. (d.Red.)

## INPUT-SCE – ohne „0“?

Der INPUT-SCE (6/87) gefällt mir sehr gut! Leider funktioniert die '0'-Taste (Punkt löschen und Cursor links) nicht! Wird sie gedrückt, dann entspricht sie der '@'-Taste (Punkt setzen und Cursor rauf). Wie kann ich Abhilfe schaffen? Außerdem wüßte ich gerne, welchen Speicherbereich der SCE belegt, da ich ihn absolut laden möchte.

(D. Jenßen, Wisch)

Der Fehler im Zeichensatz- und Sprite-Editor kann mit folgenden Arbeitsschritten behoben werden:

Laden Sie INPUT-SCE vom eigenen Datenträger, – nicht mit „RUN“ starten, – und geben Sie die folgenden POKE-Sequenzen im Direktmodus ein:

```
poke 45,80:poke 46,47:clr
poke 5009,76
poke 5010,166
poke 5011,102
```

```
poke 12102,32
poke 12103,24
poke 12104,81
poke 12105,76
poke 12106,78
poke 12107,81
```

Speichern Sie das Programm danach wieder ab. Jetzt arbeitet der SCE von 'A' bis 'O'.

Nach dem Start belegt der SCE den Bereich von 16384 bis 26288 (\$4000 bis 66B0). Man kann ihn mit einem Monitor (MLMplus, INPUT 64, Ausgabe 11/87) aus diesem Bereich abspeichern. Wird er absolut geladen, mit Sekundäradresse 1, kann er mit SYS 16384 gestartet werden. (d. Red.)

## INPUT-Calc sieht doppelt

CALC rechnet falsch!

zum Beispiel:

5 \* 125 = 12.25

und nicht wie erwartet 6.25. Die Stelle vor dem Komma wird verdoppelt, die Nachkommastelle ist richtig. (H.-J. Stelz, Florstadt)

Das Patch-Programm für INPUT-CALC in der ID-Werkstatt aus Ausgabe 10/87 erzeugt im BASIC-Teil einen Fehler.

Laden Sie die „gepatchte“ Version von INPUT-CALC 64/128 im C64-Modus vom eigenen Datenträger, starten mit „RUN“ und brechen mit RUN/STOP ab. Mit LIST 5400 erreichen Sie die fehlerhafte Zeile:

```
5400 w=int(w)+int(w*10↑nk
(zs)+.5)/10↑nk(zs)
```

Korrigieren Sie diese Zeile zu:

```
5400 w=int(w*10↑nk(zs)+.5)
/10↑nk(zs)
```

Setzen Sie mit POKE 44,8 und POKE 43,1 den Programmanfang wieder zurück und speichern das Programm wieder auf Datenträger ab. Nach dieser Operation rechnet das Programm, wie es den arithmetischen Regeln entspricht, und rundet die Ergebnisse auf die Anzahl der angegebenen Nachkommastellen. (d. Red.)

**Test:**

# C64 am Draht

## Das neue BTX-Modul für den C64/C128

Bei dem augenblicklichen Preisrutsch des C64, der auf die 200,- DM Schallgrenze abzielt (das Weihnachtsgeschäft läßt hoffen), wären die geringen Kosten ein wirklich schlagkräftiges Argument, sich mit dem Thema Btx einmal näher auseinanderzusetzen. Möglicherweise verhilft dieses Modul dem Bildschirmtext der Post zu dem seit langem angekündigten Durchbruch; blieben die Teilnehmerzahlen bei Btx doch stets hinter den Erwartungen zurück. Eine Beurteilung des Moduls soll hier nach zwei Gesichtspunkten erfolgen.

1. Was bietet das Modul im Vergleich zu einer „echten“ Btx-Anlage? Lohnt sich eventuell sogar die Anschaffung eines C64 nur zum Btx-Betrieb?
2. Was bietet das Btx-System dem Benutzer? Welche Vorteile hat der Benutzer gegenüber dem Nichtbenutzer?

Zuerst zum Modul selbst. Der Einbau ist denkbar einfach. Bei ausgeschaltetem Rechner wird das Modul in den Expansions-Port gesteckt. Der Anschluß eines Monitors oder Fernsehgerätes ist dank mitgelieferter Kabel problemlos möglich, doch gibt es unter Umständen Schwierigkeiten bei der Bildschirmdarstellung, aber dazu später mehr. Voraussetzung für den Betrieb ist das Vorhandensein eines eigenen Btx-Anschlusses im Haus, der an die normale Telefonleitung angeschlossen werden kann. Hierzu ist ein Antrag bei der Post nötig. Nach einer einmaligen Anschlußgebühr von 65,- DM kann der Spaß losgehen. Nachteil dieser Methode ist, daß für die Dauer des Btx-Betriebes das Telefon lahmgelegt ist, so daß sich eventuell zur Rettung des Familienfriedens oder zur Vermeidung von Isolationserscheinungen die Anschaffung eines Zweitanschlusses bezahlt machen kann (Firma Schwarz-Schilling dankt).

**Zum zweiten Mal hat Commodore jetzt ein Btx-Modul für den C64/C128 vorgestellt. Eine preiswerte Alternative zu den vergleichsweise teuren Anschaffungskosten bei einem Btx-fähigen Fernsehgerät mit der speziellen Btx-Tastatur. Der Weg, den man bei der Entwicklung des Moduls gewählt hat, ist naheliegend: Die vorhandene Hardware, bestehend aus Computer, Bildschirmgerät und einem Diskettenlaufwerk, wird durch das Modul für 399,- DM um die notwendige Hard- und Software ergänzt und stellt somit ein vollständig funktionsfähiges Btx-System dar.**

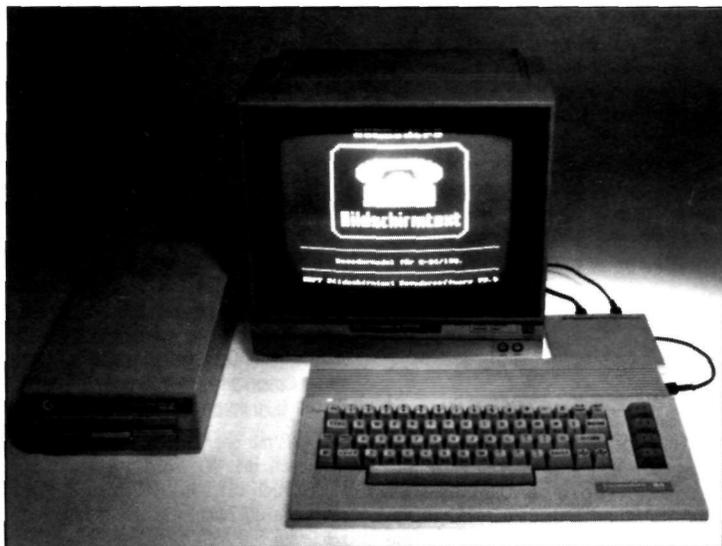
Wer als Besitzer nur eines Telefons zur Zeit des Btx-Betriebes angerufen wird, merkt nichts davon, der Anrufer hört nur das Besetzzeichen.

## ON LINE — Leinen los!

Sind diese Klippen jedoch erst einmal umschifft, kann es losgehen. Nach der Eingabe von Benutzernummer und Kennwort befinden Sie sich im System und stellen erfreut fest, daß sich die Bildschirmdarstellung von den gewohnten Btx-Geräten nicht unterscheidet. Das Modul leistet hier ganze Arbeit. Durch einen eigenen Video-Chip wird eine höhere Bildschirmauflösung von 480\*240 Punkten und die Darstellung von allen bei Btx verwendeten 4096 Farben ermöglicht.

Auch das Problem der Tastaturanpassung wurde gut gelöst. Der Benutzer hat die Wahl zwischen einer deutschen DIN- oder einer amerikanischen ASCII-Tastatur. Letztere entspricht der gewohnten C64-Belegung. Bei der deutschen Version sind alle

**Mit dem Decoder-Modul II von Commodore wird der 64er zum vollwertigen Btx-Terminal.**



Umlaute und 'ß' an gewohnter Stelle vorkommen. Auch 'z' und 'y' wurden vertauscht. Hier empfiehlt sich unter Umständen das Anbringen kleiner Aufkleber mit der neuen Belegung. Die meistgebrauchten Funktionen und die speziellen Btx-Steuerzeichen (\* und #), sind schnell über die Funktionstasten zu erreichen. Nach einiger Zeit hat man deren Belegung schnell im Kopf. Für alle Fälle enthält das Handbuch noch einen Bastelbogen zum Ausschneiden bereit, dem man – einmal auf der Tastatur fixiert – die Belegung jederzeit entnehmen kann.

Angenehm ist die Tatsache, daß das Diskettenlaufwerk in den Arbeitsablauf mit einbezogen werden kann. So ist es zum Beispiel möglich, im sogenannten CAPTURE-Modus (F7 und C) eine Anzahl von Bildschirmseiten in den RAM-Puffer des C64 zu schreiben, um sie anschließend auf Diskette abzuspeichern. Für den späteren Gebrauch kann man sie dann gebührenfrei im OFF-LINE Betrieb noch einmal betrachten. Auch die Arbeit mit Makros wird unterstützt. Das Konzept von Btx erfordert bis zum Erreichen einer bestimmten Seite das Durchblättern vieler Zwischenmenüs. Diesem kann man auf zweierlei Weise begegnen. Entweder man merkt sich die zu jeder Seite gehörige spezifische Nummer, um sie später direkt anzuwählen, oder man speichert die Tastenfolge, die zum Erreichen eines bestimmten Bildschirms notwendig ist, in einem Makro ab. Den Makros werden beim Erstellen zwecks besseren Wiederfindens eindeutige und aussagefähige, freidefinierbare Namen und jeweils eine Kennung zugeteilt. Aber auch alle anderen Btx-Funktionen, zu deren Aufruf längere, immer wiederkehrende Tastenfolgen notwendig sind, können durch Makros bequemer erreicht werden. Bis zu 36 Makros können auf einer Diskette verwaltet werden.

Interessant ist die Funktion „Telesoftware“, die das Überspielen von Programmen über die Telefonleitung ermöglicht. Voraussetzung dafür ist ein Anbieter, der diesen Service unterstützt. Die Übertragung dauert zwar einige Zeit, wie das bei allen Btx-Funktionen der Fall ist, aber die Geschwindigkeit übertrifft immer noch die einer 300-Baud-Modem-Verbindung. Die Programme werden direkt auf Diskette gespeichert, von wo aus sie später abgerufen werden können.

Leider ist hier kein einheitlicher Standard vorhanden, so daß man die Software von manchen Anbietern nicht ohne Zusatzgeräte einholen kann. Laut Commodore wurde das hier beschriebene Decoder-Modul in Zusammenarbeit mit der Post entwickelt und dabei vereinbart, daß das Empfangen von Telesoftware mit diesem Modul als der eigentliche Post-Standard gelten soll.

## Bunt, aber langsam

Unterstützt wird auch die Ausgabe des aktuellen Bildschirminhaltes im ASCII-Format auf ein File im Diskettenlaufwerk, so daß eine spätere Nachbearbeitung möglich ist. Leider kann hier jedoch immer nur eine Bildschirmseite in ein File abgelegt werden. Gerade hier wäre es besser gewesen, mehrere Seiten in einem File unterzubringen. Bei so vielen Disketteneinträgen verliert man leicht den Überblick. Die andere Möglichkeit, die dieser Modus bietet, ist die Ausgabe auf einen Drucker, wobei hier jedoch nur der Textmodus unterstützt wird. Auf dem Bildschirm enthaltene Grafiksymbold erscheinen beim Ausdruck nur als Sternchen. Mit dieser Lösung wird das Problem einer variablen Druckeranpassung etwas unbefriedigend umgangen.

Abhilfe könnte hier vielleicht eine Möglichkeit schaffen, die im Anhang des Handbuchs angesprochen wird. Dort sind eine ganze Reihe von Sprungadressen aufgelistet, die es dem Assemblerprogrammierer ermöglichen sollen, eigene Erweiterungen für das Btx-System zu programmieren. Ein File mit der spezifischen Kennung „BTX-EXTRA.MAS“ wird von der Diskette geladen und gestartet, wenn beim Einschalten die CTRL-Taste gedrückt wurde. Leider fehlt es hier an einer ausführlichen Erklärung, wie, wo und wann diese externen Programme angewendet werden können und an einem Anwendungsbeispiel, so daß der interessierte Programmierer zum Experimentierer werden muß. Denkbar wäre allerdings, mit dieser Option eine Erweiterung zum Grafikausdruck zu programmieren.

Ebenfalls nur oberflächlich gestreift wird im Handbuch die Möglichkeit, daß mit dem Modul auch ein Datex-P-Betrieb in einem speziellen 80-Zeichen-Modus möglich sein soll.

Der insgesamt positive Eindruck des Moduls wird durch eine Einschränkung getrübt, die beim Testen auftrat. Ausgerechnet bei der Verwendung des Commodore-eigenen Monitors VC-1701 war es nicht möglich, eine saubere Darstellung eines Textes auf einer Seite mit weißer Hintergrundfarbe zu bekommen. Das Bild kippte in der Monitormitte förmlich nach hinten weg. Dieser Effekt tritt allerdings nur bei extremen Farbkombinationen auf und kann notfalls durch Umschaltung der Darstellungsart in den Schwarz/Weiß-Modus beseitigt werden (Attribute setzen/löschen). Auch soll laut Commodore das Modul nur mit einigen Monitoren der Serie VC-1701 nicht funktionieren. Mit Farbfernsehgeräten oder Monochrom-Monitoren trat dieses Problem nicht auf. Letztere haben allerdings Probleme, die rote Schrift auf schwarzem Hintergrund bei der Auswahlzeile für Sonderfunktionen darzustellen. Doch hilft auch hier ein Druck auf die Taste 'f4', der den schwarz/weiß-Modus aktiviert.

## BTX NIX FIX

Trotz dieser insgesamt positiven Bewertung, von Schwächen in der Dokumentation und dem Monitorproblem einmal abgesehen, sollte sich der potentielle Benutzer die Anschaffung eines Btx-Moduls gründlich überlegen. Zuerst einmal: Btx ist kein billiges Vergnügen. Auch wenn die Anschaffung in der beschriebenen Konfiguration vergleichsweise preiswert ist, kommen noch laufende Kosten dazu. Da ist erst einmal die Monatsmiete für den Btx-Anschluß von 8,- DM. Hinzu kommt der immer noch laufende 8-Minutentakt, der bei der langsamen Arbeitszeit bei der Datenübertragung und dem Bildschirmaufbau sicherlich häufiger zuschlagen wird. Die Daten erreichen den Rechner mit einer Geschwindigkeit von 1200 Baud, vom Heimgerät wird mit mageren 75 Baud gesendet. In der Hinsicht erscheint das Vorhandensein eines PAUSE-Kommandos, mit dem der Benutzer die Geschwindigkeit von Btx noch weiter nach Belieben verlangsamen kann, wie ein Aprilscherz (In Verbindung mit Makros sind so stundenlange Pausen möglich).

Wollen Sie alle Möglichkeiten, die Ihnen Btx bietet, ausnutzen, werden Sie sich nicht nur mit den zahlreichen gebührenfreien Seiten beschäftigen, sondern auch gebüh-

renpflichtige Seiten benutzen müssen. Nach einem Hinweis, daß die nachfolgende Seite etwas kostet, kann man die Abfrage mit der Eingabe einer '19', dem Kürzel für 'JA', bestätigen und bekommt den entsprechenden Betrag auf der Btx-Rechnung angeschrieben, die dann am Monatsende ins Haus flattert. Zwar liegen diese Seitenkosten meist im Pfennigbereich, doch summieren sich diese Kleinbeträge je nach Einsatzhäufigkeit mit der Zeit zu größeren Summen. So kostet in einer Dialogbox, in der man sich mit Teilnehmern aus der ganzen Bundesrepublik in Verbindung setzen kann, das Absenden oder Lesen einer Nachricht 9 Pfennige. Welche Beträge hier bei einem Gespräch zusammenkommen können, mag sich jeder selbst zusammenrechnen . . .

## Mit elektronischen Grüßen . . .

Darüber hinaus gibt es auch die Möglichkeit, jedem Mitbenutzer von Btx eine Nachricht direkt zukommen zu lassen (Portokosten 0,20 DM). Ein elektronischer Briefkasten steht jedem Benutzer zur Verfügung. Diesen können Sie jederzeit nach eingegangenen Briefen abfragen. Leider steht für den Text nur eine Bildschirmseite zur Verfügung, von der noch die Zeilen für Anschrift und Absender abzuziehen sind. Umfangreiche Texte lassen sich somit nur übertragen,

wenn man diese auf mehrere Briefe verteilt. Für besondere Anlässe existiert sogar ein Grußkartenservice, mit dem sich Kurznachrichten mit grafischen Symbolen verzieren lassen.

Eine interessante fast gebührenfreie Anwendung, weil ab und zu mal 30 Pfennige bezahlt werden müssen, stellt das elektronische Telefonbuch 'ETB' dar, in dem man sich Telefonnummern aus dem ganzen Bundesgebiet zusammensuchen kann und sich so einen Anruf bei der Auskunft erspart. Notwendig ist die Eingabe des Namens und des Wohnortes der Person. Auch eine Suche im Nahbereich des angegebenen Ortes ist auf Wunsch möglich, falls man sich nicht sicher ist, ob die gesuchte Person möglicherweise in einem Vorort mit anderer Postleitzahl wohnt. Ebenso wurde auch das Branchenfernsprechbuch, die Gelben Seiten, auf Bildschirmformat umgesetzt. Die Suche einer Person nach gegebener Telefonnummer ist übrigens nicht möglich. Datenschutz geht vor.

So vielfältig wie die Zahl der Anbieter sind auch die Angebote. Btx lädt ein zum 'Herumschmökern' auf verschiedenen Textseiten, so wie man dies zu anderen Zeiten mal mit Büchern getan haben soll. Spezielle Wettervorhersagen mit Wetterkarte sind ebenso abrufbar wie Meyer's Taschenlexikon in 10 Bänden oder die aktuellen Nachrichten direkt von einer Presseagentur. Bestellungen bei Versandhäusern ermöglichen

den Einkaufsbummel vor dem eigenen Bildschirm. Firmen aus allen Branchen warten mit 'Produktinformationen' (klingt besser als Werbung) auf und informieren über Firmengeschichte und Tätigkeit.

## Btx machts möglich — aber auch nötig?

Abgesehen davon, daß sich dieses Medium in Zukunft sicher noch weiterentwickeln wird, scheinen im Moment Zweifel am Nutzen angebracht, kann man sich doch die meisten Informationen auch aus anderen Quellen beschaffen, und sei es aus dem Videotext, der zwar keinen echten Dialog ermöglicht, dafür aber auch in der Anschaffung und im Unterhalt preiswerter ist. Doch wenn etwas wirklich Spaß macht, und sei es nach dem Motto „Warum einfach, wenn es auch umständlich geht?“ ist eine Anschaffung in der Praxis wohl selten zu teuer. Wenn dieser Spaß dann noch den Anschein des Nutzens mit sich bringt, ist eine Rechtfertigung meist schnell gegeben. Bestes Beispiel dafür sind die zahlreichen „Computeranwendungen“ im Heimbereich, die sich in der Praxis oft sehr viel ökonomischer mit Bleistift und Papier realisieren ließen. Doch von solchen Nebensächlichkeiten einmal abgesehen — wenn man Zeit mitbringt, kann Btx wirklich eine Menge Spaß machen. Und das ist doch schon eine ganz Menge. F. Börncke/kfp

### Buchkritik

Jürgen Baums

## Das große Buch zu Btx

DATA BECKER GmbH  
Düsseldorf 1987  
ISBN 3-89011-056-8  
518 Seiten, 69,- DM

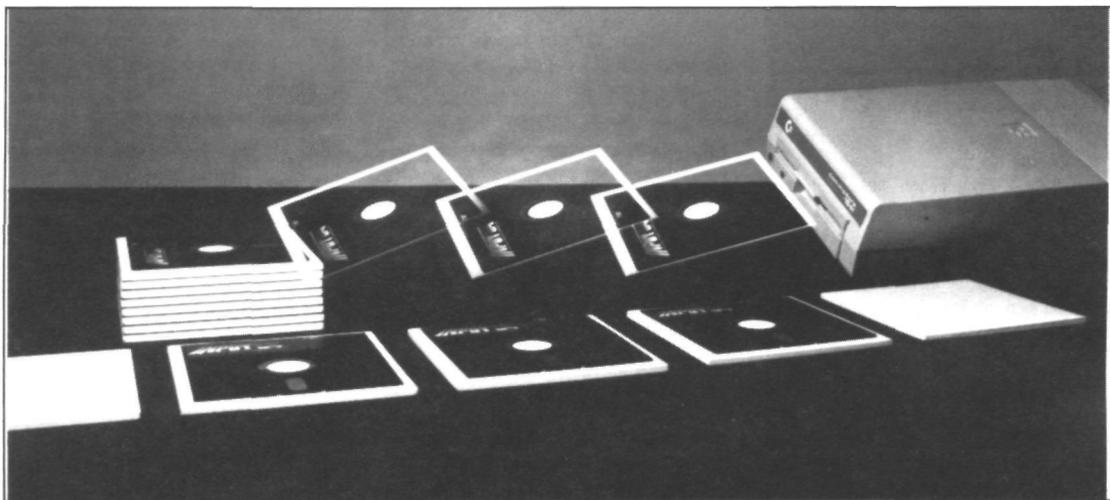
Eine umfassende und übersichtliche Darstellung des Informationssystems Btx bietet dieses Buch sowohl Anfängern als auch Fortgeschrittenen. Wer noch gar nicht weiß, worum's geht, erhält auf knapp 50 Seiten

eine einführende Darstellung über Leistungsmerkmale, Systemtechnik, Benutzung und Kosten des Btx-Systems. Hilfreich für „nur“ anwendungsorientierte Benutzer sind auch Abschnitte wie der umfangreiche Datenbankführer und die Vorstellung der gängigen Btx-Anwendungen.

Aber auch der technisch weitergehend Interessierte findet von der Erklärung „Wie funktioniert ein Btx-Decoder?“ über eine Aufschlüsselung des CEPT-Standards bis zu den diversen Schnittstellenbelegungen das, was zum tieferen Einstieg in dieses Medium an Know-how wichtig ist. Abgerundet wird die Informationsfülle durch einen Blick über die Grenzen: eine Kurzvorstellung des französischen Btx-Pendants Minitel und eine Übersicht über den Stand

der Informationssysteme anderer europäischer Länder.

Die einzige Kritik an diesem Buch bezieht sich auf das unkritische Verhältnis des Autors zu Btx selbst. Die bisherigen Versuche, dieses System massenweit zu verbreiten, werden nämlich unter Fachleuten als gescheitert angesehen: in knapp drei Jahren wurden nicht einmal 100 000 Teilnehmer angeschlossen. Neben den in der Einführungszeit zu hohen Kosten wichtigster Grund für den bisherigen Flop: der nervtötend zeitraubende Bildschirmaufbau. Darüber schweigt sich das Buch allerdings aus, stattdessen wird prophezeit, daß „Btx bald so selbstverständlich und verbreitet sein“ wird, „wie heute das Auto oder das Telefon“. JS



# Ablegen, auswählen, zugreifen

## Indexsequentielle Dateiverwaltung UniDat

Bekanntlich führen viele Wege nach Rom, etwa ebensoviele Möglichkeiten gibt es, ein Programm zu erklären. Ein Weg beispielsweise: ein Zettel, auf dem die Funktionstastenbelegung und die Bedeutung einiger Sonderzeichen vermerkt ist. Wenn man sich mit Dateiverwaltungen einigermaßen auskennt, reicht das bei UniDat aus, da das Programm weitgehend menügesteuert ist. Zudem hat der Programmierer, Michael Hanke, mit Benutzerhinweisen nicht geizigt. Einen solchen „Zettel“ finden Sie unter der Überschrift „Kurzübersicht Tastenbelegung“.

Spätestens dann, wenn es zu Schwierigkeiten kommt, ist allerdings etwas Hintergrundwissen von Nöten. Die nötigen Informationen zur Funktion des Programms finden Sie im Kasten „Wie es funktioniert?“ – wo auch sonst. Für diejenigen, die sich zum ersten Mal mit so einem Programm beschäftigen, beginnt ein einführender Lehrgang – auch eine der vielen Erklärungsmöglichkeiten – gleich hier an dieser Stelle,

**„UniDat“ ist die Abkürzung für „universelle Dateiverwaltung“ – ein universell einsetzbares, menügesteuertes Dateiverwaltungsprogramm mit frei definierbarer Eingabemaske. Das Programm arbeitet diskettenorientiert und kann bis zu 182 000 Zeichen in maximal 2 000 Datensätzen verwalten. Der schnelle Zugriff auf einzelne Datensätze wird durch Indexlisten unterstützt.**

und die einzelnen Menüpunkte sind im Kasten „Menüs, Funktionen, Formate“ erklärt.

### Vorarbeiten

Obendrein kann man an einer fertigen Datei (erzeugbar mit dem zweiten Modul auf dieser INPUT-Diskette) herumprobieren, nämlich der mit den Jahresinhaltsverzeichnis der letzten drei Jahre. Na denn:

UniDat muß wie üblich mit CTRL-S auf eine eigene Diskette überspielt werden. Dann

kann man das Programm von dort laden und mit RUN starten. Gehen wir einmal vom „Urzustand“ aus: eine neue Datei soll angelegt werden, beispielsweise ein Adreßverzeichnis. Zunächst muß die benutzte Floppy eingestellt werden, zwischen 1571- und 1541/1570-Modus kann mit der Leertaste hin- und hergeschaltet werden. Wie auch bei späteren Wahlmöglichkeiten erscheint das Ausgewählte invertiert, bestätigt wird mit der RETURN-Taste. Außer den Tasten „Space“ (Leertaste) und RETURN sollten Sie sich ab sofort merken, daß die Funktionstaste F2 immer eine Aktion abbricht und ein Menü „höher klettert“ in Richtung Eingangsmenü (Neudeutsch: Level-up-Funktion). F7 ist neben RETURN die Taste, die nach einer gültigen Eingabe eine Aktion auslöst.

Im Eingangsmenü wählen Sie mit Cursorhoch/-runter den Punkt „Datei einrichten“ aus und drücken RETURN. Wenn Sie im Untermenü noch einmal „Datei einrichten“

wählen (wieder Cursor-Taste und RETURN), befinden Sie sich im Masken-Editor. Eine Schreibmarke steht oben links in Zeile 0, Spalte 0. Dieser „Hilfs-Cursor“ kann mit den Cursor-Tasten bewegt werden, er verhält sich etwas anders als der normale Bildschirm-Editor des C64 – ausprobieren!

Bild 1 zeigt, wie eine Maske für eine Adreßdatei aussehen könnte. Der Pfeil nach oben

bezeichnet jeweils Anfang und Ende eines Datenfeldes. Die feinen Linien unter den Feldern erreicht man über die Tasten-Kombination **C-T**.

## Maskenzwang

Die fertige Maske wird mit F7 auf Diskette gespeichert, dieser Wunsch muß mit F1 be-

stätigt werden. Das Programm prüft dann zunächst, ob die Maske den Anforderungen genügt, mögliche Fehler sind: ungerade Anzahl von '}', zu lange Datenfelder (maximal 80 Zeichen), zu viele Datenfelder (maximal 20), kein Datenfeld definiert. Ist alles in Ordnung, muß ein Dateiname angegeben werden, naheliegender Weise geben wir „adresdatei“ mit abschließendem RETURN ein.

## Wie es funktioniert . . .

Vor der Erklärung von Verfahren wie der „indiziert sequentiellen Zugriffsmethode“ ist erst einmal die Klärung einiger Begriffe notwendig, um nicht aneinander vorbei zu reden. Unter einer **Datei** versteht man „für eine bestimmte Aufgabe beziehungsweise unter einem bestimmten Gesichtspunkt zusammengestellte Daten in einem äußeren Speicher einer Datenverarbeitungsanlage“<sup>1</sup>. Die Datenverarbeitungsanlage ist der C64, der „äußere Speicher“ in unserem Fall die Diskettenstation. Damit ist nicht entschieden, ob es sich um ein Programm, einen Text oder eben Daten für eine Dateiverarbeitung handelt. Da es uns hier nur um letzteres geht, soll der Begriff „Datei“ weiter eingengt werden: Eine Datei besteht aus einem oder mehreren Datensätzen.

Der Begriff **Datensatz** wird am ehesten durch den Vergleich mit einer Handkartei mit einzelnen Karteikarten deutlich. Die komplette Handkartei entspricht der Datei, eine einzelne Karteikarte einem Datensatz. Ein Datensatz ist noch einmal unterteilt in **Datenfelder**, etwa Name, Anschrift und so weiter. Der Aufbau aller Datensätze, also die Anzahl der Datenfelder und die Länge jedes Datenfeldes, einer Datei ist gleich.

Diese Datensatz-Struktur wird bei Unidat im Masken-Editor festgelegt, das Programm „merkt“ sich die entsprechenden Informationen in einem Disketten-File mit dem gewählten Dateinamen und dem Präfix 'y-'. Die „eigentliche“ Datei, die die eingegebenen Datensätze enthält, ist eine relative Datei mit dem Präfix 'z-'. Bei einer relativen Datei ist es möglich, auf jeden Datensatz (jeden Record) durch An-

gabe der spezifischen Nummer, die jedem Record zugeordnet ist, zuzugreifen.

Die Datensätze werden in der Reihenfolge der Eingabe nacheinander in diese relative Datei geschrieben. Der Genauigkeit halber sei noch gesagt, daß erst ab dem dritten Datensatz „echte“ Daten abgelegt werden, die ersten beiden Records benutzt das Programm zur Speicherung von Informationen über die Datei.

Nun ist es nicht damit getan, die Namen und Anschriften seiner Freunde auf Diskette zu lagern. Der Vorteil gegenüber einer Handkartei ist ja gerade der, daß man ein Programm hat, das die Datei nach einem bestimmten Begriff durchsucht, und der Benutzer dies nicht länger selbst tun muß. „In meinem Karteikasten habe ich die Anschriften alphabetisch nach Namen geordnet und finde jeden in Nullkommanichts wieder!“ werden Sie dagegen einwenden. Stimmt!

Aber was ist, wenn Sie einen Zettel mit einer Telefonnummer darauf in der Hosentasche finden und gerne wüßten, wie der zugehörige Teilnehmer heißt? Dann müßte Karte für Karte durchgesehen werden, es sei denn, sie hätten alle Karteikarten doppelt, die einen nach Namen, die anderen nach Telefonnummern geordnet. Und wenn Sie dann noch wissen wollten, wer von Ihren Freunden in München wohnt, müßte ein dritter Karteikasten her . . . Ach was! Natürlich würde man dann einen Katalog anlegen und die Karteikarten nur noch nummerieren. Im Katalog gäbe es dann verschiedene Stichwortverzeichnisse, eines nach Postleitzahlen geordnet, eines nach Namen und so weiter.

So löst auch Unidat das Problem, nämlich durch **ISAM**. Auf deutsch heißt ISAM

auch **ISZM**. Die deutsche Abkürzung steht für „indiziert sequentielle Zugriffsmethode“, die englische für „indexed sequential access method“. Damit ist folgendes gemeint: Man verzichtet darauf, die Datensätze selbst zu ordnen. Stattdessen wird für ein bestimmtes Datenfeld eine Indexliste angelegt, in der verzeichnet ist, wo sich der entsprechende Datensatz auf Diskette befindet. Geordnet wird nur diese Liste.

Bei Unidat sieht das so aus, daß die ersten beiden Zeichen des zum Indexfeld erklärten Datenfeldes alphabetisch geordnet vorliegen, dahinter steht die Nummer des Records, in dem der Datensatz zu finden ist. Diese Dateien sind auf Diskette mit den Präfixes 'a-' bis 't-' vor dem Dateinamen bezeichnet, 'a' entspricht dem ersten Datenfeld, 'b' dem zweiten und so weiter. Das aktuelle Indexfeld wird immer komplett in den Rechner geladen, deswegen können auch große Mengen von Datensätzen schnell durchsucht werden.

Was passiert aber, wenn zwei Indexfelder dieselben Anfangsbuchstaben haben? Etwa „Meyer“ und „Meier“? Dann müssen diese beiden Einträge der Reihe nach durchsucht werden, und so kommt auch der Begriff „sequentiell“ ins ISZM. Denn sequentiell bedeutet in der Datenverarbeitung nichts anderes als „der Reihe nach“.

Um noch einmal auf den Vergleich EDV-Datei/Karteikasten zurückzukommen: Die Arbeitersparnis besteht vor allem darin, daß der Rechner das Katalogisieren sozusagen „automatisch“ erledigt. JS

<sup>1</sup> zitiert nach: Müller/Löbel/Schmidt  
Lexikon der Datenverarbeitung  
Siemens AG, Berlin/München 1986

Indexfeld soll wie vorgegeben Feld '1' sein, also der Name. Natürlich soll die Maske gesichert werden, also wird 'j' mit RETURN bestätigt. Die Datendiskette liegt bereits im Floppy-Schacht, es kann losgehen. Diese Datei, die die Informationen für die Ein-/Ausgabemaske enthält, wird unter dem Namen „y-adressdatei“ auf Diskette gespeichert. Anschließend soll auch gleich die Datendiskette erstellt werden, die Vorgabe für die Anzahl der maximalen Datensätze wird mit RETURN übernommen. Dann hat man circa fünf Minuten Teepause, während das DOS Satz für Satz auf Diskette anlegt. Diese Datei enthält später die eigentlichen Daten, sie hat den Namen „z-adressdatei“.

## Gut angelegt

Mit F2 und RETURN geht es zurück ins Eingangsmenü. Freundlicherweise gibt das Programm vor, was jetzt zu tun ist: die Datei sichern. Diese Vorgabe von UniDat sollte man übrigens immer befolgen, sie dient der Datensicherung beziehungsweise der Datensicherheit. Erzeugt wird eine sequentielle Datei namens „a-adressdatei“. Diese Indexdatei enthält Informationen zum späteren Wiederauffinden der Daten.

Um nun endlich Namen, Adressen und Telefonnummern eingeben zu können, wird der Menüpunkt „Datei pflegen“ ausgewählt. (Mit RETURN, was künftig stillschweigend vorausgesetzt wird.) Zu sehen ist die zuvor erstellte Maske, allerdings ohne die 't', diese dienten nur zu Markierungszwecken bei der Maskenerstellung. In der Menüleiste in der unteren Bildschirmzeile kann mit der Leertaste oder mit Cursor-links-/rechts zwischen „Eingeben“, „Ändern“, „Suchen“ und „Loeschen“ hin- und hergeschaltet werden.

```

*****
*           A D R E S S E N           *
*****
* Name      :t _____ t *
* Strasse   :t _____ t *
* PLZ/Ort   :t _____ t *
* Telefon   :t _____ t *
*****

```

**Bild 1:** So sieht die Maske der Übungsdatei im Maskeneditor aus. Die Pfeile markieren die Datenfelder.

Zu Ändern gibts noch nichts, erstmal wollen die Daten eingegeben sein. Nach der Auswahl läuft das Laufwerk an, die leuchtende Floppy-LED zeigt, daß die Datei geöffnet ist. Im ersten Feld (Name) werden Eingaben erwartet. Zwischen den Feldern wird mit RETURN oder Cursor-hoch/-runter gewechselt. Ist alles korrekt eingegeben, werden durch Betätigen von F7 die Daten auf Diskette übertragen. Danach befindet man sich wieder in der Auswahl „Eingeben“, „Ändern“ und so weiter.

## Total erfaßt

Wählt man erneut „Eingeben“, steht der Cursor wieder im ersten Datenfeld. Die noch vorhandenen Eingaben können überschrieben oder mit F1 gelöscht werden. Geben Sie, um dieses Training weiterführen zu können, erst einmal zehn bis zwanzig Datensätze ein, achten Sie darauf, daß die Anfangsbuchstaben der Namen möglichst gut im Alphabet gestreut sind. Jeder Datensatz muß mit F7 auf Diskette übertragen werden. Der Modus „Datei pflegen“ wird mit F2 verlassen, das Verlöschen der roten LED an der Floppy signalisiert, daß die relative Datei geschlossen wurde. Die stille Aufforderung „Datei sichern“ sollte man unbedingt befolgen. Gesichert wird jetzt nämlich die Indexdatei, um die abgelegten Daten schnell wiederfinden zu können. Jetzt könnte man die Arbeit mit UniDat durch Ausschalten des Rechners beenden, alle Arbeitsergebnisse sind auf Diskette festgehalten und Teil 1 des Trainings abgeschlossen.

## Gesucht — gefunden

Im zweiten Teil der Übung geht es um die Auswertung einer vorhandenen Datei, nämlich der Adreßdatei. Nehmen wir an, Sie hätten nach dem ersten Trainingsteil den Rechner ausgeschaltet und UniDat am nächsten Tag neu geladen. (Den gleichen Effekt hat übrigens die Betätigung der Tastenkombination RUN/STOP-RESTORE.) Nach der Auswahl des Floppy-Modus geht es weiter mit „Datei pflegen“, und der Name der Datei wird erfragt, „adressdatei“ also.

Da zur Zeit nur Indexfeld Nummer 1 existiert, wird natürlich dieses geladen. Der erneute Wunsch, die Datei zu „pflegen“, führt

## Kurzübersicht Tastaturbelegung

F1	Rahmenfarbe Maske
F2	Abbruch/„Level up“
F3	Hintergrundfarbe Maske
F7	Aktion auslösen/ Daten übergeben
Leertaste	Auswählen
Cursor-Tasten	Auswählen
RETURN	Auswahl bestätigen
RUN/STOP-	Neustart/
RESTORE	Dateiwechsel

jetzt auf die schon bekannte Seite mit der Ein-/Ausgabemaske und der Menüleiste „Eingeben“ und so weiter. Wie viele Einträge hat Ihre Datei, deren Namen mit 'A' beginnen? Das können wir Ihnen natürlich nicht sagen. Wenn Sie aber die Option „Suchen“ aufrufen, im Namensfeld 'A\*' eingeben und die Suche mit F7 auslösen, werden Sie dies schnell feststellen. Der Stern ist ein sogenannter Joker, das heißt, alle Buchstaben nach dem 'A' werden nicht verglichen. Gefunden würden also „Aasgeier“, „Abelmann“ und „Azteke“.

## Ein Schritt vorwärts

Jeder gefundene Datensatz wird angezeigt. Die voreingestellte Möglichkeit '+' bedeutet, daß nach dem alphabetisch nächsten Datensatz, auf den der Suchbegriff zutrifft, gesucht wird, '-' bedeutet sinngemäß das Gegenteil, '0' beendet den Suchvorgang. '+' und '-' funktionieren so lange, wie es in der jeweiligen Suchrichtung noch passende Datensätze gibt. Ansonsten erscheint die Meldung „Kein Datensatz vorhanden!“. Umschalten wie immer mit der Leertaste, mit RETURN auswählen.

Wollen Sie zum Beispiel einen bestimmten Datensatz ändern oder löschen, wird dieser zunächst durch die Funktion „Suchen“ aus der Datei in den Rechner geholt. Nach dem Abbruch der Suche mit '0' kann dieser Datensatz dann gelöscht werden — sicherheitshalber fragt dann das Programm noch einmal nach. Oder Sie können durch Aufruf von „Ändern“ Schreibfehler korrigieren, eine neue Telefonnummer eintragen oder ähnliches. An die Datei übergeben werden

Forts. Seite 10

## Menüs, Funktionen, Formate

### Grundsätzliches

Nach Starten des Programms und Wahl des Floppy-Modus befindet man sich im Eingangsmenü. Mit dem Cursor oder der Leertaste und anschließendem RETURN gelangt man in das gewünschte Unterprogramm. Befindet sich keine Datei im Rechner, so sind nicht alle Unterprogramme zugänglich. Die Tastenkombination RUN/STOP und RESTORE löscht alle Dateien im Rechner und startet das Programm neu.

Bestimmte Tasten haben in allen Unterprogrammen eine gleichbleibende Funktion. Mit der Leertaste und/oder den Cursor-Tasten wird zwischen verschiedenen Möglichkeiten gewählt, mit RETURN bestätigt, F2 bricht einen laufenden Programmteil ab und kehrt auf die „nächsthöhere“ (oder vorherige) Menüebene zurück, F7 startet eine Aktion und/oder übergibt gültige Daten.

### Floppy-Modus

Um die größere Speicherkapazität der Floppy 1571 auszunutzen, kann nach Programmstart auf den 1571-Modus geschaltet werden. Ist keine Diskette im Laufwerk, so ist ein Blinken der Floppy-LED normal. Die Datendiskette muß vorher in diesem Modus formatiert werden und kann nicht mehr auf 1541-Laufwerken benutzt werden. Eine 1571 läßt eine maximale Dateigröße von 182 880 Zeichen zu, auf einer 1541/1570-Diskette lassen sich 162 000 Bytes unterbringen.

### Datei einrichten

Zuerst gelangt man in den Maskengenerator. Hier wird die Ein-/Ausgabemaske erzeugt, die später zum Arbeiten mit der Datei benutzt wird. Dazu steht fast der komplette Bildschirm-Editor zur Verfügung. Datenfelder werden durch einen

Pfeil (↑) gekennzeichnet. Jedes Datenfeld muß zwei Pfeile haben, einer kennzeichnet den Beginn, der andere das Ende des Datenfeldes. Es können maximal 20 solcher Datenfelder erstellt werden. In der fertig erstellten Maske erscheinen diese Pfeile später nicht mehr. Da die Pfeile zum Datenfeld gehören, ist das kleinstmögliche Datenfeld 2 Zeichen lang, es darf maximal 80 Zeichen enthalten. Der ganze Datensatz kann bis zu 20 solcher Datenfelder umfassen und bis zu 254 Zeichen lang sein. Diese Zahl reduziert sich pro Datenfeld um jeweils 1 Zeichen; so sind bei 10 Datenfelder maximal 244 Zeichen pro Datensatz möglich.

Um ein Datenfeld bei der späteren Arbeit in der Eingabemaske wiederzuerkennen, ist es sinnvoll, dieses kenntlich zu machen (zum Beispiel durch Unterstreichen mit grafischen Zeichen). Die Bildschirm- und Rahmenfarbe können bei der Erstellung der Maske durch die Tasten F1 und F3 geändert werden. Mit F7 beendet man die Arbeit mit dem Maskengenerator.

Die erstellte Maske wird nun gesichert. Sind beim Entwurf der Datenfelder Fehler gemacht worden, wird dieses vom Programm erkannt. Jetzt wird die Diskette für die Datei vorbereitet. Es sollte eine leere, formatierte Diskette als Datendiskette verwendet werden.

Nach der Wahl eines Indexfeldes gibt der Rechner die maximal mögliche Anzahl von Datensätzen für die erstellte Datei vor. Es wird dabei auch der freie Platz auf der Diskette berücksichtigt. Ist diese Zahl ermittelt worden, schreibt der Rechner die entsprechende Anzahl von Leersätzen auf die Datendiskette.

### Datei pflegen

Über diesen Menüpunkt kann eine Arbeitsdatei bestimmt und das entsprechende Indexfeld geladen werden; entweder nach dem Laden von UniDat oder nach RUN/STOP-RESTORE.

Wenn bereits eine Datei geladen ist, erscheint die zuvor erstellte Eingabemaske. Mit der Leertaste oder den Cursor-Tasten wird der gewünschte Programmpunkt angewählt. Taste F8 sendet eine Hardcopy des Bildschirms zum Drucker.

INGEBEN: Der Cursor erscheint am Anfang des Indexfeldes. Dieses Feld muß einen Eintrag erhalten. Mit F1 kann die Eingabezeile gelöscht werden. CLR/HOME löscht alle Eingabefelder. Nach erfolgter Dateneingabe wird über F7 das Abspeichern ausgelöst.

SUCHEN: Als Suchkriterium kann ein Datenfeldeintrag oder der Joker(\*) verwendet werden. Beide Möglichkeiten können kombiniert und in jedem Datenfeld eingesetzt werden. Ausgelöst wird das Suchen durch die F7-Taste. Der Suchvorgang mit Hilfe des Indexfeldes dauert durchschnittlich ungefähr drei Sekunden. Der Zugriff über andere Felder ist erheblich langsamer, da dann jeder Satz eingelesen und geprüft werden muß; so ergeben sich Suchzeiten von bis zu zwei Minuten.

Durch '+' wird mit gleichen Suchkriterien alphabetisch aufsteigend, durch '-' fallend gesucht. Die Auswahl '0' oder 'F2' beendet den Suchvorgang. Danach ist das Bearbeiten des gefundenen Datensatzes möglich. Wenn mit 'F2' der Suchvorgang unterbrochen wurde, kehrt das Programm an den Ausgangspunkt zurück, das heißt, es kann vom zuletzt gefundenen Datensatz aus weitergesucht werden.

ÄNDERN/LÖSCHEN: Soll ein gefundener Datensatz geändert oder gelöscht werden, wird der Suchvorgang mit 'F2' unterbrochen oder mit '0' beendet. Der gefundene Datensatz kann nun entsprechend bearbeitet und über 'F7' neu auf Diskette abgelegt werden. Der Ausstieg aus den einzelnen Programmteilen erfolgt jeweils mit der F2-Taste.

Forts. Seite 10

die veränderten Daten wie bei „Eingabe“ durch F7. Natürlich kann die Datei durch „Eingeben“ auch um neue Einträge erweitert werden. Das Eingabefeld können Sie übrigens mit F2 verlassen, ohne die Daten zu übernehmen. Auch der ganze Programmteil wird mit F2 verlassen, anschließend muß die Indexdatei neu erzeugt und dann gesichert werden: „Datei sichern“ aufrufen.

## Drucken nach Maß

Jetzt sollen die Daten aber auch aufs Papier. „Datei drucken“ führt ins Druckmenü, zunächst einmal brauchen wir Druckparameter. Vorher werden noch die Dateiparameter ausgedruckt. Bild 3 zeigt, wie diese aussehen müssen. Diese Informationen sind hilfreich für die Berechnung der Druckparameter. Da wir diese erst erstellen

wollen, wird die Abfrage „Druckerparameter laden?“ mit Nein beantwortet.

Besitzer der weit verbreiteten Commodore-Drucker der MPS-Serie können die folgenden Vorgaben nach Druckeradresse und so weiter mit RETURN bestätigen und einen Testausdruck machen. Diese Voreinstellungen dürften auch für 90 Prozent aller anderen Druckermodelle gelten, ansonsten heißt

### Datei drucken

Um eine Datei drucken zu können, müssen zuerst die Druckparameter festgelegt werden. Da dies ein paar Berechnungen erfordert, ist der vorherige Ausdruck der Dateiparameter zu empfehlen. Außerdem sollte der angeschlossene Drucker angepaßt werden. Neben der Geräteadresse und der Sekundäradresse können Control-Codes eingestellt werden, die vor dem Ausdruck zum Drucker gesendet werden, etwa, um eine andere Schriftgröße einzustellen. Über die jeweiligen Parameter gibt das Druckerhandbuch Auskunft.

Um den Dateiinhalte formvollendet auf Papier bringen zu können, müssen die Druckparameter festgelegt werden, nämlich Anzahl der Zeichen pro Zeile und Datensätze pro Seite, wobei ein Datensatz über mehrere Zeilen gehen kann. Nach jeder so definierten Zeile wird ein Zeilenvorschub ausgegeben. Mit „Zeilenvorschub“ ist die Anzahl der Zeilenvorschübe nach jedem Datensatz gemeint (nicht die nach jedem Datenfeld!). Die Kopfzeile erscheint über jeder neuen Seite; das Ende der Kopfzeile wird mit dem „Klammeraffen“ (Ⓢ) markiert. Ist die gewählte Anzahl der Datensätze pro Seite erreicht, wartet das Programm auf einen Tastendruck, um dem Benutzer etwa Zeit zum Papiereinlegen zu lassen.

„Druck-Spaces“ sind die Abstände zwischen den Datenfeldern. Die zu einem Datenfeld gehörenden Leerschritte werden vor dem Datenfeld ausgegeben. Ein Minuszeichen (-) statt einer Zahl legt fest, daß das entsprechende Feld nicht gedruckt wird.

Zur Kontrolle des Aufbaus der zu druckenden Datei kann ein Testausdruck vorgenommen werden. Es ist sinnvoll, die Druckparameter abzuspeichern. Sie belegen lediglich einen Block und sollten sinnvollerweise mit auf der Datendiskette abgelegt werden. Nach Erstellung der Druckparameter kann der Dateiausdruck beginnen.

Das Programm bietet verschiedene Auswahlkriterien, die bei der Ausgabe der Datei berücksichtigt werden sollen. All diese Kriterien beziehen sich auf das erste Zeichen des durch das aktuelle Indexfeld bestimmten Datenfeldes.

Soll alles gedruckt werden, wählt man den Joker (\*). Nach Übereinstimmung mit dem Suchzeichen (alle mit 'A' beginnenden Felder etwa) wird durch "-" gedruckt, Bereiche können durch die Option „von-bis“ zu Papier gebracht werden. Außerdem existiert die Möglichkeit, nach dem Kriterium „kleiner“ (kl) oder „größer“ (gr) zu sortieren. Der Ausdruck kann mit der F2-Taste unterbrochen werden.

### Datei sichern

Nachdem eine Datei neu erstellt oder verändert worden ist, fordert das Programm zum „Datei sichern“ auf. Es wird das Hauptindexfeld gesichert, mit dem die nächste Bearbeitung erfolgen muß. Zuerst wird das Indexfeld sortiert. Dieser Vorgang dauert etwa 60 Sekunden, danach kann die Datensicherung erfolgen. Das alte Indexfeld wird dabei gelöscht. Das Sichern kann, soweit Platz vorhanden ist, auf der Datendiskette erfolgen. Sollte versehentlich das Indexfeld nicht gesichert worden sein, gelangt man nicht in den Bearbeitungs-

gang, es erscheint die Fehlermeldung „Indexfeld Fehler“. Der Index ist dann zu erneuern.

### Index erneuern

Sollte einmal das Indexfeld verloren gegangen sein oder soll ein anderes Feld als Indexfeld bestimmt werden, kann das in diesem Programmteil vorgenommen werden. Mit einem anderen als dem in der Maske festgelegten Indexfeld kann die Datei nur noch ausgewertet werden (Suchen und Drucken). Ist ein Indexfeld erneuert worden, befindet es sich automatisch im Rechner.

Etwas anderes ist übrigens der Wechsel auf ein anderes, bereits auf Diskette vorhandenes Indexfeld. Da dies programmtechnisch einem Neustart gleichkommt, betätigt man hierzu RUN/STOP-RESTORE und legt dann das gewünschte Indexfeld fest.

### Disk Befehle

In der Eingabezeile können die Befehle, die dem Floppy-Handbuch zu entnehmen sind, an die Diskettenstation gesendet werden. Die Befehlseingabe kann mit dem Zeichen ']' abgeschlossen werden. Erwartet wird nur der Befehlsstring selbst, also

N:DATEN,XY

zum Formatieren einer Diskette. Das Dollarzeichens (\$) bewirkt, daß das Inhaltsverzeichnis der Diskette auf dem Bildschirm ausgegeben wird. Das geht übrigens auch durch die Eingabe von '\$'; wenn im Programm nach dem Dateinamen gefragt wird. M. Hanke/JS

es jetzt: Das Druckerhandbuch hervorkommen.

Hoffentlich war „Alles OK“, denn erst jetzt wird's interessant. Geben Sie einmal ein: 20 Zeichen pro Zeile und 10 Datensätze pro Seite, einen Zeilenabstand von '1', als Inhalt der Kopfzeile „Adressdatei@“ (der '@' ist kein Druckfehler!) und '0' „Druckspaces“ bei allen Feldern. Gemeint sind hiermit immer die vor diesem Feld ausgegebenen Leerzeichen. Ein Testausdruck müßte dann vier untereinander liegende Reihen mit Doppelkreuzen unter der Überschrift „Adressdatei“ ergeben.

## Etikette nach Wahl

Wenn Sie jetzt dem Programm mitteilen, es sei nicht „Alles OK“, können die Druckparameter geändert werden. Mit F2 geht es zurück zur Eingabe der Zeichen pro Zeile, die wir in '80' ändern. Beim Bestätigen der anderen Angaben mit RETURN fällt auf, daß die mögliche Kopfzeile jetzt auch länger ist – logisch. Wenn Sie nun die Anzahl der Druckspaces bei den Feldern 2 bis 4 auf '60' setzen, ergibt sich bei einem Testausdruck genau das Druckbild wie vorher. Nach der Ausgabe der 20 Zeichen von Feld 1 wird jeweils der Rest der Zeile mit den 60 Leerzeichen aufgefüllt und ein Zeilenvorschub ausgegeben.

Man kann auch ein Feld beim Ausdruck übergehen, indem man ein '-' eingibt. So könnte man durch ein '-' bei Feld 4 die Telefonnummern weglassen und Etiketten mit Anschriften drucken. Experimentieren Sie ein bißchen mit diesen Möglichkeiten! Wenn Ihnen eine Möglichkeit wirklich gut gefällt, speichern Sie diese ab („Druckparameter sichern“).

A D R E S S E N	
Name	:Heinz Heise GmbH & Co KG
Strasse	:Heisterforstr. 7
PLZ/Ort	:3000 Hannover 61
Telefon	:0511-5352-0

**Bild 2: Die Maske im Einsatz der Dateiverwaltung. Hier wird ein Datensatz ausgegeben.**

Datei Parameter	
Anzahl der Felder	: 4
Laenge Feld 1	: 24
Laenge Feld 2	: 24
Laenge Feld 3	: 24
Laenge Feld 4	: 24
Laenge aller Felder	: 96

**Bild 3: Der Ausdruck der Dateiparameter hilft beim Erstellen der Druckparameter.**

## Wer soll's denn sein?

Dann kann es an das „Listen drucken“ gehen. Fünf Kriterien sind möglich, nach denen ausgedruckt werden kann. Alle Kriterien beziehen sich auf den ersten Buchstaben des aktuellen Indexfeldes. In unserem Beispiel ist dies immer noch Feld 1. Durch den Stern (auch hier gilt: Auswahl – Leertaste, Bestätigen – RETURN) werden alle Datensätze ausgedruckt; alle Datensätze, deren Anfangsbuchstabe in der ersten Hälfte des Alphabets liegt, werden durch „von-bis“ und die Angabe von 'A' als erstem und 'M' als zweitem Suchzeichen ausgedruckt; „kl“ bedeutet „kleiner“, „gr“ „größer“ und „=“ natürlich „gleich“.

Bevor wir das Training als beendet erklären, schnell noch ein paar Worte zu dem Menüpunkt „Index erneuern“. Es geht dabei um folgendes: Eine gezielte Suche nach Datensätzen wäre auch über ein anderes Datenfeld, etwa Feld 3, die Postleitzahl, möglich gewesen. Es kann aber bei größeren Dateien zu erheblichen Suchzeiten kommen, wenn nicht über das aktuelle Indexfeld gesucht wird, da dann jeder Datensatz eingelese und verglichen werden muß. Deswegen ist es sinnvoll, für jeden Datensatz ein eigenes Indexfeld anzulegen.

## Blick zur Diskette

Wählen Sie dazu den Punkt „Index erneuern“ im Hauptmenü aus, bestätigen den Dateinamen mit RETURN und geben als Nummer des Indexfeldes '3' an. Es wird nun eine zweite Indexdatei auf Diskette angelegt, mit der gezielt auf die Postleitzahlen in Feld 3 zugegriffen werden kann. Wenn Sie den Punkt „Disk-Befehle“ aufrufen und als Kommando '\$' mit der eckigen Klammer als

Abschluß () eingeben, können Sie im Directory feststellen, daß diese Datei „c-adressdatei“ heißt.

Das sollte als Einstiegswissen für den Umgang mit UniDat reichen. Um Erfahrungen im Umgang mit größeren Datenmengen zu machen, können Sie mit der Jahresinhaltsdatei aus dieser Ausgabe experimentieren. Diese muß erst UniDat-gerecht generiert werden, wie das geht, ist an anderer Stelle im Heft nachzulesen. Wenn Sie Ihren Umgang mit UniDat weiter perfektionieren wollen, sollten Sie auch die einzelnen Abschnitte zu den Menüs und „Wie es funktioniert“ lesen.

Michael Hanke/JS

## Training total

### Sammeldiskette: Englische GRAMmatik

Ab sofort beim Verlag erhältlich: Eine Zusammenfassung aller bislang erschienenen Folgen des interaktiven Lernprogramms „Englische GRAMmatik“.

Eine Diskette mit 800 Übungssätzen in 10 abgeschlossenen Einheiten zu den wichtigen Problemen der englischen Grammatik.

Erweitert um flexible Druck-Optionen und die Möglichkeit, eigene Übungen zu erstellen.

Aus dem Inhalt: „If-clauses“, „Reported Speech“, „Irregular plural forms“, „Personal and reflexive pronouns“, „Will-future and going-to-future“, „Prepositions of place, movement and time“, „Question tags and short answers“ u.v.a.m.

Preis: 19,80 DM inklusive Porto und Verpackung (nur gegen Verrechnungsscheck)

Bestelladresse:  
Verlag Heinz Heise GmbH & Co KG  
Postfach 61 04 07  
3000 Hannover 61

# Augenfällige Unterbrechungen

## Interrupt-Tips, Folge 2

In der letzten Folge konnten Sie bereits die beiden wichtigsten Prozesse kennenlernen, die sozusagen im Hintergrund ablaufen:

**IRQ** – Interrupt Request

**NMI** – Non maskable Interrupt

Diese Interrupt-Routinen (Unterbrechungs-Routinen) greifen in die laufende Verarbeitung ein, dies kann beispielsweise ein Programm oder der BASIC-Interpreter sein. Dadurch wird gewährleistet, daß die Tastatur abgefragt wird, die internen Uhren gewartet werden, auf Peripherie zugegriffen werden kann und vieles mehr. Die grafische Darstellung „Wegweiser zum Interrupt“ zeigt den Zusammenhang der benutzten Adressen.

Erst über diese Prozeduren lassen sich im C64 zeitkritische Abläufe verwirklichen. Neben den Interrupt-Routinen (Unterbrechungs-Routinen) gibt es jedoch noch andere Prozesse, die im C64 scheinbar gleichzeitig ablaufen. So wird der 6510-Prozessor von einem zweiten Prozessor entlastet, dem VIC (Video Interface Controller), der ge-

**Mehr als sieben Sprites, Text und Grafik auf einem Bildschirm, mehrere Rahmenfarben zugleich und andere animierende Effekte sind mit herkömmlicher Programmierung nicht erreichbar. Trotzdem können Sie einige dieser Spezialitäten auch in BASIC-Programmen einsetzen, vorausgesetzt, Sie sind bereit, sich ein wenig mit dem inneren Aufbau Ihres C64 zu beschäftigen.**

schickt die „Rechenpausen“ des 6510 für seine eigenen Aktivitäten nutzt.

## Anlagen

Der VIC ist ein 6569-Baustein, der ebenfalls aus der 65xx-Familie stammt und die Schnittstelle zur Bildschirmausgabe überwacht. Dies hat den Vorteil, daß der 6510 beim Bildschirmaufbau nicht mit Aufgaben

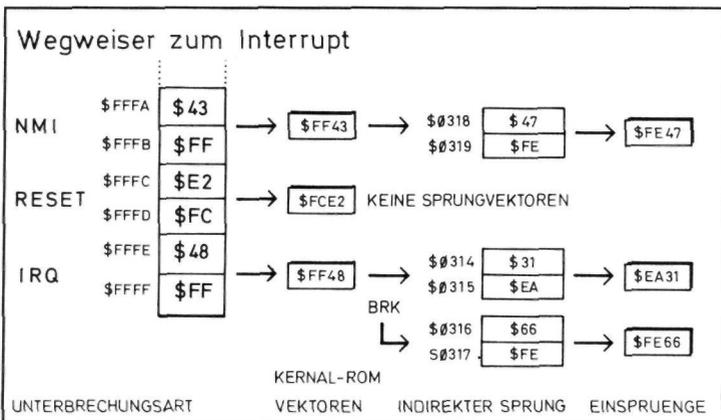
wie Speicherzugriffen, Zeichensatzauswahl und Sprite-Steuerung belästigt wird.

Der VIC muß hierzu ebenso wie die CPU (Central Processing Unit 6510) auf den Systembus (Daten- und Adreßleitungen) zugreifen. Da aber der VIC den Systemtakt erzeugt, hat er die CPU sozusagen in der Hand und kann die Lücken im Arbeitstakt des 6510 für seine Aktivitäten nutzen. Während einer solchen Takt-Pause trennt der VIC den 6510 vom Bus, bis die Zugriffe auf Video-RAM, Farb-RAM oder Zeichensatzgenerator erledigt sind, und gibt den 6510 erst danach wieder frei. Beide Prozessoren arbeiten somit im Wechsel, ohne sich gegenseitig zu stören.

Der Aufgabenbereich des VIC ist nicht gerade klein: Wie beim Fernsehempfang wird auch ein Computerbildschirm zeilenweise und Punkt für Punkt aufgebaut. Jeder dieser Punkte besitzt unterschiedliche Helligkeits- und Farbtintensität (Ein genaues Betrachten des Monitorbildes durch eine Lupe erbringt den Beweis). Da das Zeichnen eines Bildes innerhalb einer 1/25 Sekunde erledigt ist, hat unser Auge keine Chance, diesem Vorgang zu folgen, da es dazu einfach nicht reaktionsschnell genug ist. Diese „Trägheit“ läßt sich sogar noch nutzen, um spezielle Effekte zu erzielen. In der ersten Folge wurden bereits Eingriffsmöglichkeiten über den VIC erwähnt.

## Video – Video

Vom Betriebssystem wird der IRQ über CIA #1 (Complex Interface Adapter) gesteuert. Dies ist ein 6526-Baustein im C64, der das Zusammenspiel der einzelnen externen Geräte überwacht. Der VIC im C64 kann, wie in der letzten Folge bereits erwähnt, aber auch selbst einen Interrupt am 6510 auslösen.



Genau vier Unterbrechungsanforderungen können den VIC zu einem Interrupt veranlassen:

1. Kollision von Sprites untereinander
2. Kollision von Sprites mit Zeichen des Bildschirmhintergrundes
3. Erreichen einer bestimmten 'Rasterzeile'
4. Lichtgriffelimpulse

Welche der Ursachen einen IRQ auslösen soll, wird über sogenannte „Register“ des VIC gesteuert. Ein Register besteht aus einem Byte, also acht Bits. Es gibt Bits, die geschrieben werden können, und solche, die gelesen werden können.

Bei einem Schreibzugriff werden Funktionen gesteuert. Dem betreffenden Bit ist eine bestimmte Funktion zugeordnet, die durch Setzen oder Löschen ein- oder ausgeschaltet werden kann.

## Querverbindungen

Durch Auslesen einzelner Bits können Informationen abgerufen werden, die der Prozessor in den Registern hinterlegt hat, wobei jedem Bit eine bestimmte Bedeutung zugewiesen ist.

Für die IRQ-Steuerung des VIC sind folgende Register wichtig: Register 25 und Register 26, sowie die Register 18, 30 und 31.

Die Bedeutung der einzelnen Register können Sie der Register-Übersicht entnehmen.

Die Kollisions-Register können auch vom BASIC-Programm genutzt werden und sind schneller als die Abfrage der Sprite-Positionen und deren Vergleich. Direkte Programmierung der VIC-Interrupts von BASIC aus ist jedoch nicht möglich. Das Betriebssystem des C64, das Kernal, bedient in der IRQ-Routine nur die CIA-Interrupts, die VIC-Register werden nicht zurückgesetzt.

Verwenden Sie im Direktmodus den Befehl:

**POKE (53248+26),1**

Der C64 stellt anscheinend jede Tätigkeit ein, zumindest verweigert er jede Eingabe.

## Für den Maschinisten

Wollen Sie einen VIC-IRQ programmieren, gibt es zwei Möglichkeiten, dieses zu verwicklichen.

1. Ausschalten des System-Interruptes und Umschalten auf das eigene Interrupt-Programm
2. Sprung auf eigenes Interrupt-Programm und anschließende Verzweigung zum System-Interrupt

Bei der ersten Möglichkeit müssen Sie jedoch Tastaturabfrage oder andere Peripherie-Steuerungen selbst verwalten. Dies verlangt genaueste Kenntnis der Funktionsweise des „normalen“ Interrupt-Programms. Die zweite Möglichkeit erspart einige dieser Schwierigkeiten.

So wählt man häufig, wenn man einen Interrupt verarbeiten möchte, der vom VIC ausgelöst wurde, die zweite Variante. Auch hierbei gibt es zwei verschiedene Verfahren:

– Man fragt im eigenen Interrupt-Programm zuerst nach, ob die Unterbrechung vom VIC ausgelöst wurde. Ist dies der Fall, übernimmt erst einmal das eigene Programm die Kontrolle und übergibt anschließend an die ursprüngliche Interrupt-Routine.

– Durch Sperren der CIA-Interrupts, für CIA #1 zum Beispiel durch Setzen des Registers 13, entspricht der Adresse \$DC0D (56320+13), auf \$7F (127), entfällt die Abfrage, ob der Interrupt vom VIC ausgelöst wurde.

Grundsätzlich sind IRQ-Programme immer sehr zeitkritisch. Werden in solchen Programmen Abfragen durchgeführt, beispielsweise auf bestimmte Tasten, verändert sich die Dauer des Programmdurchlaufs. Dies macht sich vor allem bei Raster-Interrupts bemerkbar.

## Auf die Zusammenstellung kommt es an!

Damit Ihre IRQ-Routine und der System-IRQ ohne Störungen zusammenarbeiten, sollten Sie folgende Reihenfolge beim Erstellen eigener IRQ-Programme einhalten:

1. Alle laufenden IRQ-Anforderungen unterdrücken (SEI)
2. Eigene IRQ-Quellen definieren und freigeben
3. Vektor der Interrupt-Routine (\$0314/\$0315) auf die eigene IRQ-Routine setzen

## 4. IRQ-Anforderungen freigeben (CLI)

Wird nach dieser Initialisierung der IRQ-Routine ein IRQ an der CPU ausgelöst, wird automatisch das eigene IRQ-Programm gestartet.

Nur hier können Sie den Auslöser auswählen (CIA/VIC/RESTORE-Taste beispielsweise). Wurde der IRQ über VIC-Register 25 erzeugt, arbeiten Sie im eigenen Interrupt-Programm weiter, anderenfalls verzweigt das Programm zum „normalen“ System-Interrupt.

## Rasterstrahlen

Im C64 wird die aktuell beschriebene „Rasterzeile“ in den VIC-Registern 17/18 (53265/6) festgehalten. Im VIC-Register 17 (53265) wird allerdings nur das BIT 7 genutzt. Es können Werte zwischen 0 und 280 entstehen. Der sichtbare Teil des Monitorbildes, Zeile 1 bis 25, entspricht in etwa den Werten 50 bis 248. Diese Register lassen sich auch setzen. Die damit festgelegte Rasterzeile ist dann der Auslöser des Interruptes, falls dieser im Interrupt-Enable-Register (53274) freigegeben wurde (siehe Kollisions-Programm).

## Registerübersicht

### Register 25

Interrupt-Request-Register

Leseregister, enthält Information, welche Ursache einen IRQ ausgelöst hat.

Wichtig sind nur die Bits 0 – 3 und Bit 7. Alle anderen Bits sind mit '1' vorbelegt.

**Bit 0** – Rasterzeilen-IRQ (Register 18)

**Bit 1** – Sprite-Hintergrund-Kollision (Register 31)

**Bit 2** – Sprite-Sprite-Kollision (Register 30)

**Bit 3** – Lichtgriffelimpulse (der VIC hat hierfür eine spezielle Leitung)

**Bit 7** – Interrupt-Erkennung

Ist mindestens eins der ersten vier Bits (Bit 0 bis 3) vom VIC gesetzt worden und

das entsprechende Bit in Register 26 gesetzt, die Auslösung also eingeschaltet, wird beim Interrupt gleichzeitig Bit 7 auf '1' gesetzt.

### Register 26

Interrupt-Mask-Register

Schreib-Register, legt fest, welche Situation einen IRQ auslösen soll.

Die Bedeutung der Bits entspricht der von Register 25. Hier muß das entsprechende Bit gesetzt werden, soll beim Eintreten der gewünschten Ursache ein Interrupt stattfinden.

### Register 18

Rasterzeilennummer

Schreib-Register, hier steht die Nummer der Bildschirm-Rasterzeile, bei der ein IRQ

ausgelöst werden kann. Für Rasterzeilen größer 255 wird das Übertrags-Bit in Register 17, Bit 7 eingetragen.

### Register 30

Kollisions-Register

Lese-Register, informiert darüber, ob und welche Sprites sich berühren. Die Bits 0 bis 7 sind den Sprite-Blöcken 0 bis 7 zugeordnet. Eine '1' zeigt an, das eine Kollision stattfand.

### Register 31

Kollisions-Register

Lese-Register, gibt an welches Sprite ein Hintergrundzeichen berührt. Die Belegung der Bits entspricht Register 30.

Einzig die **⇐**-SHIFT-Taste schaltet noch zwischen Groß- und Kleinschrift um. Mit dem POKE-Befehl haben Sie den Raster-Interrupt zugelassen. Da bei jedem Bildschirmaufbau ein Raster-IRQ am VIC erzeugt wird und das VIC-Register 26 von der „normalen“ IRQ-Routine nicht zurückgesetzt wird, läuft die IRQ-Routine in eine Endlosschleife. Jedesmal nach der Abarbeitung springt der 6510 aufgrund der IRQ-Anforderung des VIC wieder in die IRQ-Routine. Nur über die Tastenkombination RUN/STOP-RESTORE gelangen Sie wieder in den BASIC-Interpreter.

Auf der letzten Bildschirmseite im Modul „64er Tips“ haben Sie wieder die Möglichkeit, ein einfaches Beispielprogramm mit CTRL-S abzuspeichern. Dieses Programm enthält zwei BASIC-Tools. Das erste überprüft die Sprite/Sprite-Kollision im Interrupt, das zweite schaltet die „Regenbogenfarben“ im Bildschirmrahmen ein.

Für Kenner und Besitzer des INPUT-Ass bietet INPUT 64 eine Source-Datei der Programme mit an, die ebenfalls über CTRL-S abgespeichert werden können, sozusagen zum Weiterbasteln.

In nächsten Folge erfahren Sie mehr über die IRQ-Steuerung durch die CIA-Bausteine. Als naheliegende Anwendung gibt es in den 64er Tips eine Software-Uhr als Dreingabe.

Thomas Stahmer/rh

## Assembler-Know-how für alle!

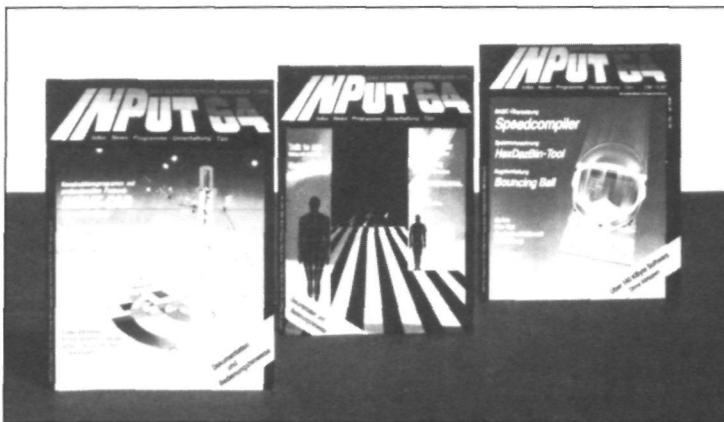
Ab sofort direkt beim Verlag erhältlich: Ein Leckerbissen für jeden Assembler-Programmierer und alle, die es werden wollen.

Eine Diskette mit dem Macro-Assembler INPUT-ASS aus INPUT 64, Ausgabe 6/86, und dazu

- der komplette Source-Code dieses Assemblers
- der Source-Code des Maschinsprache-Monitors MLM64plus aus INPUT 64, Ausgabe 11/87
- Library-Module: I/O-Routinen, Hex/ASCII/Dezimal-Wandlung, Multiplikation, Division
- Konvertierungsprogramme zur Format-Wandlung von PROFI-ASS- und MAE-Texten in das Source-Code-Format des INPUT-ASS

**Preis: 49,— zuzüglich 3,— DM für Porto und Verpackung (nur gegen V-Scheck)**

**Bestelladresse: Verlag Heinz Heise GmbH & Co KG  
Postfach 61 04 07 · 3000 Hannover 61**



## Jahresinhaltsverzeichnis INPUT 64 1985-1987

### Anwenderprogramme

<b>Dateikasten</b>	1/85
Berichtigung 3/85	
<b>Scriptor</b>	2/85
Nachtrag Drucker 4/85 & 5/85	
<b>Dictionary</b>	2/85
<b>Formelplotter</b>	3/85
<b>Kalender</b>	4/85
Mit Urlaubs-Optimierung	
<b>Einkommensteuer '84</b>	4/85
<b>Motorbike</b>	5/85
Sportmotoren berechnen	
Anpassung für PKW 9/85	
<b>Haushaltsbuchführung</b>	6/85
Private Ausgaben im Überblick	
<b>HiFi-Boxen-Berechnung</b>	7/85
Selbstbauboxen gut angepaßt	
<b>Reisekosten-Berechnung</b>	8/85
Mit Bahn-/Autokostenvergleich	
<b>Terminkalender</b>	9/85
Jahres- und Wochentermine planen	
<b>Planetarium</b>	11/85
Sternenhimmel auf dem Bildschirm	
<b>Funky-Drummer</b>	12/85
Digitaler Drum-Computer	
<b>Nutzen-Kosten-Analyse</b>	12/85
Entscheidungshilfe per C64	
<b>Lohnsteuer-Berechnung '85</b>	1/86
<b>MiniDat V1</b>	2/86
Version 1 mit INPUT-Inhalt '85	
<b>MatheGenie</b>	2/86
<b>TabCalc</b>	3/86
<b>ChipList</b>	4/86
IC's im Überblick	

<b>Fußball-WM '86</b>	5/86
<b>Bundesliga</b>	7/86
Berichtigung 10/86	
<b>Devisa</b>	7/86
Währungsumrechnung	
<b>AstroLog</b>	9/86
Planetenstände in Tabellen	
<b>LogikTab</b>	9/86
Logische Verknüpfungen	
<b>Farbcode</b>	9/86
Elektrische Bauteile bestimmen	
<b>INPUT-Calc</b>	10/86
Nachtrag 1/87; siehe auch 10/87	
<b>INPUT-CAD Teil 1</b>	11/86
Editor	
<b>INPUT-CAD Teil 2</b>	12/86
Speicherbefehle	
<b>INPUT-CAD Teil 3</b>	1/87
Druckerbefehle	
<b>INPUT-CAD Teil 4</b>	2/87
Editor-Erweiterung	
<b>INPUT-SAM/1</b>	8/86
A/D-Wandler	
<b>INPUT-SAM/2</b>	9/86
D/A-Wandler & Schalt-Interfaces	
<b>INPUT-SAM/3</b>	10/86
EPROM-Modulgenerator	
<b>INPUT-SAM/4</b>	11/86
MIDI-Interface	
<b>Kalorien-Berechnung</b>	12/86
Nährwerte ermitteln	
<b>Lohnsteuer '86</b>	1/87
<b>Julia</b>	2/87
Mandelbrot-Grafiken	

<b>Vokabel-Trainer</b>	3/87
Lernprogramm mit fertigen Dateien	
<b>Wärmebedarfsberechnung</b>	4/87
<b>CD-Manager</b>	6/87
Compact-Disc-Verwaltung	
Korrektur 8/87	
<b>INPUT-Calc64/128</b>	8/87
Berichtigung 12/87	
<b>INPUT-Graph</b>	9/87
Optische Daten-Präsentation	
<b>Electrobrief</b>	11/87
Botschaft auf Diskette	
<b>Kalender</b>	11/87
Geld sparen durch Eigendruck	
<b>Unidat</b>	12/87
Indexsequentielle Dateiverwaltung	
<b>Jahresinhalt 85-87</b>	12/87
Dateigenerator für Unidat	

### Simulations-Programme

<b>Talk to me</b>	5/85
Dialogprogramm à la ELIZA	
<b>Manager</b>	9/85
Berichtigung 11/85	
<b>Jetflight-Flugsimulator</b>	10/85
Training im Instrumentenflug	
<b>Wurfsimulator</b>	1/86
Ballistik-Programm	
<b>Life</b>	4/86
Mikroleben in der Entwicklung	
<b>Volkszählung '87</b>	5/87
Programme zum Problem der Deanonymisierung	

### Denkspiele

<b>Tower</b>	1/85
<b>Labyrinth</b>	2/85
<b>Reversi</b>	4/85
<b>Paulchen's Schiebung</b>	8/85
<b>Color Code</b>	8/85
<b>Superkaesi</b>	9/85
<b>Wortspiel</b>	10/85
<b>Super Memory</b>	11/85
<b>Kalaha</b>	12/85
<b>Memofix</b>	2/86
<b>Füll den Kreis</b>	5/86
<b>Ratefix</b>	7/86
<b>Gobang</b>	8/86
<b>Maximal</b>	9/86
<b>Dame</b>	10/86
<b>Vier gewinnt</b>	11/86
<b>INPUT-Schach</b>	12/86
<b>Kartenfuchs</b>	1/87
<b>Drei mal drei</b>	2/87

Tic-Tac-Toe	3/87
Golum	5/87
Pyramidon	5/87
Tiere-Raten	9/87
Farb-Basar	9/87
Chamaeleon	10/87
Das grosse Quiz	11/87
Triangel	12/87

## Text-/Grafik-Adventure

ARDF-Fuchsjagd	6/85
Via Roma	2/86
Ciron-3	3/86
Maya-Grab	7/86

## Video Games

Ballonjagd	1/85
Inferno	1/85
Supersonic	2/85
Biathlon	3/85
Hells Bells	3/85
Artemis	4/85
Break Out	5/85
Apple Frency	7/85
Tricky	7/85
Klicker	8/85
Cave Climber	9/85
FroSpi	10/85
Robby	11/85
Scrollo	12/85
Froggy	1/86
Rudi the Rat	2/86
Kiki	3/86
Spiel und Spiel-Editor	
Seperate	3/86
Pferderennbahn	4/86
Scrolling	5/86
Die versunkene Stadt	6/86
Mad Monkey	8/86
Herby in Action	10/86
Fantasya	11/86
Blow It	12/86
Das Millionenspiel	1/87
Happy Hacker	3/87
Spekulator	4/87
Metal Ball	6/87
Ping-Pong Classic	7/87
Spider	7/87
Bingo	8/87
Space Business	8/87
Bouncing Ball	10/87
Time Race	11/87
Fighting Hardware	12/87

## Verschiedenes

RESET-Taster	2/85
Eine kleine Bauanleitung	
Buchdatei	7/85
Bücher zum C64	
Analoguhr	10/85
Die große Uhr auf dem C64	
Schlaumeier	6/86
Berichtigung 8/86	
SIMA	8/86
Lerndemo – Maus im Labyrinth	
INPUT-Typensammlung/1	11/86
Der blutige Anfänger	
INPUT-Typensammlung/2	12/86
Der Assembler-Freak	
INPUT-Typensammlung/3	1/87
Der Hacker	
INPUT-Typensammlung/4	2/87
Der Game-Freak	
INPUT-Typensammlung/5	6/87
Der BASIC-Freak	
INPUT-Typensammlung/6	11/87
Der Adventure-Freak	
Samplersette	6/87
Timer (C-Studio)	8/87
VisHypnotica (C-Studio)	12/87

## Lernprogramme

Mathe mit Nico	
Elementares Rechnen	5/85
Flächenberechnung	6/85
Dreisatzrechnen	7/85
Körperberechnung	8/85
Gleichungen	
m. zwei Unbekannten	9/85
Quadratische Gleichungen	10/85
Prozentrechnung	3/86
Zinsrechnung	4/86
Wahrscheinlichkeitsrechnung	5/86
Zahlensysteme Teil 1	6/86
Zahlensysteme Teil 2	7/86
Strahlensatz	8/86
GGT und KGV	9/86

## Physik mit Nico

Beschleunigung	11/85
Impuls	12/85
Energie und Arbeit	1/86
Hebelgesetz	2/86
Elektrizität Teil 1	10/86
Elektrizität Teil 2	11/86
Elektrizität Teil 3	12/86
Elektrizität Teil 4	1/87

## Bits & Bytes im Video-Chip

VIC-Register	1/85
Zeichensatz	2/85
Sprites	3/85
Hochauflösende Grafik	4/85
Rasterzeilen-Interrupt	5/85

## SID-Kurs

Akustik-Grundlagen	6/85
Wellen und Hüllkurven	7/85
Einstimmiges Spiel	8/85
Mehrstimmige Stücke	9/85
Rhythmus-Effekte	10/85
Rhythmus und Bass	11/85
Filter-Steuerung	12/85
Soundcontrol und Effekte	1/86

## 64er Tips

Cursor-Steuerung	1/85
BASIC-Speicherverwaltung	2/85
Logische Verknüpfung	3/85
String-Befehle	4/85
BASIC beschleunigt	5/85
Rechengenauigkeit	6/85
READ und DATA	7/85
Der DEF FN-Befehl	8/85
Der INT-Befehl	9/85
Sortierverfahren	10/85
Betriebssystem-Routinen	11/85
Drucker-Steuerung/I	12/85
Drucker-Steuerung/II	1/86
Die Echtzeituhr	2/86
Sequentielle Dateien	3/86
Kassetten-Tips	4/86
Die USB-Funktion	5/86
Garbage Collection	6/86
Tricks zum LIST-Befehl	7/86
Block-Grafik	8/86
Eingabe-Steuerung	9/86
DOS-Tips Teil 1	10/86
DOS-Tips Teil 2	11/86
DOS-Tips Teil 3	12/86
DOS-Tips Teil 4	1/87
BASIC und Maschinensprache	2/87
System-Routinen	3/87
Mischfarben	4/87
Zeichensatz-Manipulationen	5/87
Sprites	6/87
Der BASIC-Interpreter	7/87
HiRes-Grafik Teil 1	8/87
HiRes-Grafik Teil 2	9/87
HiRes-Grafik Teil 3	10/87
Interrupt-Tips Teil 1	11/87
Interrupt-Tips Teil 2	12/87

<b>Englische Grammatik</b>					
<i>Indef. article/some pronouns</i>	12/86				
<i>Present/past tenses</i>	1/87				
<i>Tail phrases/short answers</i>	2/87				
<i>Prepositions</i>	3/87				
<i>Future tenses/passive voice</i>	4/87				
<i>More pronouns</i>	6/87				
<i>If-clauses/reported speech</i>	7/87				
<i>Nouns/adjectives</i>	8/87				
<i>Modal verbs</i>	9/87				
<i>Meaning of some verbs</i>	10/87				
<b>INPUT-Assemblerschule</b>					
<i>Einführendes</i>	3/87				
<i>Flags, Branches, Zählregister</i>	4/87				
<i>Stack- und Bit-Manipulation</i>	5/87				
<i>Adressierungsarten</i>	6/87				
<i>Interrupts</i>	7/87				
<i>System-Einbindung</i>	8/87				
<b>Werkzeuge</b>					
<b>Masken-Generator</b>	1/85				
Bildschirmmasken erstellen					
<b>Soundcontrol</b>	2/85				
Überarbeitete Version 1/86					
<b>Splitscreen</b>	1/85				
Hires und Text gleichzeitig					
<b>Zeichensatz-Editor</b>	2/85				
Berichtigung 9/85					
<b>MobEd</b>	3/85				
Sprite-Editor					
<b>Autostart-Generator</b>	3/85				
<b>MLM 64</b>	3/85				
siehe MLM64plus (11/87)					
<b>SuperTape D2</b>	4/85				
Schnelles Kassettenformat					
<b>Short-Save</b>	4/85				
Kompaktierte Abspeicherung					
<b>Kassetten-Directory</b>	4/85				
<b>C64-Analyzer</b>	5/85				
BASIC-Programme beobachtet					
<b>RAM-Floppy</b>	5/85				
Schneller Speicher im Rechner					
<b>Recorder Justage</b>	6/85				
Recorder richtig eingestellt					
<b>BASIC-Kompaktor</b>	6/85				
BASIC-Programme kürzen					
<b>TapeCopy</b>	7/85				
Kopieren normal/SuperTape					
<b>Scroll-Editor</b>	7/85				
BASIC-Editor scrollfähig					
<b>DiskMon 1541</b>	8/85				
Einblick in die Diskette					
<b>Screen Display</b>	8/85				
Nachtrag 12/85					
<b>ReAss</b>	9/85				
6502/6510 Re-Assembler					
<b>Relocator</b>	10/85				
Nachtrag 12/85					
<b>DatCopy</b>	11/85				
Kopieren sequentieller Dateien					
<b>DiscDoctor</b>	11/85				
<b>Memory-List</b>	12/85				
Speicherlupe für den C64					
<b>TextMagic</b>	1/86				
Bildschirm-Editor					
<b>MiniGraphic</b>	2/86				
Grafik im Sprite-Bildschirm					
<b>MultiTape I</b>	5/86				
Multi-Format Kassettenlader					
<b>MultiTape II</b>	8/86				
Erweitert um Datei-Handling					
<b>INPUT-Ass</b>	6/86				
6502/6510-Macro-Assembler					
<b>Sprity</b>	8/86				
Muticolor-Sprite-Editor					
<b>Centronics-Treiber</b>	12/86				
Für C64/128 mit Source					
<b>JAM</b>	1/87				
Grafische Benutzeroberfläche					
<b>SuperDisk</b>	1/87				
Floppy-Beschleuniger					
<b>IRAs</b>	4/87				
Interaktiver ReAssembler					
<b>SpeedBackup</b>	5/87				
Disketten-Kopie in 1 Minute					
<b>INPUT-SCE</b>	6/87				
Sprite- und Zeichensatz-Editor					
<b>SC-Toolbox</b>	8/87				
Für Sprites und Zeichensätze					
<b>Speedcompiler</b>	10/87				
Kompatibel zu INPUT-BASIC					
<b>MLM64plus</b>	11/87				
Monitor mit Disassembler und Floppybefehlen					
<b>Syntax-Check</b>	11/87				
Gegen Fehler in BASIC-Programmen					
<b>Hilfsprogramme</b>					
<b>2mal BASIC im Rechner</b>	1/85				
REM-Out, Delete, Merge					
<b>Directory, Renumber</b>	2/85				
Sprite-Tools mit					
<b>PrintAt &amp; Fehlerkanal</b>	3/85				
<b>Variablen-Dump, Find,</b>					
<b>Auto-Number</b>	5/85				
<b>PrintAt/Inkey</b>	6/85				
(siehe auch InLine 11/86)					
<b>Rasterzeilen-Tool</b>	6/85				
<b>BASIC-Trace und -Referenz</b>	7/85				
<b>HexBinDez</b>	8/85				
Umwandlung zwischen Zahlensystemen					
<b>DEEK &amp; DOKE</b>	10/85				
2-Byte-Befehle					
<b>Joy- und Rahmen-Sprite</b>	10/85				
<b>BASIC-Zeilen-Tool</b>	11/85				
BASIC generiert BASIC					
<b>Hard-Copy</b>	12/85				
Bildschirm direkt zum Drucker					
<b>ReList</b>	3/86				
Steuerzeichen im Klartext					
<b>Kalkul-Tool</b>	4/86				
Formeln eingabefrei					
<b>MPS801-Grafik-Hardcopy</b>	4/86				
Hires-Bilder auf den Drucker					
<b>ZS-Hardcopy</b>	6/86				
Veränderte Zeichensätze ausgedruckt					
<b>Software zur c't Uhr</b>	6/86				
<b>MPS801 lernt Deutsch</b>	7/86				
Umlaute ausgedruckt					
<b>Disk-Tools</b>	8/86				
Autostart/ReNew/@-Tool					
<b>JoyTast/InLine/PrintAt</b>	11/86				
Ein-/Ausgabe mit Komfort und Joystickabfrage					
<b>Centronics-Treiber</b>	12/86				
C64/128 mit Source					
<b>Label-Tool</b>	2/87				
<b>BCD-Arithmetik</b>	3/87				
<b>Byte-Kompaktor</b>	5/87				
<b>SuperTape-Loader</b>	8/87				
<b>Trickblende</b>	9/87				
<b>HexBinDez-Tool</b>	10/87				
<b>Multi-Hardcopy</b>	11/87				
Sprites, Zeichensätze und Grafik gleichzeitig					
<b>Graudruck</b>	12/87				
<b>Programmiersprachen</b>					
<b>HiresSpeed</b>	4/85				
Grafikerweiterung					
<b>Sound-Tool</b>	9/85				
Musikprogrammierung in BASIC					
<b>INPUT-BASIC</b>	1/86				
40 zusätzliche Befehle					
<b>LISP 64 Teil 1</b>	4/86				
Interpreter mit Demos					
<b>LISP 64 Teil 2</b>	5/86				
Macros, Editor, Tracer, Arrays und Sets					
<b>LISP 64 Teil 3</b>	6/86				
ELIZA und Demo-Expertensystem					
<b>INPUT-ASS</b>	6/86				
6502/6510 Macro-Assembler					

<b>Turtle-Graphic</b>	8/86
BASIC goes Logo!	
<b>INPUT-Windowing</b>	7/87
Textfenster-Verwaltung	
<b>ICI</b>	7/87
INPUT-Command-Interpreter	
<b>IMC-BASIC</b>	9/87
Multicolor-Grafikerweiterung	
<b>PLH</b>	12/87
BASIC-Programmierhilfen	

## ID-Werkstatt

<b>Flug-Simulator</b>	11/85
<b>Formelsammlung</b>	11/85
<b>Musikmaster</b>	12/85
<b>Spesoquick</b>	12/85
<b>Telefonautomat</b>	3/86
<b>Gitarrensenservice</b>	3/86
<b>Direct-Sound</b>	5/86
<b>Hires-Painter</b>	5/86
<b>Akkord-Synthesizer</b>	7/86
<b>Latein Lernen</b>	7/86
<b>Senso</b>	9/86
<b>Wollmaster</b>	9/86
<b>Zahlen-Memory</b>	11/86
<b>Plot (Punkte abfragen)</b>	11/86
<b>Foto-Timer, Tapezierdatei,</b>	
<b>Datei für INPUT-Calc</b>	2/87
<b>INPUT-CAD-Noten, CALC-Patch,</b>	
<b>Vokabel-Wandler 1</b>	4/7
<b>INPUT-CAD-Patch, C64 am Sharp,</b>	
<b>VT-Patch, MiniDat-Kass.V.</b>	5/87
<b>Kopfrechentainer, Zahlenrätsel,</b>	
<b>Dictionary-Dateien &gt; VT-Dateien</b>	7/87
<b>Patch für INPUT-Calc64/128</b>	
<b>Renumber, Kompaktor, Dekompaktor</b>	10/87
<b>Vokabeltrainer-Zeichensätze</b>	11/87
Mit französischen Umlauten	

## Rätsel

<b>Strings im Hires-Bild</b>	1/85
Lösung 3/85	
<b>Codierung</b>	3/85
Lösung 5/85	
<b>Zahlen im Viereck</b>	5/85
Lösung 7/85	
<b>Sortierung</b>	7/85
Lösung 9/85	
<b>Send More Money</b>	9/85
Lösung 11/85	
<b>Primzahlen</b>	11/85
Lösung 1/86	
<b>Magische Zehner</b>	1/86
Lösung 4/86	
<b>Permutation</b>	3/86
Lösung 6/86	

<b>Schönheitswettbewerb</b>	7/86
Lösung 10/86	
<b>Das Magische Quadrat</b>	9/86
Lösung 12/86	
<b>Das n-Damen-Problem</b>	1/87
Lösung 4/87	
<b>Einer gegen alle</b>	3/87
Lösung 6/87	
<b>Schnelles Zirkeln</b>	7/87
Lösung 10/87	
<b>Einer gegen alle</b>	9/87
Lösung 12/87	

## Testberichte, News

<b>Turboass (Assembler-Test)</b>	1/86
<b>16k-Cmos-Speicher</b>	1/86
<b>Buchtip: Durchblick</b>	1/86
<b>Floppy-Beschleuniger (Test)</b>	3/86
<b>LISP-Bücher im Vergleich</b>	4/86
<b>Kopierprogramm Nibbler</b>	4/86
<b>Star-NLQ-Drucker</b>	5/86
<b>C64c und GEOS (Test)</b>	7/86
<b>Selbstbauplotter</b>	
<b>Fisher-Technik</b>	9/86
<b>Ergebnisse Leserumfrage '86</b>	1/87
<b>3 1/2-Zoll Floppy</b>	6/87
<b>Dolphindos</b>	6/87
<b>Funkuhrmodul ACC64</b>	7/87
<b>Drucker-Interface 92128 GTI</b>	8/87
<b>Drucker-Interface</b>	
<b>24-Nadel-Drucker</b>	10/87
<b>BTX-Decoder-Modul 2</b>	
<b>Vorstellung</b>	11/87
<b>Test</b>	12/87

## Know-how

<b>BASIC-Speicherverwaltung</b>	11/86
<b>Spiel-Algorithmen</b>	11/86
<b>128er CP/M-Startup</b>	12/86
<b>Assembler-Know-how/1</b>	3/87
Symbole und Berechnungen	
<b>Assembler-Know-how/2</b>	5/87
Struktur und Modularität	
<b>64er festgenagelt</b>	6/87
Tips für 128er-Besitzer	
<b>ICI unter Assembler</b>	9/87
<b>Assembler-Know-how/3</b>	9/87
Makros und bedingte Assemblierung	
<b>Tips zum Speedcompiler/1</b>	11/87
Differenzierte Fehlermeldungen,	
RS232	
<b>Tips zum Speedcompiler/2</b>	12/87
Programm-Optimierung	
<b>BTX-Einführung</b>	12/87

# Alles automatisch

## Jahresinhaltsdatei-Generator

Ein Elektronisches Magazin wie INPUT 64 kann sich natürlich mit einem gedruckten Jahresinhaltsverzeichnis nicht zufrieden geben. Deswegen gibt's das Ganze noch 'mal als Datei für UniDat auf Diskette.

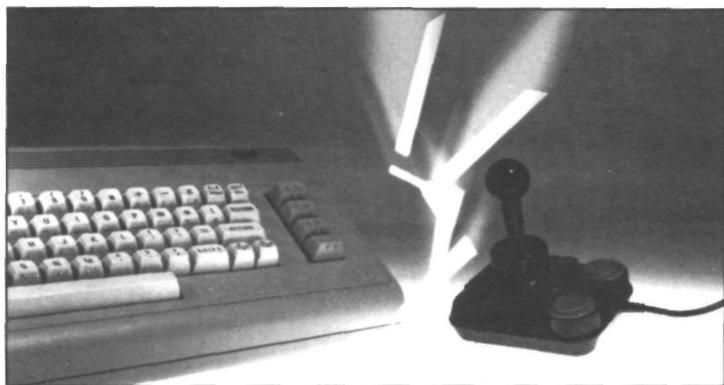
Ganz so einfach, wie es klingt, ist dieser Datenservice aber doch wieder nicht. Allein aus Platzgründen können wir auf der INPUT-Diskette keine relative Datei<sup>1</sup> mitliefern, unser Dateiverwaltungs-Programm UniDat mag aber nur solche. Deswegen befindet sich unter dem Namen „Jahresinhalt 85-87“ als zweites Modul auf der INPUT-Diskette<sup>2</sup> ein Generierungs-Programm, das die notwendigen Dateien erzeugt.

Sie müssen nichts weiter tun, als eine Diskette bereitzuhalten. Diese wird nach dem Start des Programms innerhalb von INPUT 64 zunächst formatiert – es sollten sich also auf keinen Fall Dateien auf dieser Diskette befinden, die Sie noch benötigen. Anschließend werden die relative Datei mit den Daten selbst, das File, das die Maske enthält, und eine Indexdatei für das erste Datenfeld (Programmname) erstellt, unter den Namen „z-input64/85-87“, „y-input64/85-87“ und „a-input64/85-87“.

War dies erfolgreich, können Sie diese Datei zusammen mit UniDat benutzen. Der Dateiname, der von UniDat abgefragt wird, ist „input64/85-87“. Wenn Sie auch über andere Datenfelder als die Programmnamen schnell auf die Datei zugreifen wollen, lassen Sie bitte weitere Indexfelder erzeugen. JS

<sup>1</sup>In den 64er Tips der Ausgabe 12/86 ist Aufbau und Programmierung relativer Dateien ausführlich erklärt.

<sup>2</sup>Ausgabe 12/87, in der auch die Dateiverwaltung UniDat veröffentlicht wurde.



# Computer-Fans Traum

## Spiel: Fighting Hardware

So eine, auch jedem Hacker geläufige, fixe Idee ist der Anstoß gewesen, dieses Spiel zu kreieren. Zwei allen bekannte Hardware-Teile tanzen miteinander und kämpfen gegen Begrenzungen und Hindernisse. Akteure sind ein Herr Joystick und ein Fräulein Mouse. Sie tanzen beide, jede(r) schön auf Abstand auf eigenem Parkett, auf einem rechteckigen Spielfeld, das in der Mitte waagrecht geteilt ist: obere Hälfte als Tanzfläche für Mausi, die untere für Joyi. Wehe, einer der beiden berührt diese Trennungslinie: Sofort ist der Tanz für dieses Spiel beendet. Aber die Ränder links, rechts, oben müssen die beiden genauso meiden. Jegliche Linienberührung bedeutet unter heftigem Zucken und Schütteln das Aus.

### Nicht nur Vater Zeus . . .

Unser Spiel ist ausgelegt für genau zwei Spieler. Der eine steuert Mausi, der andere Joyi, wahlweise jeder über einen Joystick oder über die Tastatur. Nach Erscheinen des Titelbildes entweder 'J', dann steuert der Joystick in Port 1 die Maus, der in Port 2 den Bildschirm-Joystick Joyi, oder 'T' drücken und damit die Tastatursteuerung wählen (siehe Bild).

**Welcher Freund der Computer-Spiele hat nicht irgendwann schon mal davon geträumt—nachts, so zur Geisterstunde, würde seine Hardware, etwa die Diskette mit der Datassette, die Floppy mit dem Handbuch, der Koppler (FTZ) mit dem Modem (nur für Export!) oder auch der Joystick mit der Mouse ein Tänzchen wagen. Hat sich nicht jeder schon mal gefragt, warum wohl die Diskettenbox morgens verstreut am Boden lag, obwohl man doch sicher war, sie am Abend noch mitten auf dem Tisch abgestellt zu haben?**

Funktion	Spieler 1 (Mausi) Taste	Spieler 2 (Joyi) Taste
Feuer	E	@
Hoch	W	P
Links	A	L
Rechts	S	:
Runter	Z	.

Für jeden Spieler die richtige Taste

Kampfgeleüste kommen dadurch auf, daß man mittels Feuerknopf dem Tanzpartner/ Gegner Lichtblitze in das Feld schleudert, die dann als Hindernisse das freie Bewegungsfeld im gegnerischen Teilrechtlich merklich eingrenzen. Die Positionierung der Lichtblitze beim Gegner hängt von der eigenen Stellung im Augenblick des Schleuderns ab. So befindet sich mit etwas Geschick/Training der Gegner bald arg in der Klemme. Wird eines dieser Hindernisse berührt oder wie zuvor gegen eine der vier Umgrenzungslinien gestoßen, ist das Spiel aus.

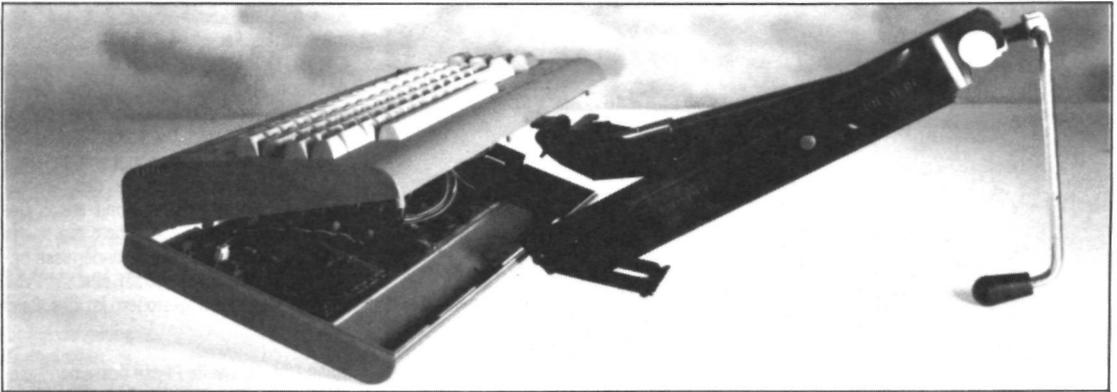
Dann tritt die zweite Phase desselben Spieles ein. Im Zentrum des Bildschirms erscheint ein Sichtfenster mit der Zahl Fünf (soviel Punkte hat der Sieger des Spiels bisher erreicht). Jetzt kommt es darauf an, daß der Sieger ganz rasch auf den Feuerknopf drückt, und schon erhöht sich die Fünf auf Zehn, wiederum rasch gedrückt, von Zehn auf Fünfzehn und so weiter — jeweils um 5 Punkte. Das würde so weitergehen, wenn nicht von Mal zu Mal schneller eine Klappe mit der Aufschrift „Stop“ fiel. Drückt man den Feuerknopf, wenn „Stop“ anstelle der Punktezahl erscheint, dann ist auch Phase 2 beendet, und der Sieger (Joyi beziehungsweise Mausi) erhält die Punktezahl, die zuvor zuletzt korrekt gedrückt wurde, bevor das Malheur mit dem Stoppschild passierte.

### . . . konnte blitzen

So kann der Sieger durchaus 45 oder gar 50 Punkte pro Spiel einheimsen. Dann beginnt das nächste Spiel der Runde, wobei eine Runde immer aus fünf Spielen besteht. Die jeweils erreichten Punkte pro Spiel werden innerhalb einer Runde aufaddiert. Am Ende des fünften Spieles ist dann ablesbar, wer besser tanzte — Mausi oder Joyi. Rundensieger ist der, der die meisten Punkte erreicht hat.

Auch ein Unentschieden ist denkbar: Gelingt es nämlich beiden Spielern, in einem Spiel über längere Zeit (etwa 60 sec lang) allen Rändern und Hindernissen geschickt auszuweichen, so bekommen beide zunächst ihre fünf Siegerpunkte gutgeschrieben.

W. Schmidt-Pabst/Kfp



# Programmer's Little Helper

## PLH: BASIC-Entwicklungssystem

In BASIC, oft als Einsteigersprache verschrien, läßt sich mit einigen Einschränkungen durchaus ernsthafte Software programmieren. Dies gilt erst recht, wenn BASIC in Verbindung mit einem Compiler benutzt wird. In der letzten Oktoberausgabe hat INPUT64 den SPEEDCOMPILER veröffentlicht, der mit vielen Optionen ausgestattet ist und den Geschwindigkeits-Killer BASIC zu einem echten Renner werden läßt.

Um jedoch einen Compiler einsetzen zu können, ist oft eine Anpassung des Programmtextes an bestimmte Konventionen erforderlich! Um diese Anpassung zu erleichtern, aber auch um eine völlige Neuentwicklung von BASIC-Programmen zu unterstützen, können Sie PLH als hilfreiche Unterstützung benutzen. Mit Funktionen wie Scrolling nach oben und unten sowie den Befehlen FIND und CHANGE bekommt der Editor beinahe die Qualität einer Textverarbeitung.

### Wiederfinden

In den letzten Jahren hat INPUT 64 zahlreiche Tools veröffentlicht, die die Entwicklung von BASIC-Programmen erheblich er-

**Mit PLH erhalten Sie ein mächtiges Werkzeug, das Ihnen bei der Programmentwicklung, Fehlersuche oder der Compiler-Anpassung eine große Hilfe sein wird. Mühsames Durchblättern Ihres BASIC-Textes mit LIST entfällt, sollten Sie auf der Suche nach bestimmten Ausdrücken sein, oder im gesamten Programm eine Formulierung in eine andere ändern wollen, hier hilft FIND and CHANGE. Außerdem unterstützt PLH modulare Programmierung durch MERGE und teilweise Speichern mit dem erweiterten SAVE-Befehl.**

leichtert. Sind diese für sich meist sehr leistungsfähig, so ist es doch oft ein Problem, verschiedene Tools gleichzeitig im Speicher miteinander laufen zu lassen. In PLH sind nun die wichtigsten Funktionen in einer Neuauflage enthalten. Außerdem sind noch einige Funktionen hinzugekommen.

Das Programm belegt den Bereich von \$A000 bis \$CFFF (40960 bis 53247). Der BASIC-Bereich bleibt somit unberührt und steht für eigene Entwicklungen zur Verfügung. Tools wie PRINT AT oder INKEY, die

am BASIC-Anfang liegen, können Sie weiterhin ohne Einschränkung benutzen, solange Sie den PLH-Bereich nicht zur Zwischenspeicherung verwenden. Dies dürfte aber bei den meisten Tools keine Probleme bereiten.

Das Programm stellt Erweiterungen aus zwei Bereichen zur Verfügung: Zum einen wurde der Editor um zahlreiche Funktionen erweitert. Außerdem existieren eine Reihe von Befehlen, die Ihnen die Entwicklung von BASIC-Programmen erheblich erleichtern. Diese Befehle sind nicht dazu gedacht, innerhalb von Programmen eingesetzt zu werden, wie das zum Beispiel bei einer Grafikbefehlsweiterung der Fall ist. Dies wäre bei den nachfolgenden Befehlen auch wenig sinnvoll. Eine Funktion wie RENUMBER wird bei einem fertigen Programm sowieso nicht mehr benötigt, dafür bei der Programmerstellung um so häufiger. Zu Beginn sollten Sie sich mit dem erweiterten Editor vertraut machen.

### Kreuz und quer

Es ist jetzt möglich, das Listing auf dem Bildschirm mit den CRSR-Tasten nach oben

und nach unten zu scrollen. Dazu müssen Sie lediglich mit dem Cursor an den jeweiligen Bildschirmrand fahren. Der Cursor wird dabei automatisch auf den Beginn der eigentlichen BASIC-Zeile positioniert.

Zusätzlich können Sie in Verbindung mit der CTRL-Taste zahlreiche Funktionen aufrufen. Sie finden im Kasten „Unter Kontrolle“ die einzelnen Kombinationen und deren Belegung. Einige der Funktionen, die Sie hiermit erreichen können, bedürfen einer zusätzlichen Erläuterung.

Bei Zeilen löschen (CTRL-f5, -DEL, -CRSR ↓) ist der Cursor ausschlaggebend. Löschen nach rechts beseitigt die Zeichen ab Cursor-Position, Löschen nach links die Zeichen zwischen Cursor-Position und Zeilennummer, das bedeutet: die Zeilennummer bleibt erhalten. Benutzen Sie CTRL-CRSR ↓, werden die Zeichen links vom Cursor gelöscht und die Zeichenfolge rechts von Cursor nach rechts an das erste Leerzeichen herangerückt.

HEXDEZBIN (CTRL-H) ermöglicht die komfortable Umrechnung von Zahlen in hexadezimaler, dezimaler und binärer Schreibweise in die jeweils anderen Zahlensysteme. Es kann mit der SPACE-Taste oder RUN/STOP wieder abgebrochen werden.

Der Notizbildschirm (CTRL-N): Mit einem Tastendruck können Sie den Inhalt des aktuellen Bildschirms samt Farbinformationen mit dem Inhalt eines zweiten Bildschirm-puffers austauschen. Erscheint bei der ersten Benutzung Zweifelhafes, reicht ein kurzes CLR/HOME zum Löschen. Nach einem Reset bleibt der Inhalt des zweiten Bildschirm-puffers erhalten.

Mit RUN/STOP können Sie sich aus dem vielgehaßten Hochkomma-Modus befreien. Ein Druck, und der Cursor kann wieder frei bewegt werden.

## Kommandosprache

AUTO n schaltet die automatische Zeilennummerierung mit der Schrittweite „n“ ein. „n=0“ schaltet den AUTO-Befehl wieder ab. Achten Sie darauf, daß Sie durch Betätigen der RETURN-Taste nicht versehentlich Zeilen in einem bereits bestehenden Programm löschen. Sollten Sie innerhalb eines Programmlistings den AUTO-Modus einge-

schaltet haben, verlassen Sie die letzte Zeile mit SHIFT-RETURN. Dadurch wird diese Zeile nicht mit übernommen.

RENUMBER n,m numeriert ein Programm neu durch in der Schrittweite „m“ beginnend bei „n“.

DELETE n,m löscht die Zeilen von 'n' bis 'm'. Soll nur eine Zeile gelöscht werden, reicht auch der Parameter 'n'.

CUTREM entfernt aus einem Listing alle eventuell enthaltenen REM-Befehle mit den zugehörigen Kommentaren. Wenn die Zeile nur aus einem REM-Befehl besteht, bleibt die Zeile durch einen eingefügten Doppelpunkt erhalten. Es ist ja immerhin möglich, wenn auch unschön, daß ein Sprungbefehl im Programm ausgerechnet auf eine solche REM-Zeile verzweigt.

## Suchen und Finden

CHANGE exp1;exp2 sucht zuerst nach dem Ausdruck „exp1“ (beispielsweise „i,j“) und

### CTRL-Taste und . . .

f1	= Erste Programmzeile
f3	= Letzte Programmzeile
f5	= Zeile links löschen
f7	= Leerzeile einfügen
DEL	= Zeile rechts löschen
RETURN	= Cursor rechter Rand
CRSR ⇒	= Cursor links unten
CRSR ↓	= Zeilenrest nach links
H	= HEXDEZBIN
N	= Notiz-Bildschirm

### Unter Kontrolle: Tastenbelegung der CTRL-Funktionen

ersetzt diesen durch „exp2“ (zum Beispiel „k,j“). Dies geschieht aber erst nach vorheriger Sicherheitsabfrage (JA/NEIN/STOP). Diese gibt die aktuelle Zeile aus und zeigt an, um das wievielte Zeichen im String es bei der Abfrage geht. Eine '2' bedeutet also, daß CHANGE das zweite Zeichen austauschen möchte. Hierbei ist es sehr wichtig zu wissen, daß BASIC-Befehle intern mit nur einem Byte (Token) abgespeichert werden, jeder Befehl zählt also nur als ein Zeichen. Somit folgt nach einem Befehl PRINT

erst das zweite Zeichen und nicht das sechste. Es nicht möglich und auch nicht sinnvoll, Ausdrücke verschiedener Art auszutauschen. Es geht also nur:

TOKEN gegen TOKEN  
Beispiel:END gegen STOP

ASCII gegen ASCII  
Beispiel:„HALLO“ gegen „HEY!“

ASCII-Strings werden durch Hochkomma gekennzeichnet. Da der CHANGE-Befehl intern mit der CHRGET-Routine arbeitet, die Leerzeichen und Doppelpunkte überliest, kann kein Text eingefügt werden, der solche Zeichen enthält.

FIND exp findet Token (Schlüsselcode der BASIC-Befehle) oder ASCII-Strings im BASIC-Text. Die Ausgabe kann mit SPACE festgehalten und mit RUN/STOP abgebrochen werden. Das Zeichen 'E' findet als Joker Verwendung. Beispiel: FIND „WerE“ findet ASCII-Strings wie „Wer?“, „Werte“, „Wertpapiere“ oder ähnliche. Das Pfund-Zeichen selbst kann natürlich nicht mehr gesucht werden, was aber wohl zugunsten des Komforts zu verschmerzen ist.

## Innen und Außen

Mit MEM bekommen Sie die augenblickliche Speicher-Konfiguration angezeigt. Es werden dabei berücksichtigt: BASIC-Speicherbereich, Programmlänge, Variablen, Felder, Strings sowie der verbleibende freie Speicher.

DIR gibt das Directory aus, ohne daß das aktuelle BASIC-Programm zerstört wird. Dank des neuen Formats der Directory-Darstellung brauchen Sie, um ein Programm zu laden, den Cursor nur auf die entsprechende Zeile zu positioniert und RETURN drücken. Die Ausgabe läßt sich mit SHIFT festhalten.

LLIST gibt ein Listing auf den angeschlossenen Drucker aus. Sie ersparen sich das leidige OPEN4,4:CMD4:LIST:CLOSE4

LOAD: Der „Load“-Befehl wurde erweitert. Die Bedeutung der Sekundäradresse ist erweitert:

keine = normal<sup>2</sup>

0 = an den BASIC-Anfang

1 = absolut

2 = an die angegebene Adresse

## Unterstützung übersichtlich

<b>AUTO a</b>	Automatische Zeilennr.	<b>MEM</b>	Bestandsaufnahme
<b>DELETE n,m</b>	Löschen von Zeilen	<b>NORM</b>	Zum BASIC-Start
<b>RENUMBER n,a</b>	Neue Zeilennummern	<b>OFF</b>	PLH abschalten
<b>LLIST [n-m]</b>	Listing zum Drucker	<b>OLD</b>	Programm zurückholen nach NEW
<b>CUTREM</b>	REM - raus damit		
<b>CHANGE</b>			
<b>exp1;exp2</b>	Ändern in		
<b>FIND exp1</b>	Suchen nach	<b>DIR</b>	Directory
<b>TRON</b>	Jeden Schritt sehen	<b>@ "DOS-Befehl"</b>	DOS direkt
<b>TROFF</b>	Wieder ausschalten	<b>@</b>	Status
<b>TRACESPEED v</b>	Schnell oder langsam	<b>@ ga</b>	Anderes Gerät

<b>MERGE "name".ga</b>	Verketten von Programmteilen
<b>SAVE "name".ga,lsa,val1,val2]</b>	Speichern mit Besonderheiten
<b>LOAD "name".ga,lsa,val1]</b>	LOAD-Befehl erweitert

### Legende:

n,m	Zeilennummern	exp	Ausdruck (ASCII oder Token)
a	Schrittweite		
v	Zahlenwert	val	Zahlenwert (Zeile oder Adresse)
ga	Geräteadresse	[ ]	Angaben sind optional
sa	Sekundäradresse		

### Kleine Hilfen im Überblick: sämtliche Befehle zu PLH

Beispiele:

LOAD"OBJ",8,2,9\*4096

lädt das File "OBJ" an die Adresse \$9000

LOAD"PROG",8,0

lädt "PROG" an den BASIC-Anfang, der durch die Adressen 43/44 bestimmt wird.

MERGE erlaubt es, Programmteile zu löschen. Die Syntax entspricht dem normalen LOAD-Befehl. Ein BASIC-Programm wird zu dem bestehenden nachgeladen, Zeilennummern werden eingefügt und eventuell überschrieben.

Der SAVE-Befehl wurde modifiziert. Abhängig von der Sekundäradresse ist es möglich, Programmteile, Speicherbereiche und komplette Programme zu speichern.

keine = normal<sup>2</sup>  
0 = BASIC teilweise  
1 = BSAVE

Beispiele:

SAVE"PROG",8,0,1000,1100

speichert aus dem BASIC-Programm die Zeilen 1000 bis 1100 ab.

SAVE"OBJ",8,1,9\*4096,9\*4096+256 speichert den Bereich von \$9000 bis \$9100 auf Diskette ab.

SAVE"TEST",8 speichert wie sonst auch.

NORM setzt den BASIC-Start auf \$0801. Damit ersparen Sie sich das lästige „POKE 43,1:POKE 44,8“, wenn Sie zum Beispiel mit PRINT AT und INKEY arbeiten.

OFF schaltet die Erweiterung ab. Ein Neustart ist mit ?USR(X) oder SYS 49152 möglich.

OLD regeneriert ein mit NEW oder RESET gelöscht BASIC-Programm.

## Zugeschaut

TRON schaltet den sogenannten „Trace-Modus“ ein. Am oberen Bildschirmrand wird jeweils die aktuell bearbeitete Zeile

ausgegeben. Das Programm kann mit der **C**-Taste angehalten werden.

TROFF schaltet den Trace-Modus wieder ab.

TRACESPEED n: Der Wert von 'n' (0-255) bestimmt, wie schnell das Programm im Trace-Modus abgearbeitet wird. Dieser Befehl kann ausnahmsweise auch sinnvoll in einem Programm während der Entwicklungsphase eingesetzt werden, um bestimmte Programmteile schnell, andere aber besonders langsam ablaufen zu lassen.

@ "..." der Befehls-String zwischen den Anführungsstrichen wird zur Floppy geschickt.

@ liest den Fehlerkanal aus und gibt eine Kurzmeldung auf den Bildschirm.

@n wählt aktuelle Geräteadresse n für das Diskettenlaufwerk. Auf diesen Wert beziehen sich DIR, @ und @ "..."

## Ein Blick zurück . . .

INPUT 64 Insider werden in PLH sicherlich viele Funktionen wiedererkennen, die schon irgendwann einmal veröffentlicht worden sind. In diesem Sinne haben eine ganze Reihe von Autoren an diesem Projekt mitgewirkt, die hier noch einmal kurz genannt werden sollen.

Wolfgang Cassing (REMOUT/DELETE)  
Werner Jentzen (FIND und CHANGE)  
Sven Hesselbach (INPUT-BASIC<sup>3</sup>)  
Hajo Schulz (HEXDEZBIN)  
Peter Seeliger (SCROLLHILFE)  
Frank Börncke (Anpassung, Erweiterung und Einbindung in PLH)

Frank Börncke/rh

<sup>1</sup> In INPUT 64, Ausgabe 10/87, Seite 8, finden Sie die Hinweise zur Anpassung an den SPEEDCOM-PILER aus INPUT 64.

<sup>2</sup> Meint Laden eines Programms nach Art des Betriebssystems. V2- und V3-Kernal von Commodore, das Standard-Betriebssystem laden grundsätzlich an den BASIC-Anfang (entspricht 8,0), Speed-DOS lädt absolut, an die Ladeadresse (wie 8,1).

<sup>3</sup> Die in INPUT 64, Ausgabe 1/86 veröffentlichte BASIC-Spracherweiterung, in der ein Teil der Befehle bereits enthalten ist.

# Schnelles schneller machen

## Teil2: Optimiertes Programmieren

Bei der Programmierung des in INPUT64 10/87 veröffentlichten Speedcompilers spielten die Konzeptionen der Programmiersprachen C und Pascal eine Rolle. Das spiegelt sich vor allem in der Behandlung der Datentypen, sprich der Variablen, durch den Compiler wieder. So kennt der Compiler beispielsweise den Datentyp „Array of Char“, der in diesen beiden Sprachen benutzt wird; ein Feld mit einzelnen Zeichen.

Und aus gutem Grund legt der Compiler großen Wert auf die Integer-Verarbeitung. Wenn man nämlich überall dort, wo dies möglich ist, Integer-Variablen und -Ausdrücke statt der Gleitkomma-Darstellung benutzt, kann man eine erhebliche Geschwindigkeitssteigerung erreichen. Verbunden mit dem Geschwindigkeitsvorteil, den der Compiler von Haus aus bringt, ergibt das sehr schnelle Programme.

### Speed ohne Kommata

Integer-Zahlen sind ganze Zahlen im Bereich von -32768 bis +32767. Diese Zahlen werden intern durch 2 Bytes dargestellt und verbrauchen deshalb nur sehr wenig Speicherplatz. Vor allem aber lohnt sich die Benutzung von Integer-Variablen mit dem Speedcompiler, weil dieser reine Integer-Berechnungen in besonders schnellen Code umsetzt. Denn der Compiler besitzt eine besonders schnelle Integer-Multiplikations-Routine, und auch die anderen Integer-Arithmetik-Routinen sind um ein Vielfaches schneller als ihre Gegenstücke in den BASIC-Gleitkomma-Routinen.

Bei sehr vielen Programmen ist es sogar möglich, ausschließlich mit Integer-Variablen zu arbeiten. Der Compiler selbst, der ja auch in BASIC geschrieben und dann compiliert wurde, benutzt fast nur Integer-Va-

**Um wirklich das Letzte aus dem Speedcompiler herauszuholen, bedarf es ein wenig Beschäftigung mit dessen zahlreichen Compiler-Optionen. In dieser Folge unserer dreiteiligen Serie geht es vor allem um Zeitfragen, nachdem beim letzten Mal die möglichen Fehlermeldungen abgehandelt wurden.**

riablen. Dadurch ist er trotz komplexer Algorithmen und Strukturen (Stichwort: rekursive Programmierung) sehr schnell.

In gewissen Grenzen lassen sich auch Zahlen, die eigentlich keine Integers sind, ganzzahlig darstellen:

```
5 data 10.80, 9.95, 1.00, .99, 195.00
10 endpreis = 0 : for i=1 to 5
20 read preis
30 endpreis = endpreis + preis
40 next i
50 endpreis = endpreis * 1.14:
rem Mehrwertsteuer
60 print "Sie zahlen"; endpreis; "DM"
```

Die Preise sind zunächst keine Integers, weil sie Nachkommastellen besitzen. Multipliziert man sie aber mit 100, erhält man ganze Zahlen, die kleiner als 32000 sind. Folgendes umgeformte Programm liefert also das gleiche Ergebnis wie oben:

```
5 data 1080, 995, 100, 99, 19500
10 ep% = 0 : for i%=1 to 5
```

```
20 read preis%
30 ep% = ep% + preis%
40 next i%
50 endpreis = ep% * .0114:
rem Mehrwertsteuer
60 print "Sie zahlen"; endpreis; "DM"
```

Die Preise wurden in Pfennigen berechnet. Das geht natürlich nur, solange der Wert von 327,67 DM nicht überschritten wird. Die Multiplikation mit 1.14 muß in Gleitkomma-Arithmetik ausgeführt werden, da hier wieder Nachkommastellen auftreten können; dadurch hat auch die Zuweisung auf eine Gleitkomma-Variablen zu erfolgen. Die Multiplikation mit 0.0114 rückt das Komma wieder an die richtige Stelle.

Bei beiden Programmen können bei der Berechnung der Mehrwertsteuer mehr als zwei Nachkommastellen auftreten. Abhilfe schafft Rundung:

```
55 endpreis = int(endpreis*100+.5)/100
```

### Schnelle Schleifen

Besonders interessant in diesem Zusammenhang sind, wie schon oben im Programm zu sehen, Integer-Schleifen. Nimmt man nämlich als Schleifen-Variablen eine Integer-Variablen (was beim Interpreter nicht möglich ist), wird eine Integer-Schleife vom Compiler generiert, die – abgesehen vom Schleifenrumpf – zehnfach mehr Geschwindigkeit bringt. So wäre die Verwendung von 'i%' statt 'i' als Schleifen-Variablen im obigen Beispiel auch ohne die Umwandlung der anderen Gleitkomma-Variablen sinnvoll gewesen.

Allerdings sind in der Integer-Schleife nur zwei Step-Werte zugelassen: Das ist zum

einen '+1', bei dem keine Step-Angabe erfolgen darf, und zum anderen „Step -1“. Im letzteren Fall muß '-1' eine Konstante sein. Selbstverständlich müssen auch Integer-Schleifen den Integer-Zahlenbereich von -32768 bis 32767 einhalten.

## Endlich kommafrei!

Arbeitet man fast nur mit Integer-Variablen, so kann man ab der Compiler-Option 'EVO' das Prozentzeichen nach dem Variablen-Namen weglassen. Diese Option bewirkt, daß alle Variablen ohne zusätzliche Angaben als Integers behandelt werden. Will man trotzdem einige Gleitkomma-Variablen benutzen, so müssen diese dann mit einem angehängten '&' gekennzeichnet werden.

Das '&' im Variablen-Namen führt unter dem normalen BASIC-Interpreter zum „Syntax error“. Will man Integer-Schleifen benutzen und trotzdem das Programm so schreiben, daß es noch mit dem Interpreter läuft, kann man diese Option vor dem FOR-Befehl einschalten (VO) und dann direkt dahinter wieder ausschalten (V1). Dadurch interpretiert der BASIC-Interpreter die Variable als Gleitkomma und der Compiler als Integer. Nur wenn man die Schleifen-Variablen auch beim NEXT-Befehl angeben will, muß man diese Option vor dem NEXT ein- und danach wieder ausschalten. Das NEXT muß dann natürlich in einer eigenen Zeile stehen, genauso wie der FOR-Befehl.

Da bei einer Integer-Division die Nachkommastellen nicht ermittelt werden, gibt es die Möglichkeit, beim Speedcompiler den Divisions-Rest zu erfahren. Er ist nämlich direkt nach jeder Integer-Division in der internen Status-Variablen QA zu finden:

```
10 rem Modula Bildung
20 a%=7 : b%=20
30 c%= b%/a%
40 print qa
```

Das Programm druckt die Zahl '6', den Divisions-Rest, aus. Zwischen der Division und der Abfrage von QA dürfen keine weiteren Befehle mehr stehen. Im übrigen wird nur dann eine Integer-Division durchgeführt, wenn alle Variablen und Konstanten in der Formel Integers sind und wenn die Variable, der das Ergebnis zugewiesen wird, vom Typ Integer ist. Die Status-Variablen QA ist grundsätzlich eine Integer-Variablen, die nur

gelesen werden und der nichts zugewiesen werden darf (wie ST).

## Alles ganz positiv

„Positive Integers“ sind sozusagen ein Überbleibsel der allerersten Compiler-Version, in der es noch keine Gleitkomma-Zahlen gab. Um aber trotzdem Zahlen größer als 32767 darstellen zu können, habe ich mir einen Trick ausgedacht. Mit der Compiler-Option '+' kann man nämlich eine interne Darstellung erzwingen, die Zahlen von 0 bis 65535 erlaubt. Sie werden dabei genauso wie normale Integers abgespeichert, mit dem Unterschied, daß das Vorzeichen zur Darstellung der Mantisse mit einbezogen wird. Dieser Trick ist nicht neu; bei C-Compilern wird er bei den sogenannten „Unsigned integers“ ebenfalls benutzt.

Die Fußangel bei dieser Option ist, daß sie, einmal aktiviert, vom Compiler auf die Werte aller (0) Integers angewendet wird. Hatte 'a%' vorher den Wert '-1', hat dieselbe Variable nachher den Wert '+65535'. Des Weiteren ist es nicht für alle BASIC-Funktionen wichtig, ob eine Integer-Variablen vorzeichenlos ist oder nicht. POKE zum Beispiel kann auch mit 'POKE -1,0' etwas anfangen. Die '-1' wird als die positive Zahl '65535' aufgefaßt.

Zur Übersicht hier die wichtigsten Funktionen, bei denen positive Integers eine andere Behandlung als vorzeichenbehaftete Integers erfahren:

```
Division
Vergleich
Wandlung nach String (mit STR$)
Ausgabe einer Zahl durch PRINT
FOR-NEXT-Schleifen
```

Sinnvoll sind diese positiven Integers etwa in Programmen, die nur positive ganze Zahlen verwenden. Auch bei Integer-Schleifen werden positive Integers benötigt, wenn der normale Integer-Bereich nicht ausreicht. Man kann nämlich durch gezieltes Setzen und Rücksetzen dieser Option nur einzelne Befehle von den positiven Integers abhängig machen. Hier ein Beispiel:

```
10 a%--1987
20 for i%=0 to 65000
30 print a%
40 rem £+
```

```
50 next i%
60 rem £-
```

Hier wird tatsächlich 65001mal '-1987' ausgegeben. Die Compiler-Option '+' wirkt nämlich nur auf das „next i%“, wodurch alle anderen Programmteile unbeschadet bleiben. Eine Angabe der Compiler-Option schon im Kopf der Schleife ist in solch einem Fall bei einer FOR-NEXT-Schleife nicht nötig.

## Ins Positive geglitten

Positive Integers sind in DATA-Zeilen nur dann zulässig, wenn sie zunächst mittels Gleitkomma-Variablen ausgelesen werden. Außerdem ist die Zuweisung von positiven Integers an Gleitkomma-Variablen nicht so einfach möglich, da sie dann trotzdem als vorzeichenbehaftete Zahlen angesehen werden. Die Zuweisung funktioniert aber

```
100 a = a%: if a% < 0 then a = a+65536
```

Noch ein grundsätzlicher Hinweis: positive Integers sollten nur von Programmierern benutzt werden, die die Folgen ihres Tuns genau überschauen, da sich leicht unerwünschte Nebeneffekte einstellen. Hierzu ein Beispiel:

```
10 rem totales Chaos
20 rem £+
30 for i% = 999 to 0 step -1
40 poke i+1024, 32
50 next i%
```

Dieses Programm soll den Bildschirm von unten nach oben löschen. Das tut es auch, POKEd dann aber munter weiter und stürzt ab. Denn: bei positiven Integers gibt es keine kleinere Zahl als Null, und die Schleife wird zu einer Endlosschleife, die nach '0' mit '65535' fortgesetzt wird. Deshalb sollte man bei positiven Integers niemals Null als Endwert einer Schleife mit negativer Schrittweite benutzen.

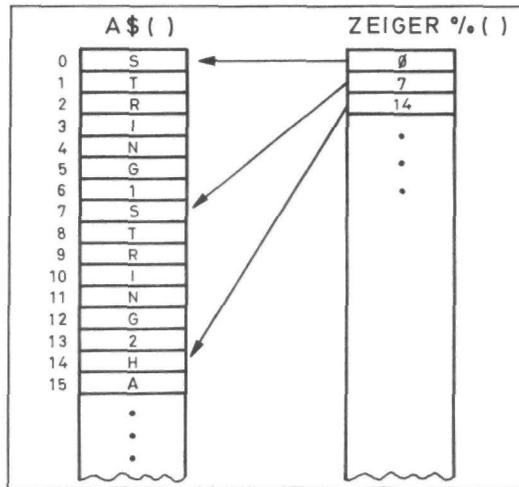
Auch in anderen Fällen kann etwas ähnliches passieren, wenn als Endwert einer Integer-Schleife ein Wert eingesetzt wird, der das Maximum oder Minimum eines Zahlenbereichs darstellt, etwa durch die Konstruktion

```
for i%=0 to 32767
bei „normalen“ Integers.
```

## Strings festgelegt

Der Speedcompiler hat eine etwas „eigen-sinnige“ String-Verarbeitungs-Konzeption, die sich sehr stark von der des Interpreters unterscheidet. Die Strings werden nicht wie in BASIC meist üblich dynamisch verwaltet, sondern so wie in C und Pascal statisch im Speicher gehalten. Das hat den Vorteil einer wesentlich höheren Verarbeitungsgeschwindigkeit und des gänzlichen Fehlens der störenden Garbage-Collection. Programme, die ihre Strings nicht variabel anlegen müssen und viel String-Verarbeitung benutzen, werden durch den Speedcompiler nicht nur schneller, sondern oft überhaupt erst möglich, zum Beispiel Editoren.

Doch wo Licht ist, ist auch Schatten: Das Fehlen einer dynamischen String-Verarbeitung macht es oft nötig, die benutzten



So werden die Strings im Puffer gespeichert

## Bugs und Mißverständnisse

In den Abteilungen „Compiler-Optionen“ und „Inkompatibilitäten“ haben sich Fehler beziehungsweise Unklarheiten in die Beschreibung eingeschlichen:

Die Definition einer Funktion durch DEF FNname(X) muß allein in der Zeile stehen, Schachtelungen in der Form (FN(X)-ausdruck\*FN(X)) sind nicht möglich.

Die Festlegung der String-Länge (nicht von String-Arrays) hat für den String X\$ und die Länge 10 folgende Syntax: REM£\$X\$(10). Die Einzel-Strings eines String-Arrays werden durch ein REM£Länge direkt hinter der Dimensionierung begrenzt, das REM muß natürlich durch einen Doppelpunkt vom DIM-Befehl getrennt werden. In Zeile 205 im Listing auf Seite 12 (Ausgabe 10/87) ist nachzulesen, wie das geht.

In Sachen „String-Länge“ gibt es auch einen kleinen Fehler im Compiler selbst. Bekanntlich kann ein Programm mit der Option „Bereichsüberprüfung ein“ kompiliert werden. Diese Überprüfung läßt allerdings bei einzelnen Strings ein Byte mehr als vorgegeben zu. Bei String-Arrays tritt dieser Fehler nicht auf.

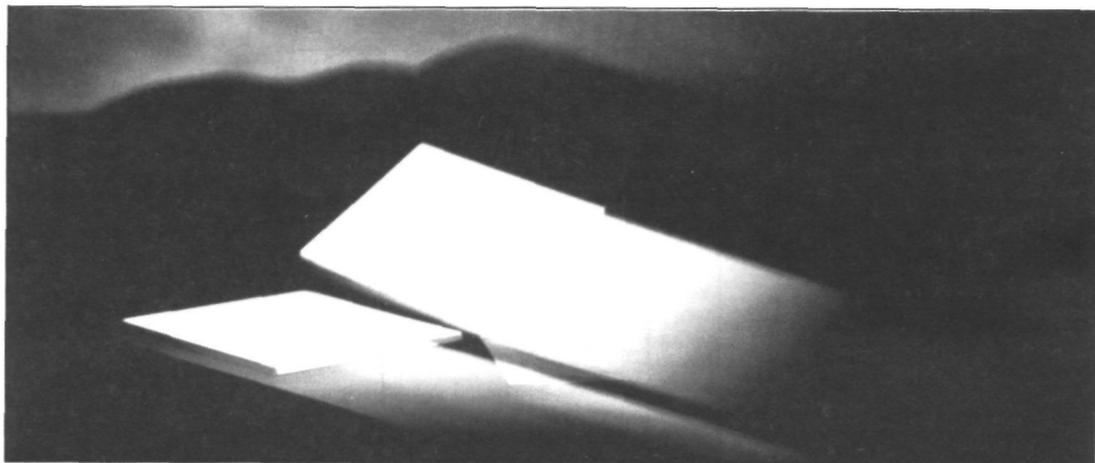
Strings ziemlich stark einzuschränken. Der Programmierer muß im voraus einschätzen, wie lang der einzelne String maximal werden kann, und dies dem Compiler mitteilen. Bei String-Arrays ist das besonders schwierig, da man die Maximal-Länge für das ganze Array angeben mußte. Wenn man nun viele Strings zwischenspeichern muß, deren Länge sehr stark variiert, so kann das zu Speicherproblemen führen. Ein Patentrezept für solche Probleme gibt es leider nicht. Aber für alle, die sich zutrauen, auch einmal in „fremden“ Datenstrukturen zu denken, hier ein Trick:

Wie bereits erwähnt, unterstützt der Speedcompiler den Datentyp „Array of char“. Das ist ein String-Array mit String-Länge '1', das vom Compiler besonders kompakt angelegt wird. Will man eine große Zahl unterschiedlich langer Strings zwischenspeichern, so kann man ganz einfach jedes Zeichen der einzelnen Strings hintereinander jeweils einem Element des Arrays zuweisen, wobei für jeden String ein Zeiger den Anfang des Strings markiert (siehe Bild). Folgende Definitionen beziehungsweise Unterprogramme sind dabei von Nutzen:

```
10 dim a$(10000):rem £1
15 rem ARRAY OF CHAR als Puffer
17 rem verbraucht nur 10001 Bytes
20 dim zeiger%(1000)
25 rem Zeiger auf 1001 Stringanfänge
27 rem verbraucht nur 2002 Bytes
```

```
30 frei%=0: rem Zeiger auf erstes
32 rem freies Pufferelement
40 anz%=0: rem Anzahl der Strings
50 zeiger%(0)=0
100 rem Schreiben eines neuen Strings
110 rem in den Puffer
120 rem s$ ist der String
140 for i%=1 to len(s$)
150 a$(frei%+i%-1)=mid$(s$,i%,1)
160 next i%
170 frei%=frei%+len(s$)
180 anz%=anz%+1
185 zeiger%(anz%)=frei%
190 return
200 rem Lesen eines bestimmten Strings
210 rem aus dem Puffer
220 rem nr% ist die Nummer des Strings
230 rem s$ enthält dann den String
240 s$=""
250 for i%=zeiger%(nr%) to
zeiger%(nr%+1)-1
260 s$=s$+a$(i%)
270 next i%
280 return
```

Die nächste und vorläufig letzte Folge behandelt die „Maschinensprachschnittstelle“ des Compilers und den internen Aufbau kompilierter Programme und deren Datenstrukturen (Ablage der Variablen im Speicher). Mit diesen Daten und einigen Ideen kann ein guter Programmierer dann auch herausfinden, wie er dieselben Variablen in mehreren unabhängigen Programmen nutzen kann. Jürgen Lindemeyer/JS



C-Studio

# Form und Farbe - bewegte Bilder

Spiel: VisHypnotica

Die Gefahr der Selbsthypnose besteht aber nicht, da Sie aktiv am Zustandekommen des Bildes beteiligt sind. Sie steuern einen Shape (wörtlich: Gestalt oder Form) in allen Himmelsrichtungen über den Bildschirm. Diese Gebilde – eben die Shapes – können sehr komplexe Formen und eine unterschiedliche optische Verweildauer haben.

Da die Steuerung (insbesondere die mit dem Joystick) eine sehr schnelle Bewegung ermöglicht, kann der Shape sich quasi selbst überholen und/oder teilweise überlagern.

## Einsichten in die Bedienung . . .

Sie müssen mit F1 festlegen, ob Sie die Richtungssteuerung mit einem Joystick vornehmen wollen, oder auf die Tastatur zurückgreifen müssen.

Die acht Bewegungsrichtungen sind über das Tastenkreuz 'Q', 'W', 'E', 'A', 'D', 'Z', 'X' und 'C' oder über den Joystick zu steuern.

**Wenn Sie den weichen sich wiederholenden Bewegungen längere Zeit konzentriert zusehen, werden Sie verstehen, warum der Autor sein Programm mit visueller Hypnose in Verbindung bringt. Die erzeugten Momentaufnahmen haben gleichzeitig etwas Beruhigendes und Anregendes an sich.**

Wenn Sie nur die Richtung verändern, geschieht nichts Aufregendes. Erst wenn Sie gleichzeitig den Aktions-Knopf beziehungsweise bei Tastatursteuerung die SHIFT-Taste drücken, wird der eingestellte Shape über den Bildschirm bewegt und zieht je nach Festlegung einen mehr oder weniger langen Schwanz hinter sich her.

Das Programm verfügt zur Zeit über 9 verschiedene Shapes, so daß es naheliegender ist, die direkte Ansteuerung über die Zifferntasten '1' bis '9' erfolgen zu lassen.

Möchten Sie die Geschwindigkeit verändern, bedienen Sie sich einfach der '+' oder '-' Taste. Da diese Tasten nur das Intervall der Joystickabfrage beeinflussen, haben sie auf die Tastaturabfrage keine Auswirkung.

## . . . und Aussichten auf Erweiterungen

Wir planen, für dieses Programm in einer der nächsten Ausgaben einen Shape-Editor zu veröffentlichen, und vielleicht gelingt es auch noch, eine hintergründige Musik mit in das Programm einzubinden.

Wenn Sie jetzt froh sind, mal ein Programm zum Entspannen gefunden zu haben, das nicht krächzt und piept, versprechen wir Ihnen, daß der Ton auch abschaltbar sein wird. WM

# Einer gegen Alle

## Chiffrierungs-Rätsel wieder gelöst

Diesmal wurde es echt spannend, denn so lange hat noch keine Codierung Bestand gehabt. Selbst am Ende der Einsendefrist waren weniger als ein Dutzend Antworten bei uns eingetroffen. Nur, um noch einmal die Spielregeln in Erinnerung zu rufen, eine einzige eingeschickte Lösung hätte auch gereicht.

## Lösung und Gewinner

Die Dechiffrierung der Aufgabenstellung aus der Ausgabe 9/87 können Sie innerhalb von INPUT 64 verfolgen. Außerdem können Sie das Originalprogramm zum Chiffrieren und Dechiffrieren von

**Eigentlich ist unser Chiffrierungs-Rätsel ein ungleicher Kampf, aber darin liegt ja auch der besondere Reiz der Aufgabe. Gelingt es einem Einzelnen, eine Chiffrierung zu entwickeln, die von 100 000 Computertans nicht durchschaut wird?**

M. Schmidt aus Gailingen auf Ihren Datenträger überspielen und anschließend analysieren.

An dieser Stelle nur ein Hinweis zu diesem Lösungsprogramm. In der Zeile 1100 muß der Variablen AS der zu chiffrierende Text übergeben werden. Da in den folgenden Rechenroutinen die RND-Funktion ihre Finger im Spiel hat, wird es nicht verwundern, daß jeder Durchlauf eine andere Code-Folge erzeugt.

Aus den Einsendungen wurden die folgenden Gewinner ermittelt: Anja Fisch, 6000 Frankfurt; D. Künstler, 6501 Dienheim; Georges Philippe, L-4734 Petange; Johann Wagenhäuser, 6000 Frankfurt; Rüdiger Schink, 4230 Wesel. Den Gewinnern unseren herzlichen Glückwunsch.

## Neues Spiel, neues Glück

Die neue Aufgabe wurde von Arnold Hüwe aus Ratingen entwickelt. Die üblichen vier Code-Folgen finden Sie auch diesmal wieder in einem kleinen Programm, das Sie aus INPUT 64 abspeichern können. Dieses Programm kann auch als Rahmenprogramm für Ihre Versuche dienen, den Codierungs-Schlüssel zu knacken.

Nur sollten Sie sich mit diesen Versuchen nicht allzuviel Zeit lassen. Lösungen, die nach dem 2. Januar 1988 bei uns eintreffen, können nicht mehr gewertet werden.

## Die Spielregeln

Mit dieser Ausgabe stellen wir Ihnen bereits die vierte Dechiffrierungs-Aufgabe. Für alle neu hinzugekommenen INPUT-Leser hier noch einmal die Spielregeln: Wir geben jedesmal eine chiffrierte Code-Folge vor und erwarten von Ihnen nicht mehr und nicht weniger als den Klartext, der sich hinter dieser Code-Folge versteckt. Um Ihnen die Aufgabe zu erleichtern, erhalten Sie mehrere Code-Folgen (natürlich mit dem gleichen Algorithmus codiert), wovon wir Ihnen für zwei auch gleich den Lösungstext mitliefern.

Natürlich können Sie auch was gewinnen. Wenn Ihr Chiffrierungs-Programm (Sie können also auch ein Programm einschicken) innerhalb von vier Wochen nicht geknackt wird, bekommen Sie ein INPUT-64-Jahresabonnement. Sie spielen also alleine gegen 100 000 INPUT-Leser. Sollten – was bisher jedesmal der Fall war – in dieser Frist richtige Lösungen bei uns eintreffen, verlosen wir unter den Einsendern fünf Bücher, während der Programmentwickler dann leer aus geht.

Noch eine Bemerkung zu den Chiffrierungs-Programmen. Es sollte theoretisch auch die Möglichkeit bestehen, den Code zu knacken. Eine Verknüpfung beispielsweise mit dem Geburtstag des Autors (oder mit anderen externen Zahlen oder Texten) ist zwar ein sehr wirkungsvoller Codier-Schlüssel, bietet aber kaum Ansatzpunkte für eine Dechiffrierung. Also: alle Informationen, die für die Dechiffrierung benötigt werden, müssen als Code irgendwie und irgendwo in der gesamten Code-Folge auch enthalten sein.

(Beachte: Die Post braucht über die Feiertage noch mehr Zeit als sonst!)

Abschließend noch eine Aufforderung an Sie: Entwickeln Sie doch mal ein Chiffrierungs-Programm! Anregungen dürften in den bisher veröffentlichten Beispielen genug enthalten sein. Wenn Sie die Spielregeln einhalten, können Sie nur gewinnen.

WM

## INPUT 64 BASIC—Erweiterung

Die BASIC-Erweiterung aus INPUT 64 (Ausgabe 1/86), gebrannt auf zwei 2764er EPROMs für die C-64-EPROM-Bank.

Keine Ladezeit mehr — über 40 neue Befehle und Super-Tape integriert.

**Preis: 49,— DM  
zuzüglich 3,— DM für Porto  
und Verpackung  
(V-Scheck)**

**Bestelladresse:  
Verlag Heinz Heise  
GmbH & Co KG  
Postfach 61 04 07  
3000 Hannover 61**

# Farbe in Grautöne wandeln

**Graudruck: Multicolorgrafik zu Papier gebracht**

Dieses Druckprogramm ermöglicht es, Ausdrücke vom Multicolorgrafik-Bildschirm auf normalen Scharzweiß-Druckern zu erstellen, wobei die 16 möglichen Farben des C64 auch alle als unterschiedliche Graustufen dargestellt werden.

Im Multicolor-Modus können Sie auf dem Bildschirm waagrecht 160 Punkte und senkrecht 200 Punkte darstellen. (Weitergehende Erklärungen zur Bildschirmorganisation finden Sie in INPUT 9/87 ab Seite 3.) Das Programm ordnet nun jedem farbigen Punkt auf dem Bildschirm einen Block von 4 \* 4 Punkten auf dem Drucker zu. Je nachdem wie viele dieser 16 Druckpunkte gesetzt sind, ergibt sich ein hellerer oder dunklerer Eindruck.

## Die Drucktechnik, . . .

Um die Berechnungen zu beschleunigen, werden die Graustufen-Blöcke einer ganzen Bildschirm-Zeile (160 \* 8 Multicolor-Punkte) zuerst in einem Puffer gesammelt und anschließend auf den Drucker ausgegeben.

Auch bei der Ausgabe wendet das Programm einen Trick an. Es werden zuerst die ersten und dritten der vier waagerechten Punkte eines Blocks gedruckt, dann das Papier um 1/3 Nadelbreite nach oben geschoben, um anschließend die zweiten und vierten Punkte zu drucken. Dieses Verfahren findet auch beim Drucken von NLQ-Schrift auf 8-Nadel-Matrixdruckern statt.

Als Peripherie benötigen Sie daher einen Matrixdrucker, auf dem ein 8-Nadel-Bitmustermodus mit 640 Punkten pro Zeile einstellbar ist und der es ermöglicht, das Papier in Schritten einer drittel Nadelbreite nach oben zu schieben (zum Beispiel Epson- oder Star-Drucker; nicht jedoch die MPS-Drucker von Commodore).

Das eigentliche Druckprogramm wird erst außerhalb von INPUT 64 von den jeweiligen Generierungs-Programmen erzeugt. Wir

**Farbmonitore gehören heute schon zur Standardausrüstung; farbfähige Drucker hingegen sind in der Regel noch recht kostspielig und von daher nicht so verbreitet. Aber was nutzt die schönste mehrfarbige Computergrafik, wenn sie nicht dauerhaft festgehalten werden kann. Natürlich können wir Ihrem Drucker nicht nachträglich beibringen, farbige Ausdrücke zu Papier zu bringen, aber „was man grau auf weiß besitzt, kann man auch nach Hause tragen.“**

bieten Ihnen zwei Versionen an. Die erste erzeugte Version liegt am BASIC-Anfang und kann direkt an das BASIC-Programm angebunden werden, wenn Sie vor dem Abspeichern POKE 43,1 und POKE 44,8 eingeben. Die zweite Version verschiebt sich nach dem Starten in den Bereich \$CD00 bis \$CFFF, stört zwar damit nicht mehr im BASIC-Bereich, muß aber jedesmal wieder vor dem eigentlichen Programm geladen werden.

## . . . ihre Anpassung . . .

Zusätzlich erhalten Sie ein kleines Testprogramm, mit dem Sie Ihre Drucker-Einstellungen austesten können. Leider können wir Ihnen diese Einstellung nicht abnehmen, da nahezu jeder Drucker – gerade bei der Ansteuerung des Bitmustermodus – andere Zeichenfolgen erwartet.

Wenn Sie außerhalb von INPUT 64 eines der beiden Generierungs-Programme starten und die Taste 'A' drücken, können Sie auf einer Eingabemaske alle notwendigen Druckerparameter und die Anfangsadresse des Puffers eingeben. (Die Voreinstellung trifft für einen Epson FX-80 zu.)

Wollen Sie Änderungen vornehmen, beachten Sie bitte, daß Sie Ihre Eingabe in hexadezimaler Schreibweise vornehmen müs-

sen, wobei ein zu sendender Code immer aus zwei Zeichen bestehen muß (aus 14 dez wird also 0e hex) und zwischen den einzelnen Codes kein Leerzeichen stehen darf.

Nach Drücken der RETURN-Taste werden die Eingaben soweit wie möglich auf Plausibilität geprüft. Entdeckt das Programm einen Fehler, zeigt es dies im unteren Bereich des Bildschirms an. Nach Druck auf eine Taste blinkt der Cursor auf dem fehlerhaften Wert. Ist alles in Ordnung, so gelangen Sie mit 'W' aus dem Editor heraus. Das Programm ändert nun die Parameter und erzeugt das eigentliche Druckprogramm als startbares Programm auf Diskette.

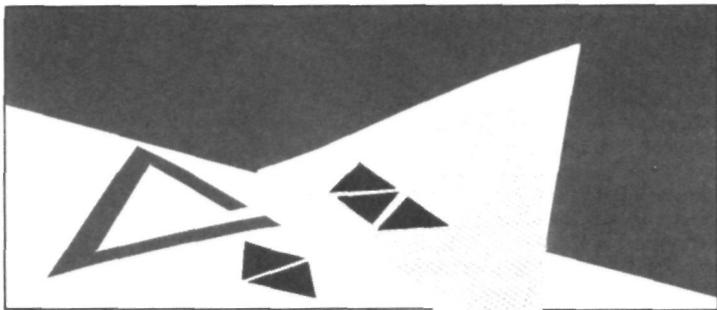
## . . . und das Ergebnis

Das Druckprogramm belegt weniger als 800 Bytes, und Sie können es von Ihrem eigentlichen Programm mit 'SYS 2133' beziehungsweise mit 'SYS 52480' aufrufen. Interessant ist sicherlich noch, daß sowohl der Bildschirmspeicher als auch der Farbspeicher unter dem ROM oder dem I/O-Bereich liegen dürfen. Auch der benötigte Pufferbereich des Druckprogramms kann von Ihnen in diese Speicherbereiche verbannt werden.

Das Testprogramm arbeitet nur mit der Version am BASIC-Anfang zusammen. Sie sollten das Druckprogramm (untere Version!) laden und starten. Danach laden Sie das Testprogramm nach und starten auch dieses. Auf dem Bildschirm wird ein mehrfarbiges Testbild erzeugt und anschließend das Druckprogramm angesprochen.

Auch wenn das einfache Bild nicht gerade ein Kunstwerk ist, so erfüllt es doch seinen (Test-)Zweck, und so Sie keinen Fehler bei der Anpassung gemacht haben, können Sie Ihren ersten Graudruck bewundern.

A. Voßloh/WM



# Trio Dominal

## Spiel: Triangles

Wie beim Original-Dominospiel geht es vor allem darum, alle eigenen Steine möglichst schnell ab- beziehungsweise anzulegen. Der Unterschied besteht darin, daß Ihnen statt rechteckiger – und das ist der Clou des Spiels – dreieckige Spielsteine zur Verfügung stehen.

### Aktion, Extras und noch mehr

Im Hauptmenü haben Sie die Möglichkeit zwischen Aktion, Extras, Dateien und Informationen zu wählen. Um sich erstmal einen Überblick über das Spiel zu verschaffen, lassen Sie sich einfach das Demo vorführen. Hier spielen drei Personen gegeneinander und zeigen Ihnen, wie's gemacht wird.

Zur Steuerung benutzen Sie entweder den Joystick in Port 2, dann erübrigen sich die weiteren Angaben, oder Sie spielen mit der Tastatur, dann wird der Pfeil mit der Taste 'A' nach oben, mit der Taste 'Z' nach unten, mit der Taste 'V' nach rechts und mit der Taste '.' nach links bewegt. Diese Tastaturbelegung gilt nicht nur im Hauptmenü, sondern auch im ganzen Verlauf des Spiels. Die Leer- oder Space-Taste entspricht dem Feuerknopf am Joystick.

Nachdem Sie über die Steuerung schon fast alles wissen, kann es richtig losgehen.

**Eine neue und außergewöhnliche Variante des bekannten Dominospiels bieten wir Ihnen mit „Triangles“. Die bekannten Platzprobleme beim Anlegen sind out: ein virtuelles Spielfeld bietet Raum für über 200 Steine. Wenn Sie wollen, brauchen Sie noch nicht einmal selber denken, denn Sie können hier auch spielen lassen.**

Bewegen Sie dazu den Pfeil auf den Begriff SPIELEN. Zum „Anklicken“ wird anschließend die Leer- oder Feuertaste betätigt. Ein Bonbon des Spiels ist die Möglichkeit, gegen eine Auswahl von acht Computern spielen zu können, die Sie auf der jetzt sichtbaren Bildschirmseite sehen. Auf der linken Seite des Bildschirms sind die Namen von acht Personen eingetragen, die Sie editieren oder durch eigene Namen ersetzen können. Die Namen der Computer sind fest eingetragen.

Um Ihre Chancen gegen die Computer zu erhöhen, können Sie diesen „menschliche“ Schwächen verleihen. Bewegen Sie dazu den Pfeil auf die rechts neben den Computern stehenden Ziffern und drücken Feuer. Bei jedem Druck erscheint über dem Namensfeld der Hinweis, wieviel Prozent der Steine vom Computer übersehen werden (0 = 0 %, 7 = 56 %).

### Allein gegen die Zeit

Um die einzelnen Teilnehmer ins Spiel zu bringen, müssen sie erst aktiviert werden. Dazu wird der Pfeil bei den „menschlichen“ Spielern hinter und bei den Computerspielern vor den Namen gesetzt und mit Feuer bestätigt. Erscheint ein Haken, kann der Spieler am Spielgeschehen teilnehmen.

Nach RETURN können Sie festlegen, welches die höchste Ziffer auf den Spielsteinen sein soll (2 bis 7). Nach erneutem Druck auf die RETURN-Taste werden Sie gefragt, mit was für einem Stein das Spiel begonnen werden soll: Ob mit einem, der drei gleiche Ziffern trägt, oder einem beliebigen anderen. Dann beginnt der, der den Stein mit der höchsten Augenzahl hat. Noch einmal RETURN, und sie sind im Spiel.

### Endlich geht's los

Es wird angezeigt, wer das Spiel beginnt. Sie bestätigen dann mit der Feuer- oder Leertaste. Danach teilt das Programm Ihnen mit, wer als nächster am Zug ist. Zum Bestätigen einer Eingabe oder zum Weiterkommen im Spiel wird immer die Feuer- oder Leertaste akzeptiert.

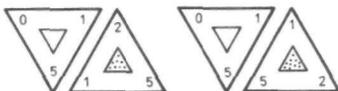
Für die Tastaturbelegung gelten folgende Regeln:

#### In der Auswahlleiste:

**Links (L), rechts (R):** Steine nach links oder rechts bewegen. In Verbindung mit der Leertaste ist ein schnelles Bewegen möglich.

**Hoch (A)** setzt den ausgewählten Stein in das Spielfeld. Wenn der Pfeil auf „ZIEHEN“ steht, wird ein neuer Stein geholt (falls vorrätig).

**Runter (Z)** wechselt von der Auswahlleiste in das Spielfeld. Das ist aber nur möglich, wenn vorher ein Stein ausgewählt wurde.



**Man kann auch punkten, wenn man den richtigen Stein an der richtigen Stelle anlegt.**

## Tastenfunktionen im Spielfeld:

**Links (L)** bewegt den Stein im Spielfeld nach links. Leertaste und links (L): der Stein wird auf die Seite gekippt, die Zahlen werden entgegen dem Uhrzeigersinn gedreht.

**Rechts (R)** bewegt den Stein im Spielfeld nach rechts. Leertaste und rechts (R): der Stein wird auf die Spitze gekippt, die Zahlen werden im Uhrzeigersinn gedreht.

**Hoch (A)** bewegt den Stein im Spielfeld nach oben. Leertaste und hoch (A): wechselt vom Spielfeld in die Auswahlleiste.

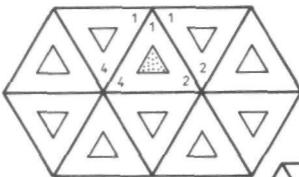
**Runter (Z)** bewegt den Stein im Spielfeld nach unten. Leertaste und runter (Z): der Stein wird an der aktuellen Position angelegt, sofern das möglich ist.

## Punkte, Punkte, Punkte

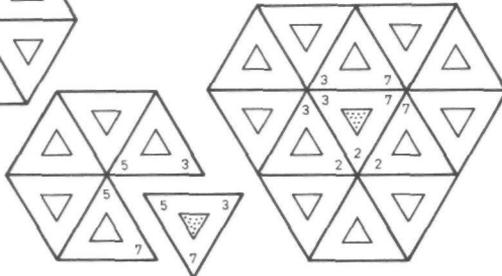
Kommen wir jetzt zur Bewertung beziehungsweise zur Vergabe der Punkte. Derjenige Spieler, der den ersten Stein setzt, bekommt 10 Bonuspunkte, wenn er einen Stein mit drei gleichen Ziffern in das Spiel bringt. Dazu wird die Augenzahl des Steins addiert. Fängt er mit einem beliebigen an, gibt es keinen Bonus. Beim Anlegen wird die Augenzahl des jeweiligen Steins gutgeschrieben.

Besondere Prämien gibt es, wenn Sie durch das Anlegen eines Steins bestimmte Figuren erstellen (siehe Zeichnungen):

- **Einfaches Sechseck:** 50 Bonuspunkte
- **Doppeltes Sechseck:** 100 Bonuspunkte
- **Dreifaches Sechseck:** 150 Bonuspunkte
- **Einfache Brücke:** 40 Bonuspunkte
- **Doppelte Brücke:** 40 Bonuspunkte



Einfache, doppelte und dreifache Sechsecke bringen zwar die meisten Punkte, sind aber auch am schwierigsten zu erstellen.



Wird eine Brücke in Verbindung mit einem Sechseck erzielt, gibt es dafür 90 Bonuspunkte.

Wenn sich in der Auswahlleiste kein Stein mehr befindet, den man anlegen kann, muß man einen Stein ziehen. Das geschieht solange, bis ein passender gefunden ist. Jeder Stein, der gezogen werden muß, schlägt sich mit 5 Minuspunkten auf dem Punktekonto nieder. Ist kein Stein mehr im Talon (dem „Vorratsbehälter“) vorhanden, werden Ihnen sogar 10 Punkte abgezogen.

**G-Taste:** Solange der Name des Spielers angezeigt wird, kommt man mit dieser Taste in eine Zwischentabelle. Mit der Leertaste geht es wieder zum Spiel. In dieser Tabelle werden folgende Abkürzungen benutzt:

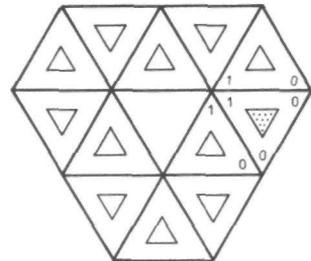
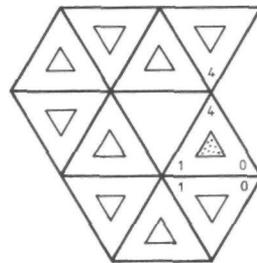
- AN = Stein angelegt
- 1S = Einfaches Sechseck gelegt
- 2S = Doppeltes Sechseck gelegt
- 3S = Dreifaches Sechseck gelegt
- BR = Einfache Brücke gelegt
- DB = Doppelte Brücke gelegt
- SB = Sechseck + Brücke gelegt
- OZ = Nicht gezogen

...  
35Z = 35 mal gezogen  
...

## Fast Endlos

Das Spiel ist beendet, wenn:

1. ein Spieler alle Steine abgelegt hat. Alle anderen Mitstreiter dürfen danach noch einen Zug machen. Der Spieler, der als erster seine Steine abgelegt hat, bekommt 25 Bonuspunkte plus die Gesamtaugenanzahl aller Steine, die bei den anderen Mitspielern noch nicht angelegt sind.

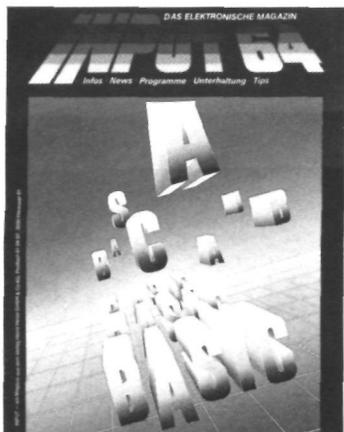


Eine Brücke kann man schon mal schnell aufbauen (oben), aber in Verbindung mit einem Sechseck wird's schon schwieriger.

2. keiner der Spieler mehr Steine anlegen und auch keine mehr ziehen kann. Dann werden jedem Spieler die Gesamtaugenanzahl aller noch nicht angelegten Steine vom Punktekonto abgezogen und so der Sieger bestimmt.

Nachdem der Sieger ermittelt wurde, ist eine große Übersicht in Form einer Tabelle zu sehen. Dabei werden die Tabellen von zwei Spielern gleichzeitig dargestellt. Es können alle Tabellen nacheinander sichtbar gemacht werden, wenn man die Leertaste und die Taste für rechts beziehungsweise für links drückt. Die Daten aller Spieler werden jetzt in tabellarischer Form angezeigt. In diesen Tabellen kann man sich mit den Tasten für rauf und runter alle Züge des Spiels ansehen. Mit der Leertaste und den Tasten für links oder rechts, werden weitere Tabellen angezeigt. Diese Tabellen und die Namen der Spieler können Sie im Hauptmenü auf Ihre Diskette abspeichern und wieder laden. Zum Spielanfang kommt man wieder mit der Leertaste und der Taste für rauf. Das Spiel kann wieder beginnen. kfp

Am 8. Januar an Ihrem Kiosk:  
INPUT 64, Ausgabe 1/88



Wir bringen unter anderem:

### Ist BASIC

INPUT-Struktur-BASIC ist unser Angebot, das Thema Spaghetti-Programme in das Reich der Vergangenheit zu verbannen. Mit **Ist BASIC** ist BASIC eine Programmiersprache, die im Hinblick auf Strukturbefehle kaum noch Wünsche offen läßt. Auch wenn Ihnen jetzt "call label", "repeat - until" und "do - loop - exit" vielleicht wenig sagen, die neuen Möglichkeiten, effizientere Programme schreiben zu können, werden Sie bald nicht mehr missen wollen. Nicht zuletzt werden wir Ihnen zeigen, wie Sie **Ist BASIC** mit INPUT-BASIC (die Spracherweiterung aus 1/86) verbinden und gleichzeitig(!) anwenden können.

### PW-Coder

Praktizierter Datenschutz: Jetzt können Sie auch sensible Daten einem elektronischen Datenträger anvertrauen. PW-Coder schützt Programme und Daten vor unberechtigten Zugriff, denn ohne Paßwort läuft nichts.

### Lohnsteuer 1987

Auf ein Drittes! Es wäre ja richtig langweilig, wenn es im Steuerrecht keine neuen Gesetze, Richtlinien und Verordnungen geben würde. Mit der erneut aktualisierten und komfortableren Version können Sie im Januar schon berechnen, was das Finanzamt endlich im August rückerstattet. Oder – sofern Sie die Wahl haben – überprüfen, ob sich ein Jahresausgleich für Sie überhaupt lohnt.

### StraMax

Mustererkennung ist ein neumodisches Reizwort in der computerunterstützten Produktion. Zeigen Sie in diesem Spiel Ihrem C64, daß WIR das doch besser können, vor allem, wenn außerdem noch strategisches Geschick gefragt ist.

## c't – Magazin für Computertechnik

Ausgabe 1/88 – ab 11. Dezember 1987 am Kiosk

Projekt: CP/M-Karte für C64 – Software, Transputer-Entwicklungspaket, 512K-Erweiterung für CPC \* Software-Know-how: Multitasking im Amiga-DOS, Volumenberechnung in BASIC, Betriebssystem EUMEL \* Programm: digitales Filter in Echtzeit, CGA-Emulator \* Prüfstand: Turbo-Pascal Version 4.0 \* u.v.a.m.

## elrad – Magazin für Elektronik

Ausgabe 1/88 – ab 28. Dezember 1987 am Kiosk

Projekt: EPROM-Programmiergerät für 22 verschiedene Typen \* Schrittmotorsteuerung Teil 3: Teach-In \* µPegelschreiber Teil 4: Mitlauf-Filter und NF-Endstufe \* Die elrad-Laborblätter: Schaltungen mit Analog-Multiplizierern \* u.v.a.m.



Bitte zum Entnehmen der Diskette die Perforation  
an den markierten Stellen aufreißen.



12/87

DAS ELEKTRONISCHE MAGAZIN  
**INPUT 64**  
*Infos · News · Programme · Unterhaltung · Tips*

DAS ELEKTRONISCHE MAGAZIN  
**INPUT 64**  
*Infos · News · Programme · Unterhaltung · Tips*

Ein Magazin aus dem Verlag Heinz Heise GmbH · Postfach 610407 · 3000 Hannover 61

**HEISE**



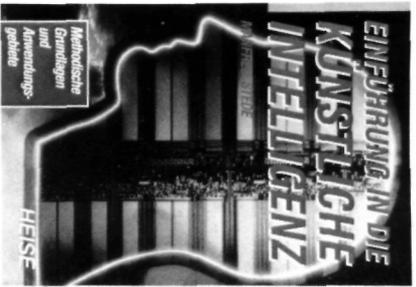
# K I Die Computer- anwendung von morgen.



Eine solide Einführung in die Hauptprinzipien der KI-Programmierung. Beschrieben wird, was künstliche Intelligenz ist und wie sich die Entwicklung Schritt für Schritt dahin vollzogen hat. Die Problem-Definition ist ein Schwerpunkt und wird an zahlreichen Beispielen und Methoden aufgeführt.



Theoretische Informationen über künstliche Intelligenz werden in konkreten Programme umgemünzt, die der Leser ausprobieren, verstehen und erweitern kann. Zum Experimentieren dienen dem fortgeschrittenen Hobby-Programmierer vor allem die Bereiche Suchverfahren und Spielstrategie.



Der umfassende Einblick in diesen hochaktuellen Bereich der Computerprogrammierung ermöglicht es dem Leser, sich sein eigenes Urteil über Chancen und Grenzen der künstlichen Intelligenz zu bilden. Die methodischen Grundlagen der KI und ihre wichtigsten Anwendungsfelder werden vorgestellt.

# COMPUTER- BUCH

Broschur, 243 Seiten  
DM 44,80  
ISBN 3-88229-012-9

Broschur, 219 Seiten  
DM 44,80  
ISBN 3-88229-126-5

Broschur, 267 Seiten  
DM 49,80  
ISBN 3-88229-018-8

Verlag  
H. Heise GmbH  
Postfach 61 04 07  
3000 Hannover 61



Im Buch-, Fachhandel oder beim Verlag erhältlich. KI/12