

ENTWICKLUNG

EUG LORCH

INFO

FEB/MAR/APR

1/88

TESTS:



LISTING- TEIL:

**MARKET
ORIENT EXPRESS**

**AUTORUN
MASTER MIND
GRAPHIK-UTILITY**



**SOUND-
EINFÜHRUNG**

BASIC-KURS

Vorwort:

Mit diesem Info, also insgesamt dem siebten, beenden wir den gut angekommenen BASIC-Kurs und den zweiteiligen EXDOS-Kurs. Mal sehen, was wir als nächsten Kurs bringen werden. Außerdem ist in diesem Info eine Sound-Einführung enthalten, die sich über 2 1/2 Seiten erstreckt und dieses Kapitel etwas mehr durchleuchtet soll.

Unsere diesmaligen Softwaretests enthalten den Leckerbissen für alle Basic-Programmierer, den Zzzip, der von Werner Lindner vertrieben wird und zwei Spiele. Im Testteil des nächsten regulären Infos (da wäre noch eine kostenlose Sonderausgabe, die Sie vor oder nach dieser Info erhalten werden) werden wir einige Programmiersprachen testen, die düsi software in den Vertrieb nehmen wird.

Die erste EUG Listings-Sammlendiskette ist inzwischen vollendet und kann gegen DM 20.-- bei Stephan Ehrmann bestellt werden. Auf ihr sind alle Listings bis einschließlich denen des sechsten Infos und einige andere, noch nicht veröffentlichte Programme. Schon allein wegen WP*, der Erweiterung für verschiedene Schriften innerhalb des Word Processors, dem EUG Racer und Tron dürfte sich die Anschaffung für die Clubmitglieder lohnen.

In eigener Sache möchten wir noch kurz berichten, daß wir inzwischen auch begeisterte Mitglieder in Jugoslawien, Dänemark und der DDR haben, die natürlich noch eine Minderheit darstellen, jedoch sehr froh sind, daß sie sich an jemanden wenden können, wenn sie Probleme haben. Grüße aus Deutschland!

Bis bald,

Stephan Ehrmann

Inhalt:

Tests	3
Listing-Teil	4
Sound-Einführung	5
Tips & Tricks	7
Hardware-Basteltip	9
EXDOS-Einführung 2. Teil	9
BASIC-Kurs letzter Teil	11
Master Mind, Spiel zum Abtippen	12

Eine Publikation der ENTERPRISE USER GROUP. Entnahmen oder Teilentnahmen nur mit Genehmigung der EUG. Verantwortlich für den Inhalt ist Stephan Ehrmann; Beiträge sind von Stephan Ehrmann und verschiedenen Mitgliedern.

Kontaktadresse:

ENTERPRISE USER GROUP
z. Hd. Stephan Ehrmann
Eberrainweg 11

7073 Lorch

Denkt daran: Wir freuen uns über jedes eingesandte Listing, über alle Artikel und selbstverständlich auch über alle Tips und Tricks, ob diese nun kurz oder lang sind.

Software-Tests :

Programm : Zzzip
Programmart : Integer Basic Compiler
Hersteller : Boxsoft, England
Preis : DM 39.--
Vertrieb : Werner Lindner
Hard- und Softwareideen
Landsberger Str. 49
8913 Schondorf a.A.

Besonderes :

Grundsätzlich kann man Basic Compiler in zwei Arten unterteilen: Die Integer und die Floating Point Compiler. Der Zzzip ist ein Integer Compiler, der Basicprogramme in Maschinenspracheprogramme übersetzt.

Er kann im Prinzip nur mit ganzen Zahlen rechnen, er beherrscht also die Fließkommatechnik nicht. Diese wird auch nur bei mathematischen Operationen benötigt. In der Werbung wird der Faktor 100 als Maximum der späteren Geschwindigkeit angegeben. Diese Geschwindigkeit gilt selbstverständlich nicht in allen Bereichen. So kann der Zzzip zum Beispiel Graphik in der Geschwindigkeit nur ca. verdoppeln.

Bildschirm Ausgaben durch Print, Inputroutinen, Sound, Tastaturabfragen und viele andere Dinge werden durch den Zzzip schon gewaltig schneller. Nachteile jedoch findet man auch hier: Die USING-Ausgabe auf Drucker und Bildschirm wird nicht beherrscht, Außerdem kann er einige Stringoperationen wie UCASE\$ bei deutschen Sonderzeichen nicht richtig compilieren. Das comfortable IS-Basic läßt normalerweise die Initialisierung von Variablen zu, die der ENTERPRISE bereits verwendet (also z. B. in DEF-Procedures usw.). Hier macht Zzzip ebenfalls halt. Er kann nur Variable als solche erkennen, wenn IS-Basic diese wirklich nicht verwendet. Ansonsten gibt er die Fehlermeldung DECLARED TWICE zurück.

Über die Fehlermeldungen des Zzzip kann man sich nicht beschweren, sie sind alle im Handbuch erläutert (die englische Sprache läßt bei Englisch-Laien jedoch oft einen Schmerz zurück), alle nicht verstandenen Optionen werden auf dem Bildschirm genau erklärt. Das Compilieren erfolgt in 4

Abschnitten, in denen der Source (= Quellcode) auch geladen und das spätere Programm gleich abgespeichert wird. Hier wird ein Lader unter dem frei eingebbaren Namen erzeugt, der das Programm mit dem Zusatz '.Z' lädt. Das Laden braucht nun etwas länger, was jedoch auch bei Cassette nicht sehr stört.

Abschließend möchte ich über Zzzip sagen, daß er für Spiele, Dateiverwaltungen und ähnliches sehr gut geeignet ist, die Nicht-Anerkennung von USING muß leider übersehen werden. Auch die Fließkommatechnik wird in den meisten Basic-Programmen nicht benötigt. Wenn Sie also diese beiden Hindernisse nicht haben, so empfehle ich Ihnen den Zzzip sehr, da er für viele Anwendungen wirklich gut arbeitet.

S.E.

Programm : Market
Programmart : Börsensimulation
Hersteller : ENTERPRISE
Preis : DM 29.--

Besonderes :

Market ist ein Spiel, das die Spekulationen an der Börse simuliert. Dabei wird der Spieler am Anfang von Market gefragt, in welcher der 7 (!) Sprachen er spielen möchte. Hier sind u. a. auch Deutsch, Dänisch, Französisch, Italienisch usw. enthalten.

Der Spieler kann nun als Spekulant an der Börse fröhlich Aktien kaufen, bis er schließlich pleite geht. Dies geht allerdings oft sehr schnell, da das Spiel nicht nach einem besonderen Prinzip läuft, sondern rein zufällig die Kurse steigen oder fallen läßt. In dem Menü können Sie die Option 'Tips' anwählen, in der Market Ihnen z. B. sagt, daß eine Firma einen großen Auftrag bekommen hat. Kaufen Sie nun viele dieser Aktien, kann es durchaus sein, daß diese Firma kurz nach Ihrem Kauf bankrott geht.

Wären diese Tips und Reaktionen nicht zufällig, so wäre das Spiel bestimmt ganz lustig, so jedoch ist es völlig unrealistisch. Das Schönste an dem ganzen Programm ist das Titelbild in der Cassette.

Programm : Orient Express
Programmart : Strategie/Actionspiel
Hersteller : AIP Ltd.
Vertrieb : ENTERPRISE
Preis : DM 29.--

Besonderes :

Dieses Spiel ist auf dem ENTERPRISE eine neue Idee, obwohl es bereits 1985 in England erschienen ist. Ziel ist, einer fahrenden Lokomotive die Gleise und Schienen so zurechtzulegen, daß sie das Ziel am Bahnhof erreichen kann. Hier wird nach dem Schiebe-Verfahren gespielt: Das Spielfeld ist in viele Quadrate aufgeteilt, von denen eines frei ist. Möchten Sie nun ein anderes bewegen, so müssen Sie es in die Leerstelle schieben, so daß wieder eine solche entsteht.

Das Ganze verläuft ziemlich hektisch, obwohl der erste Schwierigkeitsgrad noch verhältnismäßig leicht durchzuspielen ist. Ab dem zweiten werden Sie erst nach einigen Versuchen keine Probleme mehr haben, das Spiel zu schaffen. Witzig ist auch, daß jeder Level eine andere Stadt darstellt: London, Paris, Brüssel usw. Natürlich herrscht dabei in London die kleinste Unordnung.

Das Fahren der Lokomotive kann dabei durch Drücken der Feuer/Space-Taste angehalten werden, was eine oben angezeigt Bahnhofsuhr zum Laufen bringt. Nach erneutem Druck kann die Uhr wieder angehalten und die Lokomotive wieder zum Fahren gebracht werden. Ist die Uhr abgelaufen, so fängt die Lokomotive automatisch wieder zu fahren an. Sind Sie nun mit dem Bauen der Strecke noch nicht weiter, so fährt die Lok bis zum ersten Hindernis und zerschellt dann.

Für Spielefreaks ist Orient Express sicherlich von den grafischen Ansprüchen her vielen anderen Spielen etwas unterlegen, der Spielewitz und die Idee sind jedoch toll.

Listing-Teil:

AUTORUN

Dieses Utility ist sehr kurz und doch sehr nützlich für die Anwender, die Ihre Programme anderen 'unzugänglich' machen wollen. Dieses Utility nämlich lädt Ihr Basic-Programm, setzt das Auto-startflag, d. h., beim Laden mit LOAD wird es automatisch gestartet, und speichert es dann unter dem gleichen Namen wieder ab.

:AUTORUN ruft das Programm auf, dann wird der Anwender nach dem Dateinamen gefragt. Dieser Dateiname ist der Name des Basic-Programmes, das Sie allerdings vor Aufruf von AUTORUN noch einmal unter einem anderen Namen abgespeichert haben sollten. Die Angabe von üblichen Diskettenparametern wie z.B. *.* ist nur möglich, wenn auf der Diskette alle Programme Basic-Programme sind. Ansonsten wird die Fehlermeldung 'Falscher Dateianfang' zurückgegeben.

Vor dem Aufruf muß das Utility selbstverständlich geladen werden. Dies geschieht mit :LOAD AUTORUN bzw. LOAD "AUTORUN". Das abgedruckte Basic-Listing mit den DATA-Zeilen erzeugt automatisch nach dem Starten mit START oder RUN das Binär-File, das Sie später laden können. Abspeichern sollten Sie das eingetippte Listing mit SAVE "AUTORUN.BAS" sicherheitshalber auch noch einmal, für den Fall nämlich, daß Sie sich vertippt haben.

Ob AUTORUN aktiviert ist oder nicht, erfahren Sie durch Eingabe von :HELP, denn AUTORUN ist ein speicherresidentes Modul, das in der HELP-Liste aufgeführt wird.

```
1000 CALL ERZEUGE("autorun")
1010 !
1020 DATA 0,6,142,1,0,0,0
1030 DATA 0,0,0,0,0,0,0
1040 DATA 24,61,7,7,7,7,7
1050 DATA 10,13,69,85,71,45,69,114
1060 DATA 119,101,105,116,101,114,117,110
1070 DATA 103,32,65,117,116,111,114,117
1080 DATA 110,10,13,40,99,41,32,49
```



```

1090 DATA 57,56,56,32,98,121,32,84
1100 DATA 46,83,99,104,119,105,110,110
1110 DATA 7,7,7,7,7,26,121
1120 DATA 61,61,40,30,61,192,120,183
1130 DATA 62,35,40,8,205,228,192,192
1140 DATA 62,103,14,0,217,79,6,0
1150 DATA 17,48,193,62,255,247,8,175
1160 DATA 217,201,205,228,192,192,26,144
1170 DATA 254,0,40,95,61,40,92,235
1180 DATA 18,213,8,17,31,193,1,17
1190 DATA 0,62,255,247,8,209,213,8
1200 DATA 79,6,0,19,62,255,247,8
1210 DATA 1,2,0,17,81,193,62,255
1220 DATA 247,8,209,62,250,247,1,32
1230 DATA 56,62,250,14,1,17,7,193
1240 DATA 247,10,32,45,62,250,247,5
1250 DATA 32,39,120,254,4,32,32,62
1260 DATA 250,14,1,17,247,192,247,10
1270 DATA 32,23,62,250,6,1,247,7
1280 DATA 32,15,62,250,247,3,32,9
1290 DATA 14,0,201,62,253,24,2,62
1300 DATA 239,245,62,250,247,3,241,14
1310 DATA 0,201,197,213,120,4,33,23
1320 DATA 193,190,32,6,19,35,26,16
1330 DATA 248,235,209,193,201,5,0,0
1340 DATA 0,0,0,0,0,0,0,0
1350 DATA 0,0
1360 !
1370 DEF ERZEUGE(A#)
1380 OPEN #3:A# ACCESS OUTPUT
1390 RESTORE
1400 FOR N=1 TO 266
1410 READ A
1420 PRINT #3:CHR$(A);
1430 NEXT N
1440 CLOSE #3
1450 END DEF

```

SOUND-Einführung

Normalerweise ist der ENTERPRISE nicht nur für seinen Graphik-, sondern auch für seinen hervorragenden Sound-Chip bekannt. Doch wenn es einmal an die Programmierung des "Dave" geht, schrecken die meisten zurück. Ich muß zugeben, auch mir liegt dieses Kapitel nicht besonders, wahrscheinlich deshalb, weil ich absolut unmusikalisch bin. Ich habe mich jetzt aber einmal hinter dieses Thema geklemmt und bin zusammen mit Hilfe eines Artikels der englischen IEUG auf einige interessante Ergebnisse gestoßen.

Zuallererst möchte ich die wichtigsten Soundbefehle des ENTERPRISE näher erläutern (obwohl

dies im Handbuch bereits der Fall ist): Um zu vernünftigen Ergebnissen zu kommen, werden meist die Befehle PITCH (für Tonhöhe), DURATION (für Länge des Tones), SOURCE (für die Nummer des zu verwendenden Tongenerators 0-3), STYLE (für Verzerrung/Filter), ENVELOPE (für die Verwendung vorher definierter Hüllkurven), SYNC (für die Synchronisation mit mehreren anderen Tönen), INTERRUPT (für Unterbrechen eines alten Tones), LEFT (=links) und RIGHT (=rechts) in Kombination mit dem Befehl SOUND angewandt.

"Dave" verwendet vier Soundkanäle die verschieden genutzt werden können. Die ersten (0 bis 2) sind dabei für "klare Töne" gedacht, der vierte (3) für Rauschen. Diese Kanäle sind die oben bereits kurz erwähnten SOURCES (übersetzt: Quellen).

Verschiedene Filter und Verzerrungsarten wird der Befehl SOUND STYLE verwendet. Er legt genau fest, wie der Ton verändert werden soll:

SOUND STYLE 16 - Schwache Verzerrung. Unter bestimmten Umständen kann sich der Style 16 wie eine verzerrte elektrische Gitarre anhören.

SOUND STYLE 32 - Mittlere Verzerrung. Eine mittlere "Einigung" zwischen dem 16. und dem 48. Grad.

SOUND STYLE 48 - Hohe Verzerrung. Macht teilweise unerträglichen Krach, kann jedoch sehr gut angewandt werden für Geräusche wie z.B. Donner.

SOUND STYLE 64 - High-Pass-Filter wird verwendet. Er wird meist in Kombination der beiden Kanäle 1 und 2 angewandt.

SOUND STYLE 128 - Ring-Modulation durch Kombination der Kanäle 0 und 2, die zusammen diesen "metallischen" Sound erzeugen.

SOUND STYLE 144 - Eine Vereinigung der SOUND STYLES 128 und 16 bzw. 160 und 176.

SOUND STYLE 192 - Eine Kombination des High-Pass-Filters und der Ring-Modulation.

SOUND STYLE 255 - Die höchstmögliche Zahl für den Befehl Style. Vielleicht eine Vereinigung aller möglichen Arten?

Am Besten für die Erforschung des SOUND STYLE-Befehles ist immer noch das Ausprobieren. Denn ich kann die genauen Funktionen nicht so genau beschreiben, wie man es eigentlich sollte. Apropos: ich wäre dankbar, wenn ein Mitglied mit etwas mehr Sound-Erfahrung hier vielleicht noch einen Nachtrag schreibt? Adresse vorn!

Die folgenden Spielereien dienen dem Versuchen des bisher gelesenen. Tippen Sie sie ab und starten sie dann mit RUN.

```
100 DO
110   FOR C=1 TO 3
120     FOR A=125 TO 1 STEP-1
130       SOUND PITCH A,DURATION
140         NEXT A
150       FOR B=125 TO 1 STEP-2
160         SOUND PITCH B,DURATION
170           IF C=3 THEN STOP
180         NEXT B
190       NEXT C
200 LOOP UNTIL C=3
```

Die Befehle DO und LOOP, FOR und NEXT wurden bereits in den Basic-Kursen ausführlich behandelt, deshalb hier keine erneute Erklärung. Die folgende Soundprobe soll eine 'Heavy-Metal'-Gitarre andeuten, ob sie ganz gelungen ist, bleibt offen ...

```
100 DO
110   FOR C=1 TO 3
120     FOR A=18 TO 46 STEP 2.4
130       FOR B=10 TO 20 STEP 2
140         SOUND PITCH A,DURATION B
150           ,STYLE 192,SOURCE 0
160           SOUND PITCH A-9,DURATION
170             B,STYLE 64,SOURCE 1
180           SOUND PITCH A+12,DURATIO
190             N B,STYLE 16,SOURCE 2
200         IF C=3 THEN STOP
210       NEXT B
220     NEXT A
230   NEXT C
240 LOOP UNTIL C=3
```

Selbstverständlich kann das STOP der beiden Listings mit anderen Befehlen ersetzt werden (z.B. GOTO in größeren Programmen), hier dient es nur der korrekten Durchführung des Programmes.

Zu STYLE läßt sich noch hinzufügen, daß mehrere Werte miteinander addiert zu neuen Variationen führen.

Ein großer Teil der Überlegenheit des ENTERPRISE-Sounds gegenüber anderen Computern ist die Auswahl der Hüllkurven mit ENVELOPE. Hier kann der Anwender genau festlegen, wie seine Hüllkurve für auch nur einen SOUND-Befehl aussehen soll. Der ENVELOPE Befehl hat am Anfang die Nummer der gewünschten Hüllkurve 'ENVELOPE NUMBER x', dann folgt ein Semikolon und viermal eine Palette von jeweils vier Werten, wieder durch ein Semikolon getrennt:

```
100 ENVELOPE NUMBER 1;2,0,40,30;
5,20,63,20;-63,63,10;6,63,-63,50
```

Man kann den ENVELOPE-Befehl folgendermaßen aufteilen: der erste Viererblock dient dem Festlegen des Tonwertes (Hoch oder Tief), der zweite und dritte Viererblock der Links- und Rechts-Ansteuerung, der vierte Teil dient der Länge (50 Einheiten entsprechen einer Sekunde). Um die oben aufgeführte Hüllkurve zu testen, können Sie folgende Zeilen hinzufügen:

```
110 SOUND PITCH 40,DURATION 150,
STYLE 48,ENVELOPE 1
120 SOUND PITCH 45,DURATION 200,
STYLE 48,ENVELOPE 1,SOURCE 1
```

Dieses Rauschen vom linken zum rechten Kanal hat eine Länge von 110 Einheiten, also etwas mehr als zwei Sekunden.

Die Möglichkeiten der SOUND-Gestaltung bleiben dabei letztendlich Ihrer Fantasie überlassen.

Hier noch ein Beispiel:


```

100 ENVELOPE NUMBER 1;-12,63,20,
5;2,40,10,4;3,30,63,8
110 FOR A=57 TO 58 STEP .2
120 FOR B=80 TO 40 STEP-3
130 SOUND PITCH A+B*B,DURATION
9,STYLE 128
140 SOUND PITCH A/B/B,DURATION
9,SOURCE 2,STYLE 128,ENVEL
LOPE 1
150 NEXT
160 NEXT

```

Den Unterschied der SOUND-Ausgabe werden Sie erkennen, wenn Sie dieses Listing ohne bzw. mit Zeile 100 und dem Zusatz ENVELOPE 1 in Zeile 140 starten.

Die letzte Soundoption, die ich hier erklären möchte, ist das Arbeiten mit dem Soundkanal (#103), den Sie gleich behandeln, wie in den bisherigen Basic-Kursen ausführlich erläutert war. Versucht Doch einfach mal Eure Möglichkeiten in diesem Spektrum! Wenn Ihr gute Ergebnisse herausbekommt, drucken wir sie gerne hier ab.

(S.E.)

Tips & Tricks

Graphik-UTILITY

Dieses Utility ist eigentlich nur für ENTERPRISE 64k-Anwender von Nutzen, da sie hiermit die Möglichkeit erhalten, wie die Anwender des 128k-Computers Bilder abzuspeichern und zu laden. Wenn Sie das Listing abtippen und mit RUN starten, wird selbständig das eigentliche Programm auf Diskette oder Cassette (diese muß natürlich vorher zum Speichern bereitgemacht werden) abgelegt.

Zur Sicherheit sollten Sie jedoch das Basicprogramm auch mit SAVE "UTILITY.BAS" abspeichern, falls sich ein Tippfehler eingeschlichen hat. Das endgültige Programm wird dann später mit :LOAD UTILITY oder LOAD "UTILITY" geladen. In der Auflistung mit :HELP ist das speicherresidente Programm ebenfalls aufgeführt.

Dieses Utility war unter dem Namen UTIL bereits auf einigen PD-Disketten der EUG und ENTERPRISE COMPUTERS enthalten, war jedoch damit für die Cassettenbenutzer nicht zugänglich.

```

1000 CALL ERZEUGE("utility")
1010 !
1020 DATA 0,7,244,4,255,255,0,0
1030 DATA 0,0,0,0,0,0,0,0
1040 DATA 60,191,128,226,128,123,248,10
1050 DATA 202,150,192,3,214,64,245,149
1060 DATA 51,128,7,162,128,75,36,28
1070 DATA 0,3,0,66,36,0,3,36
1080 DATA 240,183,31,8,5,0,134,108
1090 DATA 85,0,96,15,136,0,224,3
1100 DATA 100,158,6,0,4,96,72,0
1110 DATA 249,254,247,4,43,219,44,146
1120 DATA 177,76,130,86,34,136,12,38
1130 DATA 227,32,128,172,76,39,144,72
1140 DATA 130,3,153,140,228,101,50,155
1150 DATA 132,7,83,161,164,216,105,58
1160 DATA 30,65,160,162,153,132,236,101
1170 DATA 57,136,12,38,227,32,128,216
1180 DATA 111,48,153,14,98,3,177,164
1190 DATA 200,101,55,136,12,102,131,9
1200 DATA 184,220,101,54,28,193,160,167
1210 DATA 232,134,29,2,49,90,164,131
1220 DATA 192,16,67,12,0,23,194,0
1230 DATA 96,76,70,26,8,62,31,172
1240 DATA 176,3,25,94,24,10,180,90
1250 DATA 164,19,192,16,67,7,192,23
1260 DATA 194,0,80,76,70,26,8,62
1270 DATA 31,172,176,2,25,94,209,96
1280 DATA 178,64,197,98,73,16,138,79
1290 DATA 29,1,74,197,50,9,88,138
1300 DATA 5,43,19,9,228,18,33,225
1310 DATA 252,166,96,54,214,100,112,112
1320 DATA 0,35,0,192,25,33,2,73
1330 DATA 184,236,97,54,26,76,130,3
1340 DATA 49,164,216,101,55,24,77,166
1350 DATA 86,108,155,255,96,63,70,192
1360 DATA 32,3,54,98,128,48,165,56
1370 DATA 3,245,150,2,35,8,2,207
1380 DATA 211,240,28,12,2,123,130,216
1390 DATA 82,92,1,250,114,1,191,78
1400 DATA 96,71,233,192,10,253,56,129
1410 DATA 159,172,176,1,24,80,8,102
1420 DATA 201,128,22,20,127,0,12,1
1430 DATA 153,179,84,5,133,29,192,4
1440 DATA 32,0,3,244,188,6,42,63
1450 DATA 72,192,80,200,65,250,253,58
1460 DATA 131,31,167,64,107,244,252,7
1470 DATA 3,0,222,224,177,1,101,250
1480 DATA 113,4,63,78,0,151,233,204
1490 DATA 20,253,57,2,223,166,224,99
1500 DATA 244,204,13,126,178,192,100,97
1510 DATA 64,57,250,203,1,151,132,0
1520 DATA 34,201,64,154,68,20,138,69
1530 DATA 32,144,203,244,234,14,126,157
1540 DATA 1,239,214,88,12,140,32,20
1550 DATA 191,79,192,96,67,54,61,129
1560 DATA 8,3,160,129,232,129,236,205
1570 DATA 130,64,66,0,163,245,150,2
1580 DATA 39,10,1,207,211,240,28,28
1590 DATA 1,15,0,95,165,96,48,24
1600 DATA 2,247,5,131,128,12,151,233
1610 DATA 152,26,253,55,3,8,132,209
1620 DATA 72,164,82,9,114,179,101,96

```


1630 DATA 19,5,155,48,128,132,14,112
1640 DATA 195,48,52,214,2,145,197,1
1650 DATA 97,8,24,0,25,15,25,79
1660 DATA 207,225,24,160,30,254,17,10
1670 DATA 11,79,225,0,128,216,35,8
1680 DATA 60,7,207,5,187,36,60,0
1690 DATA 2,138,30,226,51,243,248,64
1700 DATA 40,123,159,159,195,161,64,237
1710 DATA 252,32,20,10,90,195,1,195
1720 DATA 145,252,10,24,56,10,231,180
1730 DATA 56,225,182,135,28,54,16,99
1740 DATA 134,170,28,112,210,65,14,25
1750 DATA 229,241,24,65,180,24,100,136
1760 DATA 207,207,225,208,160,36,254,16
1770 DATA 8,23,225,7,160,97,118,35
1780 DATA 63,63,132,66,128,67,248,64
1790 DATA 32,13,4,30,129,131,187,54
1800 DATA 61,0,4,192,162,131,128,105
1810 DATA 252,34,20,10,14,225,49,27
1820 DATA 244,104,32,8,60,195,6,22
1830 DATA 108,100,0,13,63,132,2,128
1840 DATA 179,248,94,40,17,191,129,34
1850 DATA 128,115,248,68,40,38,157,194
1860 DATA 98,55,232,208,64,16,115,134
1870 DATA 1,224,81,64,48,38,26,127
1880 DATA 11,197,0,135,240,128,64,54
1890 DATA 12,60,85,236,144,152,21,149
1900 DATA 25,127,195,79,227,8,224,12
1910 DATA 254,61,140,0,77,97,3,248
1920 DATA 158,40,5,191,136,130,128,163
1930 DATA 248,64,40,114,48,226,95,251
1940 DATA 245,150,0,35,48,162,39,251
1950 DATA 245,150,2,99,6,24,111,214
1960 DATA 88,9,156,24,94,143,148,203
1970 DATA 118,75,244,230,7,114,191,92
1980 DATA 173,17,8,132,0,25,104,191
1990 DATA 70,194,0,0,3,37,250,229,112
2000 DATA 132,68,0,0,203,173,155,7
2010 DATA 0,88,15,211,168,4,250,247
2020 DATA 1,48,31,174,87,8,68,32
2030 DATA 0,12,185,70,192,1,24,216
2040 DATA 22,35,126,159,128,103,113,27
2050 DATA 244,252,4,59,136,223,167,224
2060 DATA 41,220,70,253,63,1,142,224
2070 DATA 96,80,140,108,0,8,62,218
2080 DATA 32,16,128,1,250,126,0,189
2090 DATA 193,12,151,235,149,194,17,16
2100 DATA 0,3,46,182,108,192,0,96
2110 DATA 63,78,160,19,235,220,2,192
2120 DATA 126,185,92,33,16,128,0,50
2130 DATA 229,117,128,66,0,7,233,248
2140 DATA 2,247,3,56,88,7,229,184
2150 DATA 128,218,35,63,63,129,98,3
2160 DATA 56,141,250,203,1,17,133,4
2170 DATA 55,233,248,14,247,1,149,159
2180 DATA 167,32,24,4,44,1,123,132
2190 DATA 4,101,103,233,200,8,1,11
2200 DATA 128,94,225,1,25,89,250,114
2210 DATA 2,191,79,192,49,232,160,6
2220 DATA 61,16,0,153,99,160,8,96
2230 DATA 2,247,8,8,202,207,211,144
2240 DATA 24,2,25,0,189,194,15,211
2250 DATA 240,28,35,19,255,94,224,22

2260 DATA 20,46,254,100,191,79,192,53
2270 DATA 240,128,54,35,126,159,128,139
2280 DATA 225,0,80,70,253,63,1,87
2290 DATA 194,0,104,141,250,126,3,47
2300 DATA 132,0,101,123,36,124,238,91
2310 DATA 178,71,205,181,187,37,250,203
2320 DATA 0,17,133,1,231,235,149,194
2330 DATA 17,8,0,3,46,81,176,0
2340 DATA 70,54,5,1,129,194,49,176
2350 DATA 0,32,251,104,128,66,0,7
2360 DATA 233,248,2,247,4,48,31,167
2370 DATA 224,11,220,6,201,23,0,26
2380 DATA 167,209,96,160,84,247,0,180
2390 DATA 69,1,215,243,228,80,242,122
2400 DATA 139,79,175,112,27,197,146,46
2410 DATA 0,53,79,162,193,64,77,238
2420 DATA 2,104,138,0,207,231,200,161
2430 DATA 228,24,115,181,69,167,215,184
2440 DATA 13,162,175,100,180,71,207,149
2450 DATA 187,37,250,126,1,173,197,2
2460 DATA 70,108,92,0,117,191,77,192
2470 DATA 81,48,0,154,68,20,138,69
2480 DATA 32,146,105,17,214,197,102,198
2490 DATA 48,6,11,1,250,203,1,151
2500 DATA 132,0,183,235,44,6,70,16
2510 DATA 10,95,164,224,40,24,0,253
2520 DATA 55,3,159,166,96,123,54,68
2530 DATA 128,23,132,101,103,235,44,6
2540 DATA 94,20,1,204,70,177,72,164
2550 DATA 82,25,117,191,77,192,167,233
2560 DATA 152,22,25,102,194,208,6,3
2570 DATA 244,220,14,126,153,129,226,49
2580 DATA 27,244,234,14,126,157,1,234
2590 DATA 246,75,244,220,14,126,153,129
2600 DATA 236,216,170,0,94,17,149,141
2610 DATA 102,33,72,164,82,25,126,147
2620 DATA 129,15,210,48,36,18,201,102
2630 DATA 196,16,7,88,132,0,128,91
2640 DATA 187,74,78,210,16,192,74,9
2650 DATA 98,147,72,142,21,187,180,132
2660 DATA 229,102,193,176,6,11,1,182
2670 DATA 177,30,52,214,33,16,1,0
2680 DATA 66,0,32,45,221,164,33,128
2690 DATA 28,192,105,0,128,8,1,135
2700 DATA 3,244,252,1,126,178,192,4
2710 DATA 97,64,13,238,8,100,189,192
2720 DATA 204,147,224,28,14,246,126,52
2730 DATA 214,44,181,24,45,232,201,180
2740 DATA 0,12,0,10,0,0,0,0
2750 DATA 0,0,0,0,0,0,0,0
2760 DATA 0,0,0,0,0,0,0,0
2770 DATA 0,0,0,0,0,0,0,0
2780 !
2790 DEF ERZEUGE(A#)
2800 OPEN #3:A# ACCESS OUTPUT
2810 RESTORE
2820 FOR N=1 TO 108
2830 READ A
2840 PRINT #3:CHR#(A);
2850 NEXT N
2860 CLOSE #3
2870 END DEF

Hardware-Basteltip:

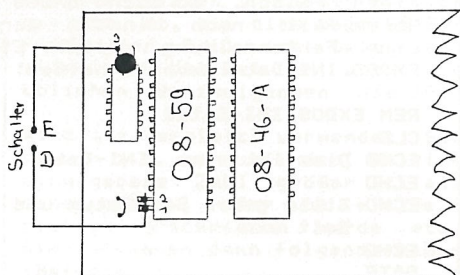
Viele Programme für den ENTERPRISE verlangen ein englisches Basic-Modul. (unter anderem auch der INTERLACE-Modus und viele Spiele). Die meisten in Deutschland verwendeten Rechner haben aber ein deutsches Basic-Modul.

Dies ist zwar nicht grundsätzlich schlecht, jedoch sollte zumindest ein Umschalten zwischen den Modi möglich sein. Der Club-Service im Heft 5/87 bot dies für DM 10,-- an, die Fa. ENTERPRISE hat den Umbau auch in ihrem Angebot.

Der Umbau des Moduls ist jedoch sehr einfach und erfordert außer etwas Kenntnis im Umgang mit einem Lötkolben keine besonderen Fähigkeiten. Für den Umbau benötigen wir folgendes:

- etwas Lötzinn
- einige cm zweiadriges Kabel
- Um- bzw. Ein/Aus-Schalter

Wenn Sie das Basic-Modul öffnen, finden Sie die Platine vor. Nun müssen Sie Pin 1 und Pin 2 (siehe Skizze) hochbiegen, so daß kein Kontakt mehr zur Platine besteht. Diese Überbrücken Sie mit dem einen Kabel und verbinden es mit dem Schalter (D=Deutsch). Nun wird an Lötstelle 3 der Skizze das andere Kabel angebracht und mit dem Schalter (E=Englisch) verbunden. Das ist alles, zu beachten ist nur, daß nach dem Umschalten ein Reset ausgeführt werden muß.



Für evtl. Schäden an Ihrem Basic-Modul können wir selbstverständlich keine Haftung übernehmen.

Einführung in EXDOS Teil 2

Im letzten regulären Info wurde von der EXDOS-Einführung der 1. Teil abgedruckt. Dieser enthielt unter anderem die Befehle HELP, EXDOS, ISDOS, DIR, FORMAT, VOL, CLS, ERA, DEL, LOAD, MKDIR (MD), RMDIR (RM), RNDIR, REN(AME). In dem 2. Teil möchten wir die anderen wichtigsten Befehle von EXDOS behandeln.

Der Befehl :COPY kopiert ein oder mehrere Files auf ein Gerät oder einen Kanal. Sie können mit COPY zum Beispiel Programme auf die RAMDISK (siehe unten !) herüberziehen:

```
:RAMDISK 5  
:COPY A:*.BAS E:
```

Voraussetzung ist natürlich, daß die RAMDISK genügend Speicherplatz für die gewünschten Dateien hat. Das obige Beispiel kopiert alle Dateien mit dem Zusatz .BAS auf die RAMDISK. Hier können Sie nun Ihre eigenen Varianten versuchen, z.B. EUG*.*, *.* , RACE.* usw. Der COPY-Befehl läßt nun noch verschiedene Zusatzoptionen zu, z.B. /A für Ascii-Kontrolle, /H für "Hidden (= versteckte)" Files. Um das aktuelle Datum und Zeit umzuändern, kann die /T-Option verwendet werden. Denn normalerweise wird Datum und Zeit von der Quelldatei mitkopiert. In bestimmten Fällen wünscht der Anwender jedoch die Aktualisierung von Datum und Zeit.

Das Datum kann auch manuell verändert werden mit :DATE TT-MM-JJ (Tag-Monat-Jahr). Ebenso die Zeit mit :TIME HH:MM:SS (Stunde:Minute: Sekunde).

Der Befehl :ECHO druckt einen gewünschten Text, der angegeben wird, auf den Bildschirm. Dies geschieht nicht nur in EXDOS-Ebene, d.h., Sie können ECHO auch in Basic verwenden. Bsp. :ECHO Dies ist ein Text ...

:MOVE kann verwendet werden, um Dateien von einem Unterdirectory (= Unterverzeichnis) in ein anderes zu verschieben. Hier kann je-

doch auch COPY verwendet werden, wenn man bestimmte Zusätze anbringt. Bsp. :MOVE E:*.TXT E:!
TEXTE

:MVDIR funktioniert im Prinzip wie MOVE, nur daß hier ganze Unterdirectories verschoben werden. Bsp. :MVDIR TEXTE BASIC.

:PAUSE im einfachen Modus fordert den Anwender mit Hinweis auf, die ENTER-Taste zu drücken ("Weiter mit ENTER ..."). Hier können auch Zusätze angebracht werden, wie z. B. :PAUSE Für Sprung in Basic-, was auf dem Bildschirm aussähe:

Für Sprung in Basic-
Weiter mit ENTER ...

Die oben bereits erwähnte RAMDISK ist im Prinzip wie eine Diskette, die im Speicher des ENTERPRISE reserviert wird. Diese läßt bei dem 128k-Modell max. 6 Einheiten zu (1 Einheit entspricht 16k, wobei insgesamt unabhängig von der Anzahl ca. 4k für das Directory abgezogen werden). 6 entspricht also dann $6 \cdot 16 = 96k$. Angesprochen wird sie mit :RAMDISK Parameter (hier: 6). Ist bereits eine RAMDISK vorhanden, so kann diese mit /D (= Delete/Löschen) gelöscht werden. Möglich ist auch das Verknüpfen dieser Optionen, :RAMDISK 6/D baut eine neue RAMDISK auf, die eine evtl. vorher vorhandene löscht. Zu beachten ist beim Parameter 6, daß hier keine anderen Module (PR, SUPERWP...) wegen des Speicherplatzes geladen sein dürfen.

Um in eine EXDOS.INI-Datei einen Kommentar einzubauen, der nicht auf dem Bildschirm erscheinen soll, kann :REM verwendet werden. Hier können nahezu alle Zeichen angegeben werden, Bsp. :REM Dies ist eine Initialisierungsdatei.

Der Befehl :TYPE dient dem Ansehen oder Ausdrucken einer Datei. Hier ist es gleichgültig, welches Format diese Datei hat (Text, Ascii-File, Basic-Listing usw.). Unter Basic können Sie den Standardkanal auf #104 setzen (PRINTER:), wenn sie dann eine Datei TYPEN, so wird diese nur auf dem

Drucker ausgedruckt. Hier können jedoch verschiedene Steuerzeichen auf dem Drucker verschiedene Auswirkungen haben (FormFeed...). Unter IS-Dos kann mit A>PRINTER der Drucker ein- bzw. ausgeschaltet werden (wird in Statuszeile angezeigt), beim TYPEN einer Datei wird diese dann auf Bildschirm und Drucker ausgegeben. Bei TYPE sind - im Gegensatz zu den meisten anderen Systemen - auch Parameter zugelassen, zum Beispiel :TYPE *.TXT usw.

Mit :VAR kann eine Systemvariable neu gesetzt oder umgeändert werden. Diese sind in Ihrem Handbuch genau erläutert, deshalb sprechen wir sie hier nicht an. Der Befehl kann heißen :VAR Zahl, wobei der eingestellte Wert der Systemvariable gezeigt wird, oder mit :VAR 79 ON bzw. OFF, wobei die Variable ein- bzw. ausgeschaltet wird.

Eine EXDOS.INI-Datei wird im Normalfall dazu verwendet, beim Neustart des Systems sofort etwas auszuführen, z.B. Datum und Zeit zu setzen, ein Modul zu laden, usw. Hier erfahren Sie, wie Sie leicht eine solche Initialisierung schreiben können.

Das einfachste Mittel für das Schreiben der Datei ist der Word Processor, in den mit :WP gesprungen wird. Sie sollten den 40 Zeichen-Modus verwenden, da EXDOS auch nur im 40-Zeichen-Modus läuft. In der EXDOS.INI-Datei dürfen alle oben erklärten Befehle stehen. Schreiben Sie diese falsch, so macht EXDOS im Normalfall nach Hinweis durch eine Fehlermeldung weiter. Eine EXDOS.INI-Datei könnte heißen:

```
REM EXDOS.INI-Datei
CLS
ECHO Dies ist eine .INI-Datei
ECHO
ECHO Bitte geben Sie Datum und
      Zeit ein -

ECHO
DATE
ECHO
TIME
ECHO
LOAD UTIL
ECHO Grafikutility geladen,
```


PAUSE Für Sprung in Basic-BASIC

Versuchen Sie doch einfach einmal Ihre eigenen Varianten!

S.E.

Achtung Gewerbe-Anbieter!

Möchten Sie in der EUG Info Hard- oder Software anbieten oder bewerben? Fordern Sie unsere Anzeigen-Preisliste an, es lohnt sich!

Basic-Kurs Teil 6

Da die letzte reguläre Info der EUG keinen Basic-Kurs enthielt, wird in diesem Heft der Kurs etwas ausführlicher behandelt werden. Wir widmen uns in diesem Teil den Entscheidungen und Verzweigungen, die der ENTERPRISE anbietet. Um diese richtig in Ihre Programme einbauen zu können, sollten Sie schon Bescheid wissen, da das Handbuch diesen Teil leider etwas vernachlässigt. Wir weisen noch einmal darauf hin, daß wir bei dem geringen Platz auf keinen Fall eine gesamte Programmiersprache erklären können. Hierfür sollten Sie sich wirklich an Fachbücher (vielleicht bringt die EUG eines heraus?) und Handbücher halten.

Die üblichen Basic-Versionen der Computer bieten in diesem Bereich weniger an, als dies beim ENTERPRISE der Fall ist. Selbstverständlich besitzt er auch die üblichen Verzweigungen wie IF:

Wird als Vergleich verwendet. Sie können z.B. herausfinden, ob eine eingegebene Zahl größer ist als eine andere, oder ob ein Text richtig geschrieben wurde etc. Sie verfahren dann folgendermaßen (Beispiel):

```
10 INPUT PROMPT "1. Zahl: ":A
20 INPUT PROMPT "2. Zahl: ":B
30 IF A>B THEN PRINT "Die 1. Zahl
   war größer als die 2."
```

```
40 IF A<B THEN PRINT "Die 1. Zahl
   war kleiner als die 2."
50 PRINT "Beide Zahlen waren gleich
   groß."
60 PRINT
70 GOTO 10
```

Der ENTERPRISE verwendet dabei, wie üblich, die normalen mathematischen Vergleichsoperatoren wie > für größer, < für kleiner, = für gleich, <> für ungleich, >= für größer oder gleich, <= für kleiner oder gleich.

Sie können nun noch für mehrere Anweisungen, die nach einem Zutreffen nach IF folgen sollen, die END IF-Anweisung benutzen. Die IF-Zeile sieht bis zum THEN gleich aus, wie oben, dann folgt der Rest in der nächsten Zeile. Die ELSE-Anweisung steht für 'ansonsten'. Trifft nämlich eine in der IF-Zeile abgefragte Bedingung nicht zu, so kann dies mit ELSE abgefangen werden. Bsp.:

```
10 INPUT PROMPT "Name ? ":N$
20 IF N$="OTTO" THEN PRINT "Sie heißen OTTO." ELSE PRINT "Sie heißen anders als OTTO."
```

Oder:

```
10 INPUT PROMPT "Name ? ":N$
20 IF N$="OTTO" THEN
30 PRINT "Sie heißen OTTO."
40 ELSE
50 PRINT "Sie heißen anders als OTTO."
60 END IF
```

Für Menu-Auswahlen ist der ON-Befehl sehr nützlich. ON braucht eine Zahl, dann folgt der Befehl GOTO X,Y,Z usw. Bsp.:

```
10 PRINT "1) Ende"
20 PRINT "2) Neustart"
30 PRINT "3) Basic"
40 PRINT
50 INPUT W
60 ON W GOTO 70,80,90
65 GOTO 50
70 STOP
80 RUN
90 EXT "BASIC"
```

Sie werden sehen, daß ON außer den möglichen Zahlen (hier 1-3) keine andere Wahl zuläßt.

Die SELECT CASE Option sieht schon etwas mehr nach einer anderen Hochsprache aus: Sie läßt ebenfalls mehrere Reaktionen, die frei definiert werden können, zu.

```
10 INPUT PROMPT "Zahl zw. 1-3 ":A
20 SELECT CASE A
30 CASE 1
40 PRINT "1 eingegeben"
50 CASE 2
60 PRINT "2 eingegeben"
70 CASE 3
80 PRINT "3 eingegeben"
90 CASE ELSE
100 PRINT "Nicht zw. 1-3"
110 END SELECT
```

Die GOSUB/RETURN-Verzweigung ist für Unterprogramme sehr nützlich: Sie möchten einen Programmteil nicht im Hauptprogramm stehen haben, sondern etwas abstrakt, was der Übersichtlichkeit und dem Durchlaufen des Programmes dient. Bsp.:

```
10 GOSUB 50
20 END
50 REM Initialisierung
60 LET A=100:LET B=A
70 PRINT "Initialisiert."
80 RETURN
```

Wichtig ist hier, daß das Unterprogramm immer mit RETURN abgeschlossen sein muß und daß der Rechner das Unterprogramm nicht ungewollt ein zweitesmal durchläuft (Dafür das END). Denn dann wäre das Unterprogramm nicht mit GOSUB angesprungen worden, sondern wäre eine logische Folge des Hauptprogramms. Für diesen Fall gibt der ENTERPRISE die Fehlermeldung 'Return ohne Gosub' zurück.

Mit diesem Heft schließen wir den Basic-Kurs ab, wenn Sie noch Fragen haben, können Sie uns wie immer gerne schreiben.

S.E.

Das folgende Listing ist die Computer Version des berühmten Spieles MASTER MIND, dessen bis zu zwei Spieler die vom ENTERPRISE ausgedachten Farben erraten müssen. Abspeichern mit SAVE. Von Robin Bowes

```
100 PROGRAM "Code-Quest.bas"(P#)
110 !
120 ! (C) Robin Bowes 1985
130 !
200 SET INTERRUPT STOP OFF
210 SET BORDER 0
220 SET KEY CLICK OFF
230 SET STATUS OFF
240 CLEAR FONT
250 CLOSE #102
260 WHEN EXCEPTION USE INSTRUCTIONS
270 STRING COL$,CODE$,WON$,CHOICE$
280 NUMERIC ROW,XX,YY,PLAY,SCORE(2
),TOTAL_SCORE(2),PLAYER_NO.,GAME_NO.
290 IF P#="" THEN CALL TITLE_PAGE
300 CALL SET_UP
310 CALL START
320 SELECT PLAY
330 CASE 1
340 LET PLAYER_NO.=1
350 DO
360 LET GAME_NO.=GAME_NO.+1
370 CALL SET_UP2
380 CALL DISPLAY
390 DO
400 CALL GAME
410 IF ROW=13 THEN EXIT DO
420 CALL NEXT
430 CALL SCORING
440 LOOP UNTIL WON#="win" OR R
OW=12
450 IF WON#="win" THEN
460 CALL WINNER
470 ELSE
480 IF ROW=12 OR ROW=13 THEN
CALL FAIL
490 END IF
500 CALL TOTAL_SCORES_ETC.
510 LOOP WHILE CHOICE#="continue"
"
520 CASE 2
530 DO
540 LET GAME_NO.=GAME_NO.+1
550 FOR PLAYER_NO.=1 TO PLAY
560 CALL SET_UP2
570 CALL DISPLAY
580 DO
590 CALL GAME
600 IF ROW=13 THEN EXIT DO
610 CALL NEXT
620 CALL SCORING
630 LOOP UNTIL WON#="win" OR
ROW=12
640 IF WON#="win" THEN
650 CALL WINNER
660 ELSE
670 IF ROW=12 OR ROW=13 TH
EN CALL FAIL
680 END IF
690 NEXT PLAYER_NO.
700 CALL TOTAL_SCORES_ETC.
710 LOOP WHILE CHOICE#="continue"
"
```



```

720 END SELECT
730 END WHEN
740 IF CHOICE$="end" THEN
750 CLOSE #102
760 END
770 END IF
780 CHAIN 0("!!")
790 !
800 !
810 !
820 DEF DISPLAY
830 CLOSE #102
840 SET VIDEO COLOUR 0
850 SET VIDEO MODE 15
860 SET VIDEO X 30
870 SET VIDEO Y 26
880 OPEN #1:"video:"
890 SET VIDEO Y 2
900 OPEN #2:"video:"
910 DISPLAY #2:AT 26 FROM 1 TO 2
920 SET VIDEO X 12
930 SET VIDEO Y 2
940 OPEN #3:"video:"
950 DISPLAY #3:AT 1 FROM 1 TO 2
960 SET #3:PALETTE BLACK,RGB(1,.2,
.2)
970 SET #1:PALETTE BLACK,BLUE,GREE
N,CYAN,MAGENTA,YELLOW,RED,WHITE
980 SET #2:PALETTE BLACK,BLUE,GREE
N,CYAN,MAGENTA,YELLOW,RED,WHITE
990 PRINT #3:"Please wait"
1000 PRINT #3:" Player ";PLAYER_NO.
1010 SET #1:BIAS RED
1020 SET BORDER BLACK
1030 SET #1:INK 1
1040 RESTORE 1090
1050 FOR LNE=1 TO 8
1060 READ A1
1070 PLOT #1:A1,935;A1,0
1080 NEXT LNE
1090 DATA 1,4,256,260,680,684,886,8
90
1100 LET A=12
1110 FOR N=867 TO 10 STEP-72
1120 SET #1:INK 7
1130 FOR POS=1 TO YY
1140 PRINT #1:TAB(14-YY+POS*2)
CHR$(133);CHR$(134);
1150 NEXT POS
1160 PRINT #1:
1170 PRINT #1:TAB(4) A;
1180 FOR POS=1 TO YY
1190 PRINT #1:TAB(14-YY+POS*2)
CHR$(135);CHR$(136);
1200 NEXT POS
1210 PRINT #1:
1220 SET #1:INK 1
1230 PLOT #1:0,N;890,N
1240 LET A=A-1
1250 NEXT
1260 PLOT #1:1,935;890,935
1270 SET #2:INK 7
1280 PRINT #1,AT 1,1:

```

```

1290 FOR N=XX TO 1 STEP-1
1300 SET #1:INK N
1310 PRINT #1:TAB(29) CHR$(137) C
HR$(138);
1320 PRINT #1:TAB(29) CHR$(139);C
HR$(140);
1330 PRINT #1:TAB(29)
1340 NEXT XX
1350 DISPLAY #1:AT 2 FROM 1 TO 24
1360 CLEAR #3
1370 PRINT #3:" CODE "
1380 END DEF
1390 !
1400 !
1410 !
1420 DEF GAME
1430 SET INTERRUPT STOP ON
1440 LET C=1:LET ROW=ROW+1
1450 LET D=0:LET COL$="0000000"
1460 DO
1470 SET #1:INK 0
1480 PRINT #1,AT(XX*3+2)-C*3,29:C
HR$(137);CHR$(138);
1490 PRINT #1:TAB(29) CHR$(139);C
HR$(140);
1500 SET #2:INK C
1510 PRINT #2:TAB(16-YY+D) CHR$(1
33);CHR$(134)
1520 PRINT #2:TAB(16-YY+D) CHR$(1
35);CHR$(136)
1530 SET #1:INK C
1540 PRINT #1,AT(XX*3+2)-C*3,29:C
HR$(137);CHR$(138);
1550 PRINT #1:TAB(29) CHR$(139);C
HR$(140);
1560 SET #2:INK VAL (COL$(D/2+2:D/
2+2))
1570 PRINT #2:TAB(16-YY+D) CHR$(1
29);CHR$(130)
1580 PRINT #2:TAB(16-YY+D) CHR$(1
31);CHR$(132)
1590 GET A$
1600 IF A$="" THEN
1610 GOTO 1470
1620 ELSE IF ORD(A$)=176 THEN
1630 IF C<XX THEN LET C=C+1
1640 ELSE IF ORD(A$)=180 THEN
1650 IF C>1 THEN LET C=C-1
1660 ELSE IF ORD(A$)=188 THEN
1670 IF D<YY*2-2 THEN LET D=D+2
1680 ELSE IF ORD(A$)=184 THEN
1690 IF D>0 THEN LET D=D-2
1700 ELSE IF ORD(A$)=32 THEN
1710 LET COL$=COL$(1:D/2+1)&STR
$(C)&COL$(D/2+3:YY+1)
1720 END IF
1730 IF ORD(A$)=13 THEN
1740 LET CORRECT$="yes"
1750 FOR N=2 TO YY+1
1760 IF COL$(N:N)="0" THEN LE
T CORRECT$="no"
1770 NEXT

```

```

1780     IF CORRECT$="yes" THEN EXI
T DO
1790     END IF
1800     IF ORD(A#)=27 THEN
1810         LET ROW=13
1820         EXIT DO
1830     END IF
1840     LOOP
1850 END DEF
1860 !
1870 !
1880 !
1890 DEF NEXT
1900     FOR N=1 TO YY
1910         SET #1:INK VAL (COL$(N+1:N+1)
)
1920     PRINT #1,AT 25-ROW*2,14-YY+N
*2:CHR$(129);CHR$(130)
1930     PRINT #1:TAB(14-YY+N*2) CHR$(
(131);CHR$(132)
1940     SET #1:INK 1
1950     NEXT N
1960     PLOT #1:0,3+72*ROW;890,3+72*RO
W
1970     IF ROW=12 THEN PLOT #1:0,935;8
90,935
1980     PRINT #2:TAB(11) "      "
!      11 spaces
1990     PRINT #2:TAB(11) "      "
2000 END DEF
2010 !
2020 !
2030 !
2040 DEF SET_UP
2050     TEXT 40
2060     LET CHOICE$="continue"
2070     LET GAME_NO.=0
2080     LET TOTAL_SCORE(1)=0
2090     LET TOTAL_SCORE(2)=0
2100     SET CHARACTER 129,0,0,7,15,31,
63,127,127,127
2110     SET CHARACTER 130,0,0,224,240,
248,252,254,254,254
2120     SET CHARACTER 131,127,127,127,
63,31,15,7,0,0
2130     SET CHARACTER 132,254,254,254,
252,248,240,224,0,0
2140     SET CHARACTER 133,0,0,7,15,28,
56,112,96,96
2150     SET CHARACTER 134,0,0,224,240,
56,28,14,6,6
2160     SET CHARACTER 135,96,96,112,56
,28,15,7,0,0
2170     SET CHARACTER 136,6,6,14,28,56
,240,224,0,0
2180     SET CHARACTER 137,0,0,31,63,63
,63,63,63,63
2190     SET CHARACTER 138,0,0,248,252,
252,252,252,252,252
2200     SET CHARACTER 139,63,63,63,63,
63,63,31,0,0
2210     SET CHARACTER 140,252,252,252,
252,252,252,248,0,0

```

```

2220     SET CHARACTER 141,28,34,65,65,
65,34,28,0,0
2230     SET CHARACTER 143,28,62,127,12
7,127,62,28,0,0
2240     END DEF
2250 !
2260 !
2270 !
2280 DEF SCORING
2290     IF COL$=CODE$ THEN
2300         LET WON$="win"
2310         ENVELOPE NUMBER 0;50,63,63,Y
Y*6
2320         FOR N=1 TO 3
2330             SOUND ENVELOPE 0,DURATION
YY*6,PITCH 10
2340         NEXT N
2350     END IF
2360     LET MIGHT$=CODE$
2370     PRINT #1,AT 26-ROW*2,23;
2380     FOR PEG=2 TO YY+1
2390         IF COL$(PEG:PEG)=MIGHT$(PEG:
PEG) THEN
2400             SET #1:INK 7
2410             PRINT #1:CHR$(141);
2420             SET #1:INK 1
2430             PLOT #1:0,3+72*ROW;890,3+7
2*ROW
2440             LET COL$=COL$(1:PEG-1)&"a"
&COL$(PEG+1:YY+1)
2450             LET MIGHT$=MIGHT$(1:PEG-1)
&"b"&MIGHT$(PEG+1:YY+1)
2460         END IF
2470     NEXT PEG
2480     FOR PEG=2 TO YY+1
2490         FOR CODE=2 TO 6
2500             IF COL$(PEG:PEG)=MIGHT$(CO
DE:CODE) THEN
2510                 SET #1:INK 7
2520                 PRINT #1:CHR$(143);
2530                 SET #1:INK 1
2540                 PLOT #1:0,3+72*ROW;890,3
+72*ROW
2550                 LET MIGHT$=MIGHT$(1:CODE
-1)&"b"&MIGHT$(CODE+1:YY+1)
2560                 GOTO 2590
2570             END IF
2580         NEXT CODE
2590     NEXT PEG
2600 END DEF
2610 !
2620 !
2630 !
2640 DEF WINNER
2650     SET INTERRUPT STOP OFF
2660     SET #3:PALETTE BLACK,BLUE,GREE
N,CYAN,MAGENTA,YELLOW,RED,WHITE
2670     CLEAR #3
2680     CLEAR #2
2690     DISPLAY #1:AT 3 FROM 1 TO 24
2700     DISPLAY #3:AT 1 FROM 1 TO 2
2710     SOUND PITCH 20,DURATION 47*YY,
STYLE 33

```



```

2720 FOR N=1 TO YY
2730 SET #3:INK VAL(CODE$(N+1:N+1
))
2740 PRINT #3,AT 1,(5-YY)+N*2:CHR
$(129);CHR$(130);
2750 PRINT #3,AT 2,(5-YY)+N*2:CHR
$(131);CHR$(132)
2760 SET #1:INK 1
2770 FOR TIME=1 TO 400
2780 NEXT TIME
2790 NEXT N
2800 FOR TIME=1 TO 2000
2810 NEXT TIME
2820 CLOSE #1
2830 CLOSE #2
2840 CLOSE #3
2850 GRAPHICS HIRES 16
2860 SET INK 2
2870 PRINT #101,AT 5,7:"W E L L"
2880 SET INK 6
2890 PRINT #101,AT 7,5:"D O N E ! !
"
2900 LET SCORE(PLAYER_NO.)=13-ROW
2910 PRINT TAB(12) "Your score : "
;SCORE(PLAYER_NO.)
2920 RESTORE 3020
2930 ENVELOPE NUMBER 1;0,63,63,2;0,
-10,-10,6;0,-63,-63,35
2940 CLEAR SOUND
2950 FOR N=1 TO 31
2960 READ DRTN,PTCH
2970 SOUND PITCH PTCH+12,DURATION
DRTN*1.4,ENVELOPE 1,SYNC 1
2980 SOUND SOURCE 2,PITCH PTCH+5,
DURATION DRTN*1.4,ENVELOPE 1,SYNC 1
2990 NEXT N
3000 WAIT DELAY 4
3010 CLOSE #101
3020 DATA 15,37,10,36,5,36,6,36,4,3
4,5,36,14,37,11,36,5,36,10,34,5,34,6
3030 DATA 34,4,32,5,34,14,36,11,32,
5,34,10,36,5,36,6,36,4,34,5,36,10,37,6
3040 DATA 39,20,41,6,41,10,39,6,37,
10,36,6,34,18,32
3050 END DEF
3060 !
3070 !
3080 !
3090 DEF FAIL
3100 SET INTERRUPT STOP OFF
3110 SET #3:PALETTE BLACK,BLUE,GREE
N,CYAN,MAGENTA,YELLOW,RED,WHITE
3120 CLEAR #3:
3130 CLEAR #2:
3140 DISPLAY #1:AT 3 FROM 1 TO 24
3150 DISPLAY #3:AT 1 FROM 1 TO 2
3160 FOR N=1 TO YY
3170 SET #3:INK VAL(CODE$(N+1:N+1
))
3180 PRINT #3,AT 1,(5-YY)+N*2:CHR
$(129);CHR$(130);
3190 PRINT #3,AT 2,(5-YY)+N*2:CHR
$(131);CHR$(132)

```

```

3200 NEXT N
3210 SET #2:INK 6
3220 PRINT #2,AT 2,3:"Press any key
to continue."
3230 DO
3240 GET A$
3250 LOOP WHILE A$=""
3260 CLOSE #1
3270 CLOSE #2
3280 CLOSE #3
3290 GRAPHICS HIRES 16
3300 SET INK 1
3310 PRINT #101,AT 6,7:"F A I L"
3320 PRINT TAB(12) "Your score : 0
"
3330 LET SCORE(PLAYER_NO.)=0
3340 SOUND DURATION 10,PITCH 24
3350 FOR TIME=1 TO 170
3360 NEXT TIME
3370 SOUND DURATION 70,PITCH 19
3380 FOR TIME=1 TO 2000
3390 NEXT TIME
3400 CLOSE #101
3410 END DEF
3420 !
3430 !
3440 !
3450 DEF START
3460 SET #102:PALETTE BLACK,WHITE,B
LACK,CYAN
3470 PRINT #102,AT 10,5:"Do you wis
h instructions (Y/N)"
3480 DO
3490 GET A$
3500 LOOP UNTIL UCASE$(A$)="Y" OR U
CASE$(A$)="N"
3510 CLEAR #102
3520 IF UCASE$(A$)="Y" THEN CAUSE E
XCEPTION 1
3530 PRINT
3540 SET #102:INK 2
3550 PRINT AT 3,8:"G A M E C H O O
S I N G"
3560 PRINT AT 2,8:"-----
-----"
3570 PRINT AT 4,8:"-----
-----"
3580 SET #102:INK 1
3590 PRINT AT 7,5:"How many holes d
o you wish to play"
3600 PRINT AT 9,3:"with ? (1-5)"
3610 DO
3620 GET YY$
3630 LOOP UNTIL VAL(YY$)>0 AND VAL(
YY$)<6
3640 LET YY=VAL(YY$)
3650 PRINT AT 11,19:YY
3660 PRINT AT 14,5:"How many colour
s do you wish to"
3670 PRINT AT 16,3:"play with ? (1
-8)"
3680 DO
3690 GET XX$

```

```

3700 LOOP UNTIL VAL(XX$)>0 AND VAL(
XX$)<9
3710 LET XX=VAL(XX$)
3720 PRINT AT 18,19:XX
3730 FOR TIME=1 TO 200
3740 NEXT TIME
3750 CLEAR SCREEN
3760 LET PLAY=1
3770 PRINT
3780 SET #102:INK 2
3790 PRINT AT 2,8:"-----
-----"
3800 PRINT AT 3,8:"G A M E C H O O
S I N G"
3810 PRINT AT 4,8:"-----
-----"
3820 SET #102:INK 1
3830 PRINT AT 8,5:"How many players
want to play the"
3840 PRINT AT 10,3:"the game ? (1
or 2)"
3850 DO
3860 GET PLAY$
3870 LOOP UNTIL VAL(PLAY$)=1 OR VAL
(PLAY$)=2
3880 LET PLAY=VAL(PLAY$)
3890 PRINT AT 13,19:PLAY
3900 FOR TIME=1 TO 250
3910 NEXT TIME
3920 END DEF
3930 !
3940 !
3950 !
3960 HANDLER INSTRUCTIONS
3970 SET INTERRUPT STOP OFF
3980 IF EXTYPE=9229 THEN
3990 SET VIDEO MODE 5
4000 SET VIDEO X 3
4010 SET VIDEO Y 3
4020 OPEN #99:"video:"
4030 DISPLAY #99:AT 25 FROM 1 TO
3
4040 END IF
4050 TEXT
4060 PRINT
4070 SET #102:PALETTE BLACK,WHITE,B
LACK,CYAN
4080 SET #102:INK 2
4090 PRINT AT 2,9:"-----
-----" ! 23 minus signs
4100 PRINT AT 4,9:"-----
-----"
4110 PRINT AT 3,9:"I N S T R U C T
I O N S"
4120 SET #102:INK 1
4130 PRINT AT 7,3:"This game is a c
omputer version of"
4140 PRINT AT 9,1:"the popular game
Master Mind."
4150 PRINT AT 12,3:"The aim of the
game is to crack the"
4160 PRINT AT 14,1:"colour code sel
ected by the computer."

```

```

4170 PRINT AT 16,1:"Up to 5 holes c
an be filled with "
4180 PRINT AT 18,1:"any combination
of up to 8 colours."
4190 CALL ANY_KEY
4200 PRINT AT 6,2:"You select your
colours at the bottom"
4210 PRINT AT 8,12:"of the screen."
4220 SET #102:INK 2
4230 PRINT AT 11,16:"CONTROLS"
4240 SET #102:INK 1
4250 PRINT AT 14,2:"Colour cursor u
p.....JOYSTICK UP"
4260 PRINT AT 16,2:"Colour cursor d
own....JOYSTICK DOWN"
4270 PRINT AT 18,2:"Hole cursor rig
ht....JOYSTICK RIGHT"
4280 PRINT AT 20,2:"Hole cursor lef
t.....JOYSTICK LEFT"
4290 CALL ANY_KEY
4300 PRINT AT 7,5:"Should you, duri
ng the game, find "
4310 PRINT AT 9,3:"the flashing cur
sors distracting"
4320 PRINT AT 11,3:"press the HOLD
key, which freezes "
4330 PRINT AT 13,3:"the game allowi
ng concentrated "
4340 PRINT AT 15,3:"thinking. To re
start the game press"
4350 PRINT AT 17,3:"the HOLD key ag
ain."
4360 PRINT AT 19,5:"Choose the hole
position and the";
4370 PRINT AT 21,3:"colour then pre
ss the SPACE bar to";
4380 CALL ANY_KEY
4390 PRINT AT 7,3:"fill in the colo
ur."
4400 PRINT AT 9,5:"When all holes a
re filled to your"
4410 PRINT AT 11,3:"satisfaction, p
ress the ENTER key."
4420 PRINT AT 13,3:"The colours wil
l be filled in on"
4430 PRINT AT 15,3:"the main board
and the computer will";
4440 PRINT AT 17,3:"mark your colou
rs against it's";
4450 PRINT AT 19,3:"hidden code usi
ng black and white"
4460 PRINT AT 21,3:"dots, (&CHR$(1
41)&CHR$(143)&")."
4470 CALL ANY_KEY
4480 PRINT AT 7,5:"A black dot (&C
HR$(141)&) means that you "
4490 PRINT AT 9,3:"have chosen a co
rrect colour and"
4500 PRINT AT 11,3:"placed it in th
e correct position."
4510 PRINT AT 15,5:"A white dot (&
CHR$(143)&) means that you"
4520 PRINT AT 17,3:"have chosen a c

```



```

orrect colour but"
4530 PRINT AT 19,3:"placed it in th
e wrong position."
4540 SET #102:INK 2
4550 PRINT AT 23,8:"Press any key t
o continue"
4560 DO
4570 GET A$
4580 LOOP WHILE A$=""
4590 CLEAR SCREEN
4600 PRINT
4610 SET #102:INK 2
4620 PRINT AT 2,17:"-----"
4630 PRINT AT 3,17:"N O T E"
4640 PRINT AT 4,17:"-----"
4650 SET #102:INK 1
4660 PRINT AT 9,5:"At an time durin
g the game you can"
4670 PRINT AT 11,3:"refer to the in
structions by"
4680 PRINT AT 13,3:"pressing the ST
OP key."
4690 PRINT AT 16,5:"You can quit th
e game at any "
4700 PRINT AT 18,3:"time by pressin
g the ESCAPE key."
4710 SET #102:INK 2
4720 PRINT AT 23,8:"Press any key t
o continue"
4730 DO
4740 GET A$
4750 LOOP WHILE A$=""
4760 CLEAR SCREEN
4770 IF EXTYPE=9229 THEN
4780 DISPLAY #2:AT 26 FROM 1 TO 2
4790 DISPLAY #3:AT 1 FROM 1 TO 1
4800 DISPLAY #1:AT 2 FROM 1 TO 24
4810 CLOSE #99
4820 SET INTERRUPT STOP ON
4830 END IF
4840 RETRY
4850 END HANDLER
4860 !
4870 !
4880 !
4890 DEF ANY_KEY
4900 SET #102:INK 2
4910 PRINT AT 23,8:"Press any key t
o continue"
4920 DO
4930 GET A$
4940 LOOP WHILE A$=""
4950 CLEAR SCREEN
4960 PRINT
4970 SET #102:INK 2
4980 PRINT AT 2,12:"-----"
--" ! 17 minus signs
4990 PRINT AT 3,12:"INSTRUCTIONS co
nt"
5000 PRINT AT 4,12:"-----"
--"
5010 SET #102:INK 1
5020 END DEF
5030 !

```

```

5040 !
5050 !
5060 DEF SET_UP2
5070 RANDOMIZE
5080 LET ROW=0
5090 LET WON$="lose"
5100 LET COL$="0000000"
5110 LET CODE$="0"
5120 FOR N=1 TO YY
5130 LET CODE$=CODE$&STR$(RND(XX)
+1)
5140 NEXT N
5150 END DEF
5160 !
5170 !
5180 !
5190 DEF TOTAL_SCORES_ETC.
5200 TEXT 40
5210 SET #102:PALETTE BLACK,WHITE,B
LACK,CYAN
5220 PRINT
5230 SET #102:INK 2
5240 PRINT AT 1,14:"-----"
5250 PRINT AT 3,14:"-----"
5260 PRINT AT 2,14:"S C O R E S"
5270 SET #102:INK 1
5280 LET TOTAL_SCORE(1)=TOTAL_SCORE
(1)+SCORE(1)
5290 SELECT PLAY
5300 CASE 1
5310 IF GAME_NO.=1 THEN
5320 PRINT AT 7,4:"Your total s
core after ";GAME_NO.;" game."
5330 ELSE
5340 PRINT AT 7,4:"Your total s
core after ";GAME_NO.;" games."
5350 END IF
5360 PRINT AT 10,17:TOTAL_SCORE(1
)
5370 CASE 2
5380 IF GAME_NO.=1 THEN
5390 PRINT AT 6,5:"Total scores
after ";GAME_NO.;" game."
5400 ELSE
5410 PRINT AT 6,5:"Total scores
after ";GAME_NO.;" games."
5420 END IF
5430 LET TOTAL_SCORE(2)=TOTAL_SCO
RE(2)+SCORE(2)
5440 PRINT AT 9,7:"Player 1
Player 2"
5450 PRINT AT 11,9:TOTAL_SCORE(1)
5460 PRINT AT 11,24:TOTAL_SCORE(2
)
5470 END SELECT
5480 PRINT AT 14,10:"Do you wish to
:-"
5490 PRINT AT 17,5:“(1) continue
this match.”
5500 PRINT AT 19,5:“(2) start
a new match.”
5510 PRINT AT 21,5:“(3) q
uit program.”

```

```

5520 SET #102:INK 2
5530 PRINT AT 24,8:"Press appropriate key.;"
5540 DO
5550 GET A$
5560 LOOP UNTIL A$="1" OR A$="2" OR A$="3"
5570 IF A$="2" THEN LET CHOICE$="restart"
5580 IF A$="3" THEN LET CHOICE$="end"
5590 END DEF
5600 !
5610 !
5620 !
5630 DEF TITLE_PAGE
5640 CLOSE #105
5650 SET VIDEO MODE 1
5660 SET VIDEO COLOUR 2
5670 SET VIDEO X 20
5680 SET VIDEO Y 10
5690 OPEN #2:"video:"
5700 DISPLAY #2:AT 1 FROM 2 TO 10
5710 SET VIDEO COLOUR 1
5720 SET VIDEO X 14
5730 SET VIDEO Y 3
5740 OPEN #1:"video:"
5750 PRINT #1,AT 3,1:"CODE-QUEST"
5760 FOR A=1 TO 6
5770 SET #2:COLOUR A+1,RGB(A/6,A/6,1)
5780 NEXT A
5790 SET #1:COLOUR 1,RED
5800 FOR N=6 TO 34 STEP 4
5810 SET #2:INK N/5+1
5820 FOR W=7 TO 318 STEP 4
5830 LOOK #1,AT W,N:Q
5840 IF Q<>0 THEN
5850 PLOT #2:W*2,50+N*6;W*2,50+N*6-15
5860 END IF
5870 NEXT W
5880 NEXT N
5890 FOR N=1 TO 18
5900 DISPLAY #2:AT N FROM 1 TO 9
5910 NEXT
5920 CLEAR #1
5930 PRINT #1:
5940 PRINT #1,AT 1,1:" by"
5950 PRINT #1,AT 2,1:" Robin Bowes"
5960 FOR N=17 TO 4 STEP-1
5970 DISPLAY #2:AT N FROM 2 TO 10
5980 DISPLAY #1:AT N+8 FROM 1 TO 3
5990 NEXT N
6000 FOR N=1 TO 200
6010 SET #1:COLOUR 1,RND(255)
6020 NEXT N
6030 CLOSE #1
6040 CLOSE #2
6050 OPEN #105:"keyboard:"
6060 CLEAR FKEYS
6070 END DEF

```

Für Spiele-Freaks

Sicher haben sich die eingefleischten Spiele-Freaks schon oft gefragt, wie man auf dem ENTERPRISE Spiele so beeinflusst, daß mehrere Leben zur Verfügung stehen, der nächste Level angewählt wird usw. Dieser kleine Artikel soll hier weiterhelfen (leider geht das nur mit Controller): Legen Sie die Diskette mit dem jeweiligen Programm (Dot Breaker und -Collector) in Laufwerk A: und eröffnen Sie eine :RAMDISK 3/D.

Die beiden abgedruckten Listings enthalten in Zeile 10 eine Variable VAR mit einer bestimmten Adresse, in der der Wert der vorhandenen Leben steht. Nun wird das jeweilige Programm geladen und die neue Version auf der Ramdisk abgespeichert. Die neue Version kann dann auf die Diskette kopiert werden.

(T.S.)

```

10 LET VAR=963
100 OPEN #150:"a:dotbreak" ACCESS INPUT
110 OPEN #160:"e:write" ACCESS OUTPUT
120 FOR N=100 TO 9000
130 GET #150:A$
135 IF N=VAR THEN LET A$=CHR$(0)
140 PRINT #160:A$;
150 NEXT N
160 CLOSE #150
170 CLOSE #160

```

```

10 LET VAR=1010:LET ZAHL=3
100 OPEN #150:"a:collect" ACCESS INPUT
110 OPEN #160:"e:write" ACCESS OUTPUT
120 FOR N=100 TO 9000
130 GET #150:A$
135 IF N=VAR THEN LET A$=CHR$(ZAHL)
140 PRINT #160:A$;
150 NEXT N
160 CLOSE #150
170 CLOSE #160

```

Eine Liste von mehreren Möglichkeiten werden wir in einem der nächsten Info's abdrucken.

Hier noch ein weiterer Tip: In Grid Trouble, dem Spielhallenklassiker von Juergen Letzel auf dem ENTERPRISE kann durch gleichzeitiges Drücken von ALT und Joystick nach links der nächste Level erreicht werden.

Beitritts-Schein

Ich möchte Mitglied der ENTERPRISE USER GROUP werden. Hierzu bezahle ich DM 5.-- für den Eintritt und einen Monatsbeitrag von DM 2.-- (für mindestens ein halbes Jahr). Ich weiß, daß ich bei vorzeitigem Austritt kein Geld zurückerhalten kann.

Anrede

Name, Vorname

Straße, Hausnr.

Plz, Wohnort

Vorwahl, Tel.

Ich bezahle den Betrag vor Absendung dieses Abschnittes durch

- Überweisung auf Konto
- Barzahlung
- Scheck bzw. V-Scheck
- Briefmarken für _____ Monate (mind. 6)

Ich besitze folgende Hardware und benutze folgende Software:

- ENTERPRISE ____k Computer _____
- Disklaufwerk (3,5" / 5,25") _____
- Drucker (EP80+ / sonst.) _____
- Kassettenrekorder _____
- Monitor (col. / mon.) _____
- Telefon-Modem _____
- Mausepaket _____
- Sprachsynthesizer _____
- Sonstiges (_____) _____

Außerdem habe ich noch Fragen bzw. Vorschläge: _____

Datum, Unterschrift: _____