

**Computer**  
persönlich

Franz Ende

Das große

# SPIELE- BUCH

Commodore 64



46 Spielprogramme zum Eintippen  
Viel über Programmier technik · Grafikerstellung  
Ansteuerung von Joysticks · Paddles



# Das große Spielebuch Commodore 64



Franz Ende

# Das große Spielebuch Commodore 64

46 Spielprogramme zum Eintippen ·  
Viel über Programmiertechnik ·  
Grafikerstellung · Ansteuerung von  
Joysticks · Paddles

Deutsche Übersetzung:  
Peter Lücke

Markt & Technik Verlag

CIP-Kurztitelaufnahme der Deutschen Bibliothek

**Ende, Franz:**

Das große Spielebuch Commodore 64 : 46 Spielprogramme  
zum Eintippen ; viel über Programmieretechnik ; Grafikerstellung ;  
Ansteuerung von joysticks ; paddles / Franz Ende.

Dt. Übers. Peter Lücke. — Haar bei München :  
Markt u. Technik Verlag, 1984.

(Computer persönlich)

Einheitssacht.: The great book of games <dt.>

ISBN 3-922120-63-6

Die Informationen im vorliegenden Buch werden ohne Rücksicht auf einen eventuellen Patentschutz  
veröffentlicht.

Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.

Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen.  
Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autoren können  
für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine  
Haftung übernehmen.

Für Verbesserungsvorschläge und Hinweise auf Fehler sind Verlag und Herausgeber dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen  
Medien. Die gewerbliche Nutzung der in diesem Buch gezeigten Modelle und Arbeiten ist nicht zulässig.

»Commodore« ist eine Produktbezeichnung der Commodore Büromaschinen GmbH, Frankfurt,  
die ebenso wie der Name »Commodore« Schutzrechte genießt. Der Gebrauch bzw. die Verwendung  
bedarf der Erlaubnis der Schutzrechtsinhaberin.

Autorisierte Übersetzung und Bearbeitung des im Verlag  
ELCOMP PUBLISHING, INC. erschienenen Buches:  
»THE GREAT BOOK OF GAMES«

English edition © Copyright 1984  
by ELCOMP PUBLISHING, INC., 53 Redrock Lane, Pomona, CA 91766, USA  
All rights reserved

15 14 13 12 11 10 9 8 7 6

88 87 86 85

ISBN 3-922120-63-6

© 1984 by Markt & Technik, 8013 Haar bei München  
Alle Rechte vorbehalten  
Einbandgestaltung: Grafikdesign Heinz Rauner  
Druck: Eimannsberger, München  
Printed in Germany

## Vorwort

Der Commodore 64 bietet erstaunlich viel für seinen Anschaffungspreis. Der große 64K-Arbeitsspeicher, ein hervorragender Grafikteil, der interessante Tonerzeugungsteil und nicht zuletzt eine der besten Tastaturen in dieser Preisklasse, sind handfeste Vorteile gegenüber allen Mitbewerbern.

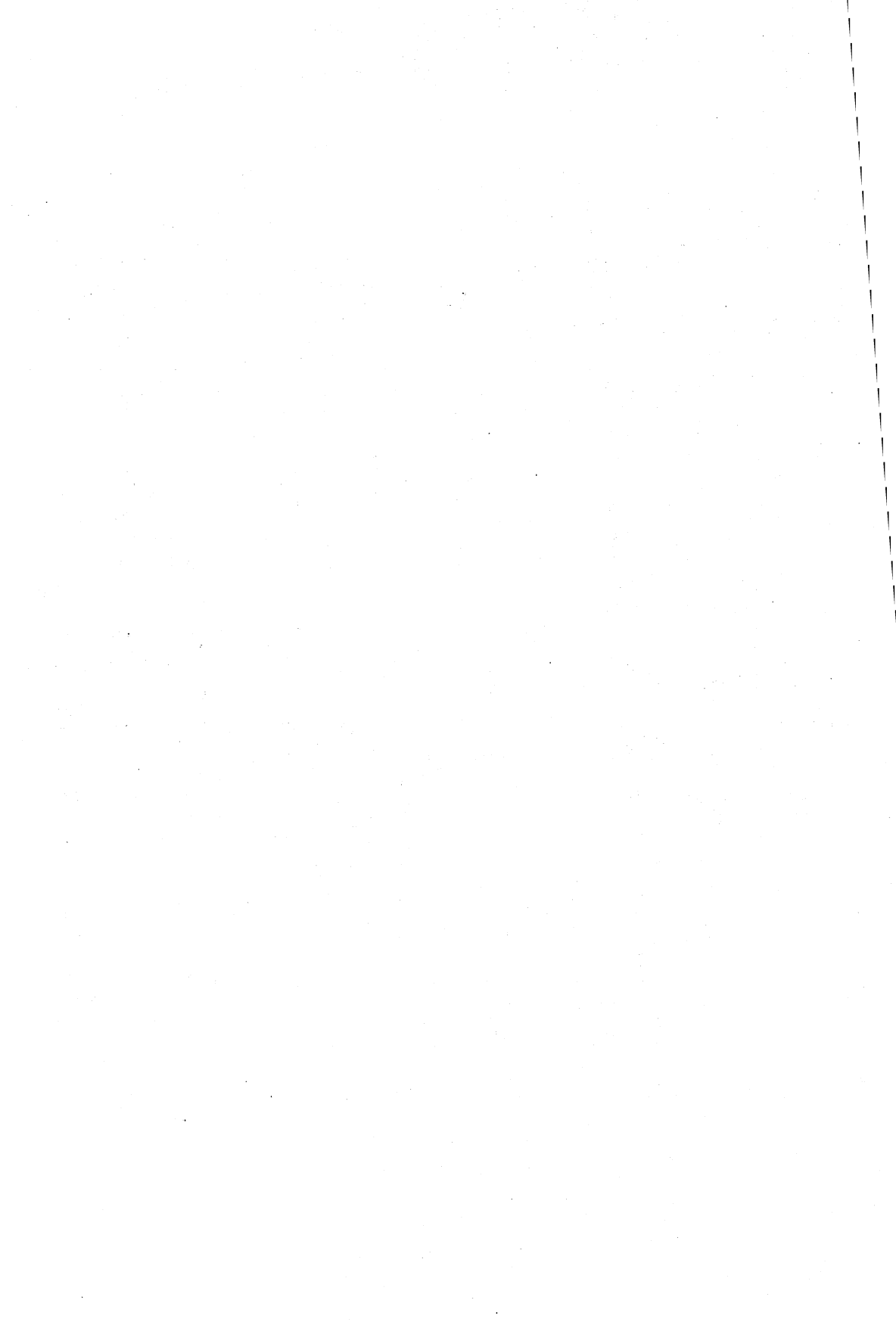
Der Rechner arbeitet mit der 6510-CPU, einer Weiterentwicklung der beliebten und weitverbreiteten 6502-CPU.

Dieses Buch ist in zwei Teile gegliedert. Im ersten Teil finden Sie eine Reihe unterhaltsamer Spielprogramme, die Sie nur noch eintippen müssen, um sie auf Ihrem 64'er zu benutzen. Wenn Ihnen das zu mühsam ist, können Sie vom Verlag alle in diesem Buch gezeigten Programme auf Diskette bzw. Kassette erhalten. Alle Programme sind in 'Commodore-Basic' geschrieben.

Der zweite Teil des Buches soll Ihnen Anregungen geben, wie Sie selbst mit entsprechenden Programmen die Leistungsfähigkeit Ihres Rechners voll nutzen können. Kleine, übersichtlich gehaltene Beispielprogramme zeigen Ihnen, wie die einzelnen Ausstattungsmerkmale angesprochen werden können. Sie erlernen den Umgang mit der Farbgrafik, wie man Sprites und den Tonteil einsetzt und welche Möglichkeiten mit Joysticks und Paddles zur Verfügung stehen.

Wenn Sie den zweiten Teil des Buches aufmerksam durcharbeiten, dürfte es Ihnen keine Schwierigkeiten mehr bereiten, eigene Programme zu schreiben, oder nach Lust und Laune bereits vorhandene zu ändern. Wir wünschen Ihnen viel Spaß dabei.

Franz Ende



## Vorwort zur deutschen Ausgabe

Vor gar nicht allzu langer Zeit hätten manchem Rechnerbetreiber die Haare zu Berge gestanden, wäre jemand auf die Idee gekommen, sein im wahrsten Sinne des Wortes teuerstes Stück nahezu ausschließlich für Hobby und Spiel einzusetzen.

Heute ist das ganz anders. Die erstaunliche Zunahme der Leistungsfähigkeit, gepaart mit dem starken Preisrückgang der Computer, erlaubt mehr und mehr Interessierten die Anschaffung eines eigenen Gerätes. Einer der populärsten Rechner für den privaten und den semiprofessionellen Bereich ist sicher der Commodore 64, dessen Hardwareausstattung auch im Hinblick auf die Erstellung von Spielen einiges zu bieten hat.

Entsprechend dem englischen Original geht das Buch von diesem Rechner aus. Es bietet eine gute Mischung aus Gebrauchsprogrammen für den Spielbetrieb und aus Anregungen und konkreten Beispielen für die ersten eigenen Gehversuche als Commodore-64-Programmierer. Wie der Übersetzer und seine Familie aus eigener Erfahrung bestätigen können, dürften sicher nicht nur Anfänger ihren Spaß an diesem Buch und den darin enthaltenen Spielprogrammen haben.

Peter Lüke













## Inhaltsverzeichnis

Vorwort	5
Vorwort zur deutschen Ausgabe	7
Inhaltsverzeichnis	9
C-64 Cursor-Kontroll-Zeichen	11
Spielautomat	13
Roulette	17
Mäuse	25
Die tollen Drei	29
Lotto	35
Billard	37
Flip-Flop	41
Irrgarten	47
Angriff	49
Artillerie	53
Das magische Quadrat	59
Karussell	61
Die Entfernung zwischen zwei Städten	63
Kurven	67
Mosquitos	69
Weltzeit	73
Konzentrationstest	77
Ballspiel	81
Space shuttle	83
Das Arbeiten mit binären Zahlen	85
Der freie Fall und der Wurf nach oben	89
Der schräge Wurf	91
Grafik mit dem Commodore 64	93
Wie man mit der hochauflösenden Grafik arbeitet	101
Das Arbeiten mit mehreren Farben	107
Das Verschieben des Bildinhalts	109
Sprites	113
Ein Editor für Sprites	117
Die Priorität von Sprites	121
Wie man Spritekollisionen feststellt	123
Der Tonteil	131
Joysticks und Paddles	139



## C-64 Cursor-Kontroll-Zeichen

Da man die Steuerzeichen zur Cursor-Bewegung aus dem BASIC-Listing oft nicht leicht entziffern kann, haben wir hier die gebräuchlichsten Befehle zur Cursorpositionierung zusammengefaßt:

ASCII	Druckbild	Beschreibung
146		invers ein (mit CTRL-Taste)
18		invers aus (mit CTRL-Taste)
3		RUN/STOP
147		Bildschirm löschen (CRL)
19		Cursor in linke obere Ecke
145		Cursor nach oben
17		Cursor nach unten
157		Cursor nach links
29		Cursor nach rechts
148		Zeichen einfügen



## Spielautomat

Dieses Programm 'simuliert' einen Spielautomaten. Am Anfang müssen Sie Geld einwerfen ('Münzen' zu DM 1.00, DM 2.00 oder DM 5.00). Dies geschieht, indem Sie die Tasten 1, 2 oder 5 drücken. Ihr momentanes Guthaben wird vom 'Automaten' angezeigt. Wie Sie dies von anderen Spielautomaten her kennen, werden in drei Fenstern durchlaufende Zahlen und Symbole angezeigt. Sie können den Durchlauf mit der Leertaste stoppen. Bei bestimmten Kombinationen, die Ihnen vom Programm vor Spielbeginn genannt werden, können Sie unterschiedlich große Beträge gewinnen. Sie werden Ihnen gutgeschrieben. Für jedes Spiel werden DM 0.20 von Ihrem Guthaben abgezogen.

```

10 REM SPIELAUTOMAT
31 GOTO132
32 PRINT" "
33 PRINT"#####|_____|"
34 PRINT"#####|_____|"
35 PRINT"#####| 064-GLUECK  |"
36 FORZ=1TO4:PRINT"#####|#####|"
37 NEXTZ
38 PRINT"#####| * \ | / * |"
39 PRINT"#####| * \ | / * |"
40 PRINT"#####| * _____ * |"
41 PRINT"#####| * | _____ | * |"
42 PRINT"#####| * _____ * |"
43 PRINT"#####| *   ↑↑   * |"
44 PRINT"#####| * GUT- * |"
45 PRINT"#####|   HABEN   |"
46 PRINT"#####|   - - -   |"
47 PRINT"#####|<- STOP |"
48 PRINT"=LEERTASTEN|   |"
49 PRINT"#####|   |"
50 PRINT"#####|_____|"
51 PRINT"#####|_____||_____||_____||"
52 PRINT"#####|100 ||100 ||100 ||"
53 PRINT"#####|_____||_____||"
54 REM
55 GOSUB109

```

```

56 IFG=0THEN100
57 FORX=1TO20
58 GETQ$: IFQ$=" "ANDX>2THENX=20:GOTO78
59 IFX=2THENG=G-20:GOSUB109
60 A=INT(RND(1)*10)*10:POKE1240,87:POKE55512,4:IF(A/20)<>INT(A/20)THEN60
61 POKE1245,81:POKE55517,4
62 B=INT(RND(1)*10)*10:POKE1245,87:POKE55517,4:IF(B/20)<>INT(B/20)THEN62
63 POKE1240,81:POKE55512,4
64 C=INT(RND(1)*10)*10:POKE1240,87:POKE55512,4:IF(C/20)<>INT(C/20)THEN64
65 POKE1245,81:POKE55517,4
66 IFA=0THENA$="**":GOTO68
67 A$=RIGHT$(STR$(A),2)
68 IFB=0THENB$="**":GOTO70
69 B$=RIGHT$(STR$(B),2)
70 IFC=0THENC$="**":GOTO72
71 C$=RIGHT$(STR$(C),2)
72 REM
73 PRINT"XXXXXXXXXXXXXXXXXXXXXXXX";
74 IFA=BANDB=CTHENPRINT"A$"B$"C$":GOTO78
75 IFAC>CANDR=0THENPRINTA$B$C$:GOTO78
76 IFA=CANDB=0THENPRINT"A$"B$"C$":GOTO78
77 PRINTA$B$C$
78 NEXTX
79 POKE1240,102:POKE1245,102:POKE55512,4:POKE55517,4
80 IFA=0ANDB=0ANDC=0THENG=G+5000:GOSUB118:GOSUB125:GOTO88
81 IFA=CANDB=0THENG=G+A:GOTO88
82 IFB=0ANDR<>CTHENG=G+20:GOTO88
83 IFA=BANDR=CTHENG=G+A:GOTO88
84 IFA=BANDB<>CTHEN88
85 IFA=CANDB<>CTHEN88
86 IFB=CANDR<>BTHEN88
87 IFR<>BANDR<>CTHENG=G+0
88 GOSUB109
89 IFG<20THEN100
90 PRINT" ";FORI=1TO20:PRINT" ";NEXT:FORI=1TO17:PRINT" ";NEXT
91 PRINT" "
92 FORI=1TO18:PRINT" ";NEXT:PRINT" "
93 PRINT"          LEERTASTE DRUECKEN"
94 GETF$:IFF$<>" "THEN94
95 PRINT" ";FORI=1TO20:PRINT" ";NEXT:FORI=1TO17:PRINT" ";NEXT
96 PRINT"          "
97 FORI=1TO18:PRINT" ";NEXT:PRINT" "
98 PRINT"          "
99 GOTO54

```



```

170 PRINT" 80 80 80  = 80 PFENNIG
171 PRINT" 60 60 60  = 60 PFENNIG
172 PRINT" 40 40 40  = 40 PFENNIG
173 PRINT" 20 20 20  = 20 PFENNIG
174 PRINT" 20 20 60  = -----
175 PRINT" ** 60 40  = -----
176 PRINT" 20 40 **  = -----
177 PRINT" 80 80 **  = -----
178 PRINT:PRINT"STERNE IN DER MITTE SIND EIN JOKER."
179 PRINT"WENN DIE ZAHLEN LINKS UND RECHTS GLEICH"
180 PRINT"SIND. GEWINN=ANGEZEIGTER BETRAG"
181 PRINT
182 PRINT"WEITER --> LEERTASTE DRUECKEN"
184 GETN$:IFN$<>" "THEN184
185 GOTO32
187 IFE>0THENPRINT"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
188 FORZ=1TO150:NEXT
190 PRINT"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
191 GOTO100

```

## Roulette

Das Programm bildet ein Roulettespiel nach, dem die international üblichen Spielregeln für Roulette zugrunde liegen. Sie können auf Zahl (1-36), auf Rot (alle roten Zahlen), auf Schwarz (alle schwarzen Zahlen), auf die ersten, auf die zweiten oder auf die dritten zwölf Zahlen, auf alle geraden oder ungeraden Zahlen und auf die ersten oder auf die zweiten achtzehn Zahlen setzen. Das Programm fragt Sie am Anfang, wieviel Geld Ihnen zur Verfügung steht und zeigt Ihnen, mit welchen Symbolen gesetzt wird.

```
100 REM ROULETTE
110 PRINT"?"
120 PRINT"      ♠ ROULETTE ♠"
130 PRINT:PRINT
140 PRINT"DIESES ROULETTESPIEL ENTSPRICHT DEN"
150 PRINT"INTERNATIONALEN REGELN"
160 PRINT
170 PRINT"SIE KOENNEN AUF DIE ZAHLEN VON 1 BIS 36"
180 PRINT"SETZEN"
190 PRINT
200 PRINT"SIE KOENNEN AUSSERDEM AUF REIHEN,"
210 PRINT"SPALTEN, GERADE UND UNGERADE SOWIE AUF"
220 PRINT"DUTZEND UND SERIEN SETZEN"
230 PRINT"BEISPIEL:"
240 PRINT"♠12♠=REIHE 1-2-3"
250 PRINT"♠456♠=REIHE 4-5-6"
260 PRINT"♠34-35-36♠"
270 PRINT
280 PRINT"WENN NULL KOMMT, DANN ERHAELT DIE BANK ALLE EINSATZE"
300 PRINT
310 PRINT"WEITER --> BELIEBIGE TASTE DRUECKEN"
320 GETA$:IFA$=""THEN320
330 PRINT"?"
340 PRINT
350 PRINT"PAIR (♠P♠)=ALLE GERADEN ZAHLEN"
360 PRINT"IMPAIR (♠I♠)=ALLE UNGERADEN ZAHLEN"
370 PRINT
380 PRINT"MANQUE (♠MA♠)=ZAHLEN 1-18"
390 PRINT"PASSE (♠PA♠)=ZAHLEN 19-36"
```

```

400 PRINT
410 PRINT"RD1=ZAHLEN 1-12"
420 PRINT"RD2=ZAHLEN 13-24"
430 PRINT"RD3=ZAHLEN 25-36"
440 PRINT
450 PRINT"R=ROTE ZAHLEN"
460 PRINT"S=SCHWARZE ZAHLEN"
470 PRINT:PRINT"WEITER--> BELIEBIGE TASTE DRUECKEN"
480 GETA$:IFA$=""THEN480
490 PRINT"J"
500 GOSUB2210
510 FORZZ=1TO5
520 X$="*":Z$="J":U$=" "
530 X=INT(RND(1)*37)
540 IFX<10THENY$="J":GOTO560
550 Y$="J1"
560 GOSUB2180
570 IFG=1THENA$="R":GOTO590
580 A$="B"
590 G=0:PRINT"RD000":IFX<10THENPRINTTAB(6)"X"|"A$:GOTO610
600 PRINTTAB(5)"X"|"A$
610 IFX>12THEN630
620 ONX+1GOTO660,690,700,710,720,730,740,750,760,770,780,790,800
630 IFX>25THEN650
640 ONX-12GOTO810,820,830,840,850,860,870,880,890,900,910,920,930
650 ONX-25GOTO940,950,960,970,980,990,1000,1010,1020,1030,1040
660 PRINT"RD":PRINTTAB(17)X$
670 GOSUB2000
680 PRINTTAB(17)Y$:GOTO1050
690 K=1:GOTO2480
700 K=2:GOTO2480
710 K=3:GOTO2480
720 K=1:GOTO2490
730 K=2:GOTO2490
740 K=3:GOTO2490
750 K=1:GOTO2500
760 K=2:GOTO2500
770 K=3:GOTO2500
780 K=1:GOTO2510
790 K=2:GOTO2510
800 K=3:GOTO2510
810 K=1:GOTO2520
820 K=2:GOTO2520
830 K=3:GOTO2520

```

```

840 K=1:GOTO2530
850 K=2:GOTO2530
860 K=3:GOTO2530
870 K=1:GOTO2540
880 K=2:GOTO2540
890 K=3:GOTO2540
900 K=1:GOTO2550
910 K=2:GOTO2550
920 K=3:GOTO2550
930 K=1:GOTO2560
940 K=2:GOTO2560
950 K=3:GOTO2560
960 K=1:GOTO2570
970 K=2:GOTO2570
980 K=3:GOTO2570
990 K=1:GOTO2580
1000 K=2:GOTO2580
1010 K=3:GOTO2580
1020 K=1:GOTO2590
1030 K=2:GOTO2590
1040 K=3:GOTO2590
1050 NEXTZZ
1060 GOSUB2000
1070 GOSUB2170
1080 PRINT"                SIE SIND DRAN!"
1090 GOSUB2000:GOSUB2170
1100 INPUT"WIEVIEL GELD HABEN SIE";A5
1110 GOSUB2170
1120 INPUT"SETZEN SIE (ZAHL O. GRUPPE)";B#
1130 GOSUB2170
1140 INPUT"IHR EINSATZ";C
1150 IF C<=A5 THEN 1200
1160 GOSUB2170:PRINT"SOVIEL GELD BESITZEN SIE NICHT MEHR!":GOSUB2000
1170 GOSUB2170:GOSUB2000
1180 GOSUB2170:GOTO1140
1190 GOSUB2130:GOSUB2170
1200 IF VAL(B#)>0 AND VAL(B#)<37 THEN 1790
1210 IF B#="F" THEN 1400
1220 IF B#="I" THEN 1450
1230 IF B#="MA" THEN 1500
1240 IF B#="PA" THEN 1530
1250 IF B#="D1" OR B#="D2" OR B#="D3" THEN 1570
1260 IF B#="R" THEN 1620
1270 IF B#="B" THEN 1650

```

```

1280 IFB$>"T"ANDB$<"TN"ANDRIGHT$(B$,1)<>"J"THEN1310
1290 GOSUB2170:PRINT"FALSCH EINGABE, BITTE WIEDERHOLEN";:GOSUB2000
1300 GOSUB2170:GOTO1120
1310 GOSUB2000
1320 IF(B$="TA"ANDX<4)OR(B$="TB"ANDX>3ANDX<7)THEN1390
1330 IF(B$="TC"ANDX>6ANDX<10)OR(B$="TD"ANDX>9ANDX<13)THEN1390
1340 IF(B$="TE"ANDX>12ANDX<16)OR(B$="TF"ANDX>15ANDX<19)THEN1390
1350 IF(B$="TG"ANDX>18ANDX<22)OR(B$="TH"ANDX>21ANDX<25)THEN1390
1360 IF(B$="TI"ANDX>24ANDX<28)OR(B$="TK"ANDX>27ANDX<31)THEN1390
1370 IF(B$="TL"ANDX>30ANDX<34)OR(B$="TM"ANDX>33)THEN1390
1380 GOTO1720
1390 W=12:GOTO1680
1400 GOSUB2000
1410 J=1
1420 IFJ=XTHEN1720
1430 J=J+2:IFJ<36THEN1420
1440 GOTO1670
1450 GOSUB2000
1460 J=1
1470 IFJ=XTHEN1670
1480 J=J+2:IFJ<36THEN1470
1490 GOTO1720
1500 GOSUB2000
1510 IFX>0ANDX<19THEN1670
1520 GOTO1720
1530 GOSUB2000
1540 IFX>0ANDX<19THEN1720
1550 GOTO1670
1560 END
1570 GOSUB2000
1580 IFB$="D1"ANDX>0ANDX<13THENW=2:GOTO1680
1590 IFB$="D2"ANDX>12ANDX<25THENW=2:GOTO1680
1600 IFB$="D3"ANDX>24THENW=2:GOTO1680
1610 GOTO1720
1620 GOSUB2000:GOSUB2180
1630 IFG=1THENG=0:GOTO1670
1640 GOTO1720
1650 GOSUB2000
1660 G=0:GOSUB2180:IFG=1THENG=0:GOTO1720
1670 W=1
1675 PRINT"XXXXXXXXTAB(6)" "
1680 PRINT"XXXXXXXXTAB(6)X" " :GOSUB2160
1690 PRINT"          SIE HABEN GEWONNEN! "TAB(24):STR$(A5+W*C)" "
1700 GOSUB2000:PRINT"TT":GOSUB2170

```

```

1710 A5=A5+W#C:GOTO1120
1715 PRINT"XXXXXXXXTAB(6)" "
1720 PRINT"XXXXXXXXTAB(6)X" " "
1730 GOSUB2160
1740 PRINT"          SIE HABEN VERLOREN!";TAB(24);STR$(A5-C);" "
1750 A5=A5-C:GOSUB2000
1760 GOSUB2170
1770 IFA5>0THEN1120
1780 GOTO1830
1790 B=VAL(B$)
1800 GOSUB2080
1810 IFX=BTHENW=36:GOTO1680
1820 GOTO1720
1830 PRINT"!"
1840 PRINT"      SIE SIND"
1850 PRINT
1860 PRINT"      P L E I T E!"
1870 PRINT:PRINT:PRINT
1880 PRINT"WENN SIE WEITER MACHEN WOLLEN, BRAUCHEN"
1890 PRINT"MEHR GELD, UND SIE MUESSEN EINE TASTE"
1895 PRINT"DRUECKEN"
1900 GETA$:IFA$=""THEN1900
1970 GOSUB2210:GOTO1070
1980 END
1990 PRINTTAB(28)Z$
2000 FORI=1TO3000:NEXTI:RETURN
2010 PRINTTAB(28)U$:GOTO1050
2020 PRINTTAB(15)X$:RETURN
2030 PRINTTAB(15)Y$:GOTO2010
2040 PRINTTAB(18)X$:RETURN
2050 PRINTTAB(18)Y$:GOTO2010
2060 PRINTTAB(21)X$:RETURN
2070 PRINTTAB(21)Y$:GOTO2010
2080 X=INT(RND(1)*37)
2090 IFX<>0THENRETURN
2094 PRINT"XXXXXXXXTAB(6)X" " "
2096 GOSUB2160
2100 PRINT"          DIE BANK HAT GEWONNEN!";TAB(24);STR$(A5-C);" "
2105 A5=A5-C:PRINT"!";
2110 IFA5<=0THENGOSUB2000:GOTO1830
2120 GOSUB2080:GOSUB2170:GOTO1120
2130 FORJ=1TO5PRINT"YOU ARE STUCK!";
2140 FORI=1TO70:NEXTI

```

```

2150 PRINT"YOU ARE STUCK!":FORI=1TO70:NEXTI:NEXTJ:RETURN
2160 FORI=1TO15:PRINT:NEXTI
2170 PRINT"XXXXXXXXXXXXXXXXXXXXXXXXXXXX";FORE=0TO117:PRINT " ";NEXTE:PRINT " "
2175 PRINT"XXXXXXXXXXXXXXXXXXXXXXXXXXXX";:RETURN
2180 IFX=10RX=30RX=50RX=70RX=90RX=120RX=140RX=160RX=180RX=19THENG=1:RETURN
2190 IFX=210RX=230RX=250RX=270RX=300RX=320RX=340RX=36THENG=1
2200 RETURN
2210 PRINT"Q"
2220 PRINT"
                _____"
2230 PRINT"                |P|    0    |M|"
2240 PRINT"      _0_ |A| _____ |A|"
2250 PRINT" |---| |S|  11| 2|  33| IN|"
2260 PRINT" | | |S|  4|  55|  6 |Q|"
2270 PRINT" |---| |E|  77|  8|  99| IU|"
2280 PRINT" _____ | 10|11| 12||E|"
2290 PRINT"                _____"
2300 PRINT"                |P|13|14||15 |I|"
2310 PRINT"                |A|16||17|18||M|"
2320 PRINT"                |I|19||20|21||P|"
2330 PRINT"                |R|22|23||24 |.|"
2340 PRINT"                _____"
2350 PRINT"                |N|25||26|27||R|"
2360 PRINT"                |O|28|29|30||O| IHR REST"
2370 PRINT"                |I|31|32||33 |U| KAPITAL: "
2380 PRINT"                |R|34||35|36||G|"
2390 PRINT"                _____"
2400 PRINT"                | 11.12.13. | |"
2410 PRINT"                _____"
2420 PRINT
2430 RETURN
2440 ONKGO TO2450,2460,2470
2450 GOSUB2020:GOSUB1990:GOTO2030
2460 GOSUB2040:GOSUB1990:GOTO2050
2470 GOSUB2060:GOSUB1990:GOTO2070
2480 PRINT"XXXXX":GOTO2600
2490 PRINT"XXXXX":GOTO2600
2500 PRINT"XXXXX":GOTO2600
2510 PRINT"XXXXX":GOTO2440
2520 PRINT"XXXXX":GOTO2440
2530 PRINT"XXXXX":GOTO2440
2540 PRINT"XXXXX":GOTO2440
2550 PRINT"XXXXX":GOTO2440
2560 PRINT"XXXXX":GOTO2440
2570 PRINT"XXXXX":GOTO2440

```

```
2580 PRINT"XXXXXXXXXXXXXXXX":GOTO2440
2590 PRINT"XXXXXXXXXXXXXXXX":GOTO2440
2600 ONKGOTO2610,2620,2630
2610 PRINTTAB(16)X$:GOSUB1990:PRINTTAB(16)Y$:GOTO2010
2620 PRINTTAB(19)X$:GOSUB1990:PRINTTAB(19)Y$:GOTO2010
2630 PRINTTAB(22)X$:GOSUB1990:PRINTTAB(22)Y$:GOTO2010
```



## Mäuse

Dies ist ein Spiel für zwei Spieler. Jeder der beiden steuert eine Maus. Der erste Spieler benutzt dazu die Tasten A, W, D und X, der zweite die Tasten 4, 8, 6 und 2. Wer zuerst mit seiner Maus den Rand oder eine der beiden Mäusespuren berührt, hat verloren.

```
10 REM MAEUSE
20 PRINT"␣"          RULES":PRINT
22 PRINT"JEDER SPIELER KANN SEINE MAUS IN"
23 PRINT"VIER RICHTUNGEN BEWEGEN.  A,W,D,X"
24 PRINT"STEUERN DIE LINKE.  4,8,6,2 STEUERN"
25 PRINT"DIE RECHTE MAUS.  KEINE MAUS DARF DIE"
26 PRINT"WAND ODER EINE MAEUSES PUR BERUEHREN"
27 PRINT"DAS SPIEL GEHT UEBER SECHS RUNDEN"
30 PRINT:PRINT"WEITER --> TASTE DRUECKEN"
32 GETA$:IFA$=""THEN32
50 PRINT"␣"
65 GOTO1135
68 REM START
69 GOSUB4000:REM WAND
70 GOSUB3000:REM TASTE
80 IFL$="W"THENLY=LY-1:IFLY<0THENLY=0
90 IFL$="D"THENLX=LX+1:IFLX>39THENLX=39
100 IFL$="A"THENLX=LX-1:IFLX<0THENLX=0
110 IFL$="X"THENLY=LY+1:IFLY>24THENLY=24
200 PL=1024+LY*40+LX:CL=55296+LY*40+LX
205 IFPEEK(PL)<>32THEN1000
210 POKEPL,87:POKECL,8
220 GOSUB2000
270 GOSUB3000
280 IFR$="8"THENRY=RY-1:IFRY<0THENRY=0
290 IFR$="6"THENRX=RX+1:IFRX>39THENRX=39
300 IFR$="4"THENRX=RX-1:IFRX<0THENRX=0
310 IFR$="2"THENRY=RY+1:IFRY>24THENRY=24
400 PR=1024+RY*40+RX:CR=55296+RY*40+RX
405 IFPEEK(PR)<>32THEN1010
410 POKEPR,81:POKECR,13
```

```

420 GOSUB2000
500 GOT070
1000 FORI=1T01000:NEXTI:PRINT"#####DIE LINKE MAUS HAT BERUEHRT"
1005 ZR=ZR+1:GOTO1100
1010 ZL=ZL+1
1012 FORI=1T01000:NEXTI:PRINT"#####DIE RECHTE MAUS HAT BERUEHRT"
1100 FORI=1T02000:NEXTI:PRINT"#####PUNKTE"
1120 PRINT"#####";PRINTZL:PRINT":":PRINTZR:FORI=1T03000:NEXTI
1130 IFZL+ZR=6THEN1140
1135 FORI=1T0500:NEXTI:GOTO68
1140 PRINT"#####DAS WAR EIN GUTES SPIEL  !"
1150 FORI=1T03000:NEXTI
1160 PRINT"#####FUER NOCH EIN SPIEL TASTE DRUECKEN "
1170 GETA$:IFA$=""THEN1170
1180 ZL=0:ZR=0:GOTO68
2000 FORI=1T050:NEXTI
2010 RETURN
3000 GETA$
3010 IFA$<"W"THEN3100
3020 L$=A$:RETURN
3100 IFA$<"A"THEN3200
3120 L$=A$:RETURN
3200 IFA$<"X"THEN3300
3220 L$=A$:RETURN
3300 IFA$<"D"THEN3400
3320 L$=A$:RETURN
3400 IFA$<"2"THEN3500
3420 R$=A$:RETURN
3500 IFA$<"4"THEN3600
3520 R$=A$:RETURN
3600 IFA$<"8"THEN3700
3620 R$=A$:RETURN
3700 IFA$<"6"THEN3800
3720 R$=A$:RETURN
3800 RETURN
4000 S=1024:O=55296
4040 PRINT"O":V=0
4050 FORH=0T039:GOSUB5000:NEXTH:H=39
4060 FORV=0T023:GOSUB5000:NEXTV:V=23
4070 FORH=39T00STEP-1:GOSUB5000:NEXTH:H=0
4080 FORV=23T00STEP-1:GOSUB5000:NEXTV
4090 LX=1:LY=23
4100 RX=38:RY=23
4110 L$="W":R$="S"

```

```
4120 RETURN
5000 POKE(S+V*40+H),102:POKE(C+V*40+H),4
5010 RETURN
```



## Die tollen Drei

Das Programm erzeugt neun Quadrate, die ihrerseits in jeweils neun Quadrate unterteilt sind. Jedes der Quadrate ist mit einer Zahl von 3 bis 11 gekennzeichnet. Gewinner ist derjenige, der zuerst drei große Quadrate in einer Reihe oder in einer Spalte oder in einer der möglichen Diagonalen belegt hat, oder derjenige, der zuerst fünf große Quadrate in beliebiger Anordnung belegt. Die Würfel unterhalb des eigentlichen Spielfeldes geben an, welche der kleinen Quadrate belegt werden können. Erscheinen zum Beispiel eine Vier und eine Sechs, dann können alle der neun mit der Zahl 10 gekennzeichneten kleinen Quadrate belegt werden, die nicht schon von einem Mitspieler besetzt sind. Das gleiche gilt für alle kleinen Quadrate innerhalb des großen Quadrates mit der Nummer 10. Sie belegen ein Quadrat, indem Sie zuerst die Nummer des jeweiligen großen und dann, durch ein Komma getrennt, die Nummer des kleinen Quadrates angeben.

Wirft der Rechner die Würfel einmal so, daß eine nicht belegbare Zahlenkombination entsteht, würfelt er bis zu dreimal weiter. Anschließend ist der nächste Spieler an der Reihe.

Einige Zahlen besitzen eine besondere Bedeutung. Erhalten Sie eine Zwölf, dürfen Sie jede freie Position des Spielfeldes besetzen. Mit einer Sieben steht Ihnen jedes freie mittlere Quadrat und jede freie Position innerhalb des großen sieben Quadrates zu. Jede beliebige von Ihren Mitspielern belegte Position dürfen Sie löschen, wenn Sie eine Zwei bekommen. Ausgenommen davon sind nur große Quadrate, die schon vollständig einem Spieler gehören.

```

5 REM DIE TOLLEN DREI
10 DIM T%(12,11):A=INT(RND(2)*2+1)
20 PRINT " ";
30 PRINT " | (3) (4) (5) | (3) (4) (5) | (3) (4) (5) |";
40 PRINT " | | | | | | |";
50 PRINT " | (6) (33) (8) | (6) (34) (8) | (6) (35) (8) |";
60 PRINT " | | | | | | |";
70 PRINT " | (9)(10)(11) | (9)(10)(11) | (9)(10)(11) |";
80 PRINT " | | | | | | |";
90 PRINT " | (3) (4) (5) | (3) (4) (5) | (3) (4) (5) |";

```

```

100 PRINT" | | ";
110 PRINT" | (6) (8) | (6) (8) | (6) (8) | (6) (8) ";
120 PRINT" | | | | ";
130 PRINT" | (9)(10)(11) | (9)(10)(11) | (9)(10)(11) ";
140 PRINT" |-----| | ";
150 PRINT" | (3) (4) (5) | (3) (4) (5) | (3) (4) (5) ";
160 PRINT" | | | | ";
170 PRINT" | (6) (8) | (6) (8) | (6) (8) | (6) (8) ";
180 PRINT" | | | | ";
190 PRINT" | (9)(10)(11) | (9)(10)(11) | (9)(10)(11) ";
200 PRINT"-----";
210 PRINT" ";
220 PRINT" [ ] ";
230 PRINT" | | | | ";
240 PRINT" | | | | ";
250 PRINT" | | | | ";
260 PRINT" [ ] [ ] ";
270 PRINT"TTTT";
280 X=INT(RND(1)*6+1):Y=INT(RND(1)*6+1):Z=Y+X:IFZ=2THEN1520
290 FORR=3TO11:IFT%(Z,R)=0THENM=0:GOTO340
300 NEXTR
310 FORR=3TO11:IFT%(R,Z)=0THENM=0:GOTO340
320 NEXTR:M=M+1:IFM=3THEN1210
330 GOTO210
340 ONXGOTO350,360,370,380,390,400
350 PRINT" ";:GOTO410
360 PRINT" * ";:GOTO410
370 PRINT" * * ";:GOTO410
380 PRINT" * * * ";:GOTO410
390 PRINT" * * * * ";:GOTO410
400 PRINT" * * * * * ";
410 B=B+1:IFB=2THENB=0:GOTO430
420 ONYGOTO350,360,370,380,390,400
430 PRINTTAB(13)"SPIELER"R"IST AN DER REIHE"
440 INPUT" ";I,J
450 IFZ=2THEN1270
460 IFZ=7THEN490
470 IFZ=12THEN500
480 IFJ=7THEN1260
490 IFI<>ZANDJ<>ZTHEN1260
500 IFI<3ORJ<3THEN1260
510 IFI>11ORJ>11THEN1260
520 IFT%(I,J)<>0THEN1260
530 PRINT" ";:ONI-2GOTO540,670,680,690,700,710,720,730,740

```



```

980 PRINT"#####";GOTO1040
990 PRINT"#####";SPC(13);GOTO1040
1000 PRINT"#####";SPC(26);GOTO1040
1010 PRINT"#####";GOTO1040
1020 PRINT"#####";SPC(13);GOTO1040
1030 PRINT"#####";SPC(26);
1040 IFA=2THEN1070
1050 FORR=1TO5:FORV=1TO12:PRINT"※";NEXTV:PRINT"#####";NEXTR
1060 TX(I,2)=5:FORR=3TO11:TX(I,R)=A+4:NEXTR:GOTO1110
1070 FORR=1TO5:FORV=1TO12:PRINT"¶";NEXTV:PRINT"#####";NEXTR
1080 TX(I,2)=6:FORR=3TO11:TX(I,R)=A+4:NEXTR:FORR=3TO11:IFTX(R,2)=6THENQ=Q+1
1090 NEXTR:IFQ=5THEN1690
1100 GOTO1130
1110 FORR=3TO11:IFTX(R,2)=5THENQ=Q+1
1120 NEXTR:IFQ=5THEN1690
1130 Q=3:V=0
1140 FORR=QTOQ+2:V=V+TX(R,2):NEXTR:GOSUB1240
1150 Q=Q+3:IFQ=12THENQ=2:GOTO1170
1160 GOTO1140
1170 Q=Q+1:IFQ=6THEN1190
1180 FORR=QTOQ+6STEP3:V=V+TX(R,2):NEXTR:GOSUB1240:GOTO1170
1190 FORR=3TO11STEP4:V=V+TX(R,2):NEXTR:GOSUB1240
1200 FORR=5TO9STEP2:V=V+TX(R,2):NEXTR:GOSUB1240
1210 IFZ=2THEN210
1220 A=A+1:IFA=3THENA=1
1230 PRINT"TT":GOTO210
1240 IFV=15ORV=18THEN1690
1250 V=0:RETURN
1260 PRINTSPC(13)"REGELVERLETZUNG   #####TTTT":GOTO430
1270 IFI<3ORJ<3ORI>11ORJ>11THEN1260
1280 IFTX(I,2)<>0THEN1260
1290 IFTX(I,J)=0THEN1260
1300 IFA=1ANDTX(I,J)=5THEN1260
1310 IFA=2ANDTX(I,J)=6THEN1260
1320 PRINT"####!";
1330 ONI-2GOTO1340,1430,1440,1450,1460,1470,1480,1490,1500
1340 ONJ-2GOTO1350,1360,1370,1380,1590,1390,1400,1410,1420
1350 PRINT"3":GOTO1510
1360 PRINTSPC(4)"4":GOTO1510
1370 PRINTSPC(8)"5":GOTO1510
1380 PRINT"#####";GOTO1510
1390 PRINT"#####";SPC(8)"8":GOTO1510
1400 PRINT"#####";GOTO1510
1410 PRINT"#####10":GOTO1510

```





## Lotto

Mit diesem Programm können Sie Lottozahlen erzeugen. Es wählt aus Zahlen, deren Wert zwischen 1 und 49 liegt, jeweils sechs aus. Mit der Taste N erhalten Sie einen neuen Wertesatz.

```
5 REM LOTTO
10 T0$="" | | | | | | |
20 T1$="|_|_|_|_|_|_|_|"
30 T2$="| 01 |02 |03 |04 |05 |06 |07 |"
40 T3$="| 08 |09 |10 |11 |12 |13 |14 |"
50 T4$="| 15 |16 |17 |18 |19 |20 |21 |"
60 T5$="| 22 |23 |24 |25 |26 |27 |28 |"
70 T6$="| 29 |30 |31 |32 |33 |34 |35 |"
80 T7$="| 36 |37 |38 |39 |40 |41 |42 |"
90 T8$="| 43 |44 |45 |46 |47 |48 |49 |"
100 T9$="|_|_|_|_|_|_|_|"
110 S$=""
120 PRINT "J"
130 PRINT "N" S$+T1$
140 PRINT S$+T2$
150 PRINT S$+T0$
160 PRINT S$+T1$
170 PRINT S$+T3$
180 PRINT S$+T0$
190 PRINT S$+T1$
200 PRINT S$+T4$
210 PRINT S$+T0$
220 PRINT S$+T1$
230 PRINT S$+T5$
240 PRINT S$+T0$
250 PRINT S$+T1$
260 PRINT S$+T6$
270 PRINT S$+T0$
280 PRINT S$+T1$
290 PRINT S$+T7$
300 PRINT S$+T0$
310 PRINT S$+T1$
320 PRINT S$+T8$
```

```

320 PRINTS#+T8#
330 PRINTS#+T9#
340 PRINT"NOCH EIN SPIEL --> N DRUECKEN"
350 PRINT" SPIELLENDE --> E DRUECKEN"
360 X(0)=99
370 FORZ=1TO6
380 W=0
390 X(Z)=INT(RND(1)*49)
400 FORY=0TOZ-1
410 IFX(Z)=X(Y)THENW=1
420 NEXTY
430 IFW=1THEN380
440 XV=120*INT(X(Z)/7)
450 XH=4*(X(Z)-7*INT(X(Z)/7))
460 XX=1030+XV+XH:XC=55302+XV+XH
470 POKEXX,160:POKEXC,10
480 POKEXX+1,160:POKEXC+1,10
490 POKEXX+2,160:POKEXC+2,10
500 POKEXX+3,160:POKEXC+3,10
510 POKEXX+40,160:POKEXC+40,10
520 POKEXX+43,160:POKEXC+43,10
530 POKEXX+80,160:POKEXC+80,10
540 POKEXX+81,160:POKEXC+81,10
550 POKEXX+82,160:POKEXC+82,10
560 POKEXX+83,160:POKEXC+83,10
580 D=PEEK(XX+41)
590 POKEXX+41,D+128
600 D=PEEK(XX+42)
610 POKEXX+42,D+128
620 NEXTZ
630 GETA#:IFA#=""THEN630
640 IFA#="N"THEN130
650 END

```

## Billard

Dieses Spiel entspricht einem Billard-Spiel. Sie sehen einen Tisch mit zwei Kugeln, die mittels einer dritten mittig angestoßen werden sollen. Dazu geben Sie einen Winkel von 0 bis 360 Grad ein. Er bestimmt die Richtung, in der sich die dritte Kugel bewegt.

```
100 REM BILLARD
110 REM DIESES SPIEL SIMULIERT BILLARD
120 PRINT"♣"
400 GOSUB1000
410 GOSUB9000
420 GOSUB1500
500 P=P+1:IFP>NPTHENP=1:I=I+1
510 IFI>NITHEN7500
520 GOSUB2000
530 GOSUB2500
540 GOSUB3000
550 GOSUB3500
560 GOSUB4000
570 GOSUB4500
700 GOSUB5000
710 GOSUB5500
720 IFNB>=3THENGOSUB6000:GOTO500
730 GOSUB6500
740 IFCU>=8THENGOSUB6000:GOTO500
750 GOTO700
1000 RX=0:DX=0:RY=0:DY=0:TX=0:TY=0:XR=0:YU=0:CU=0:IX=0:IY=0:X1=0
1010 Y1=0:X2=0:Y2=0:A=0:B=0:NB=0:I=0:S1=0:S2=0:P=2:DIIMP$(2):RETURN
1500 GOSUB8000:GOSUB8400:PRINT"          ⚡COLLISION⚡":PRINT
1510 INPUT"SPIELERZAHL (1/2)";NP
1520 IFNP<1ORNP>2THEN1510
1530 IFNP=1THENINPUT"IHR NAME";P$(1):GOTO1560
1540 INPUT"ENTER NAME OF PLAYER 1";P$(1)
1550 INPUTDEFENTER NAME OF PLAYER 2";P$(2)
1560 INPUT"ZAHL DER ZUSAMMENSTOESSE";NI
1570 NX=39:NY=18:SR=.865
1590 K=RND(TI):XR=NX-2
1600 YU=NY-2:RETURN
```

```

2000 GOSUB8800
2010 GOSUB8400:PRINT"FERTIG ";P$(P):PRINT:PRINT
2020 GOSUB8900:GOSUB8800
2030 GOSUB8400:PRINT:PRINT
2040 PRINT"VERSUCH";I;"OF";NI,"STOESSE, "; " ";P$(P);" ";PRINT
2050 PRINT"SCORE      ";P$(1);" ";S1:IFNP=2THENPRINT"      ";P$(2);" ";S2
2060 RETURN
2500 GOSUB8600
2510 A=X:FORK=1TONX:PRINT" ";:GOSUB8100:NEXTK:A=L:B=U:C=X
2520 FORK=1TONY-1:GOSUB8300:NEXTK:GOSUB8600:A=U:B=X:C=L
2530 FORK=1TONY-1:GOSUB8300:NEXTK:A=X:FORK=1TONX-1:GOSUB8100
2540 NEXTK:PRINT" ";:RETURN
3000 X1=INT(RND(1)*(NX-3))+1:Y1=INT(RND(1)*(NY-3))+2
3010 X2=INT(RND(1)*(NX-3))+1:Y2=INT(RND(1)*(NY-3))+2
3020 K=ABS(X1-X2):Q=ABS(Y1-Y2)
3030 IFK<3ANDQ<3THEN3010
3040 XB=INT(RND(1)*(NX-4))+2:YB=INT(RND(1)*(NY-4))+2
3050 K=(X1-XB)^2+(Y1-YB)^2:IFK<4THEN3040
3060 K=(X2-XB)^2+(Y2-YB)^2:IFK<4THEN3040
3070 RETURN
3500 LX=X1:LY=Y1:GOSUB7000:A=OH:B=OH:GOSUB8200
3510 A=L:B=L:C=D:GOSUB8300:A=OH:B=OH:GOSUB8200
3520 LX=X2:LY=Y2:GOSUB7000:A=OH:B=OH:GOSUB8200
3530 A=L:B=L:C=D:GOSUB8300:A=OH:B=OH:GOSUB8200
3540 LX=XB:LY=YB:GOSUB7000:A=AS:GOSUB8100:RETURN
4000 GOSUB8400
4010 INPUT"WINKEL (0-360)";G:G=G*PI/180
4020 DX=COS(G):DY=SIN(G):IFSR>1THENDX=DX/SR
4030 IFSR<1THENDY=DY*SR
4040 IFABS(DX)<.5ANDABS(DY)<.5THENDX=DX*2:DY=DY*2:GOTO4040
4050 RETURN
4500 LX=XB:LY=YB:GOSUB7000
4510 NB=0:CU=0:IX=XB:IY=YB:RX=XB+.49:RY=YB+.49
4520 GOSUB8900:A=SP:B=L:GOSUB8200
4530 RETURN
5000 RX=RX+DX:RY=RY+DY
5010 TX=INT(RX):TY=INT(RY)
5020 A=N:B=N:IFTX>IXTHENA=R:GOTO5040
5030 IFTX<IXTHENA=L
5040 IFTY>IYTHENB=U:GOTO5060
5050 IFTY<IYTHENB=D
5060 GOSUB8200:IX=TX:IY=TY
5070 A=AS:B=L:GOSUB8200
5080 A=SP:B=L:GOSUB8200

```

```

5090 RETURN
5500 IFNB>1THEN5540
5510 IFIX<X2ORIX>X2+10RIY<Y2-10RIY>Y2THEN5540
5520 A=PL:B=L
5530 GOSUB8200:NB=NB+2
5540 IFNB=10RNB=3THEN5570
5550 IFIX<X1ORIX>X1+10RIY<Y1-10RIY>Y1THEN5570
5560 A=PL:B=L:GOSUB8200:NB=NB+1
5570 RETURN
6000 GOSUB8400
6010 PRINT
6020 IFNB<3THENPRINT"KEINE PUNKTE!";GOTO6070
6030 SC=4-ABS(CU-4):PRINT"DIESER STOSS ZAEHLT:";SC:IFP=1THENS1=S1+SC
6040 IFP=2THENS2=S2+SC
6050 PRINT:PRINT:PRINT"PUNKTE  ";P*(1);":":S1
6060 IFNP=2THENPRINT"  ";P*(2);"-":S2
6070 FORQ=1TO3:GOSUB8900:NEXTQ:RETURN
6500 IFIX>1ANDIX<XRTHEN6530
6510 CU=CU+1
6520 A=BL:GOSUB8100:DX=DX*-1
6530 IFIY>1ANDIY<YUTHEN6550
6540 CU=CU+1:A=BL:GOSUB8100:DY=DY*-1
6550 RETURN
7000 GOSUB8600:A=R
7010 FORK=1TOLX:GOSUB8100:NEXTK
7020 A=U:FORK=1TOLY:GOSUB8100:NEXTK
7030 RETURN
7500 PRINT"DAS WAR'S !":END
8100 PRINTCHR$(A);
8110 RETURN
8200 PRINTCHR$(A);CHR$(B);
8210 RETURN
8300 PRINTCHR$(A);CHR$(B);CHR$(C);
8310 RETURN
8400 A=19
8410 GOSUB8100
8420 RETURN
8600 GOSUB8400
8610 A=17
8620 FORK=1TO24:GOSUB8100:NEXTK
8630 RETURN
8800 A=147
8810 GOSUB8100
8820 RETURN

```

8900 FORK=1T01000:NEXTK  
8910 RETURN  
9000 R=29:L=157:U=145:D=17  
9010 X=S2:QH=215:AS=209:SP=32:BL=0:N=0:PL=43  
9020 RETURN

## Flip-Flop

Ziel dieses Spiels ist es, die Steine des Gegners durch die eigenen zu ersetzen. Steine, die gerade von einem Spieler ausgetauscht wurden, dürfen nicht unmittelbar erneut ausgetauscht werden. Sieger ist der Spieler, dem es zuerst gelingt, eine Reihe oder eine Spalte vollständig mit seinen Steinen zu belegen.

```
5 REM FLIP-FLOP
10 POKE59468,12:DIMA%(8,8),B%(8,8),RW(8),RS(8)
20 Z=215:T$(0)="■":T$(1)="□":T=S0
30 FORI=1TO8:A$(I)=CHR$(I+64)
40 FORJ=1TO8
50 IF(I+J)AND1THENA%(I,J)=-1:GOTO70
60 A%(I,J)=0
70 NEXT:NEXT
80 GOSUB10000
90 PRINT"0"TAB(9)"F L I P - F L O P"
100 Y$=" | | | | | | | | !":Z$=" | | | | | | | |"
110 PRINTTAB(10)"1 2 3 4 5 6 7 8"
120 PRINTTAB(9)" | | | | | | | |"
130 FORI=1TO7:PRINTTAB(8)A$(I)Y$:PRINTTAB(9)Z$:NEXT
140 PRINTTAB(8)A$(I)Y$:PRINTTAB(9)" | | | | | | | |"
150 GOSUB500:IFT$="Y"THEN240
170 S=INT(8*RND(TI)+1):W=INT(8*RND(TI)+1):IFAZ(W,S)=0THEN170
180 GOTO200
190 GOTO6000
200 IFU=WANDV=STHEN190
210 PRINTTAB(27)"MEIN ZUG : ";A$(W);RIGHT$(STR$(S),1):U=W:V=S
220 GOSUB600:GOSUB900:XT=XT+1:GOSUB500
230 IFSP=1ORCB=1THENCE=1:GOTO1100
240 GOSUB700
250 IFAZ(W,S)=-1THENPRINT"II":GOTO240
260 IFU=WANDV=STHENPRINT"II":GOTO240
270 U=W:V=S
280 GOSUB600:GOSUB900:XT=XT+1:GOSUB500
290 IFSP=1ORCB=1THENCE=0:GOTO1000
300 GOTO190
499 REM***PRINT-ROUTINE
```

```

500 PRINT"NOCH ZUG: NOCH NEIN"XT
510 FORI=1TO8:PRINTTAB(10);WZ(I)=0:SZ(I)=0
520 FORJ=1TO8:PRINTCHR$(Z+6*WZ(I,J))"II";BZ(I,J)=WZ(I,J):NEXT
530 PRINT:PRINT:NEXT:RETURN
599 REM *** INVERS-ROUTINE
600 FORI=1TO8:WZ(I,S)=NOT(WZ(I,S))
610 WZ(W,I)=NOT(WZ(W,I)):NEXT
620 WZ(W,S)=NOT(WZ(W,S)):RETURN
699 REM *** SPIELZUEGE
700 PRINT"NOCH EIN ZUG: NEIN"
710 GOSUBS00
720 IFT$(CHR$(65)ORT$(72))THEN710
730 W=ASC(T$)-64:PRINTT$;
740 GOSUBS00
750 IFT$(CHR$(20))THENPRINT"II II";GOTO710
760 IFT$(CHR$(49)ORT$(56))THEN740
770 PRINTT$;S=VAL(T$):IFWZ(W,S)=-1THENPRINT"TT":GOTO700
780 IFU=WRANDV=S THENPRINT"TT":GOTO700
790 PRINTTAB(37)" ":RETURN
799 REM *** EINGABE SPIELER
800 OPEN1,0,0
810 GETT$:PRINTT$(ABS(X));FORI=1TOT:NEXT:X=NOT(X):IFT$=""THENS10
820 A=ASC(T$):CLOSE1:RETURN
899 REM *** BERECHNUNG DER PUNKTE
900 FORI=1TO8:FORJ=1TO8
910 WZ(I)=WZ(I)+WZ(I,J)
920 SZ(I)=SZ(I)+WZ(J,I)
930 NEXT
940 IFWZ(I)=-8ORSZ(I)=-8THENSP=1
950 IFWZ(I)=0ORSZ(I)=0THENCB=1
960 NEXT:RETURN
1000 PRINT"DU HAST GEWONNEN!!!SPIELST DU NOCH EINMAL? (J/N)":
1010 GOTO110
1100 PRINT"ICH HABE GEWONNEN!!!SPIELST DU NOCH EINMAL? (J/N)":
1110 GETT$:IFT$=""THEN110
1120 IFT$="J"THENCLEAR:PRINT"IIIIIIIIII JA ":GOTO10
1130 IFT$<"N"THEN110
1140 PRINT"NOCH NEIN":END
6000 REM *** BERECHNUNG DER ZUEGE
6010 FORI=1TO8:FORJ=1TO8
6040 ZW(I)=ZW(I)+BZ(I,J)
6050 ZS(I)=ZS(I)+BZ(J,I)
6060 NEXT:NEXT:W0=0:SG=0
6070 Z7=0:Z1=0:FORI=1TO8

```

```

6080 IFZW(I)=-7THENZ7=Z7+1
6090 IFZS(I)=-7THENZ7=Z7+1
6100 IFZW(I)=-1THENZ1=Z1+1:WG=I
6110 IFZS(I)=-1THENZ1=Z1+1:SG=I
6120 ZS(I)=0:ZW(I)=0
6130 NEXT:IFZ7>0ORZ1>0THEN6300
6140 W=INT(8*RND(TI)+1):S=INT(8*RND(TI)+1):IFB%(W,S)=0THEN6140
6150 FORI=1TO8:BX(I,S)=NOT(BX(I,S))
6160 BX(W,I)=NOT(BX(W,I)):NEXT
6170 B%(W,S)=NOT(B%(W,S))
6180 FORI=1TO8:FORJ=1TO8
6190 ZW(I)=ZW(I)+BX(I,J)
6200 ZS(I)=ZS(I)+B%(J,I)
6210 NEXT:NEXT
6220 Z7=0:Z1=0:FORI=1TO8
6230 IFZW(I)<=-7THENZ7=Z7+1
6240 IFZS(I)<=-7THENZ7=Z7+1
6250 IFZW(I)=-1THENZ1=Z1+1
6260 IFZS(I)=-1THENZ1=Z1+1
6270 ZS(I)=0:ZW(I)=0:NEXT
6281 FORI=1TO8:BX(I,S)=NOT(BX(I,S))
6282 BX(W,I)=NOT(BX(W,I)):NEXT
6283 B%(W,S)=NOT(B%(W,S))
6284 IFZ7<>0THEN6140
6290 GOTO200
6300 IFZ1=0THEN6140
6400 IFWG=0THEN6450
6410 FORI=1TO8:IFB%(WG,I)=-1THENS=I
6415 NEXT
6420 FORI=1TO8:IFI=WGTHEN6440
6430 IFB%(I,SG)=-1THENW=I:S=SG
6440 NEXT:GOTO200
6450 FORI=1TO8:IFB%(I,SG)=-1THENWG=I
6455 NEXT
6460 FORI=1TO8:IFI=SGTHEN6430
6470 IFB%(WG,I)=-1THENW=WG:S=I
6480 NEXT:GOTO200
9999 REM *** ANFANGSBILD
10000 PRINT"DN"TAB(6)" "
10010 PRINTTAB(5)"  F L I P - F L O P  "
10020 PRINTTAB(6)" "
10030 PRINT"DU SPIELST GEGEN DEN"
10040 PRINT"COMMODORE 64"
10050 PRINT"KENNST DU DIE SPIELREGELN? (J/N)"

```

```

10051 GETT$:IFT$=""THEN10051
10052 IFT$="N"THEN30000
10053 IFT$<>"J"THEN10051
10055 PRINT"DU BEKOMMST DIE WEISSEN STEINE(0)"
10060 PRINT"BIST DU FERTIG ? (J,N)"
10061 GETT$:IFT$=""THEN10061
10062 RETURN
29999 REM *** RULES
30000 PRINT"J"TAB(12)"REGELN"
30010 PRINT"JEDEDER SPIELER (DU UND DER COMPUTER)"
30020 PRINT"VERSUCHT DIE GEGNERISCHEN STEINE"
30030 PRINT"GEGEN EIGENE AUSZUTAUSSCHEN."
30040 PRINT"DABEI WERDEN ALLE STEINE"
30050 PRINT"VERTIKAL UND HORIZONTAL"
30060 PRINT"ZUM AUSGETAUSCHTEN STEIN EBENFALLS"
30070 PRINT"AUSGETAUSCHT."
30080 PRINT" ( 0 -> 0 AND 0 -> 0 )"
30090 PRINTTAB(31)"_____ "
30100 PRINTTAB(30)"<RETURN>"
30110 GETT$:IFT$=""THEN30110
30120 PRINT"BEISPIEL : "
30130 PRINT"ODER GETAUSCHTE STEIN(PFEIL)UND SEINE"
30140 PRINT"HORIZONTALE UND VERTIKALE REIHE"
30145 PRINTTAB(10)"4"TAB(25)"4"
30150 PRINTTAB(10)"0"TAB(25)"0"
30160 PRINTTAB(10)"0"TAB(25)"0"
30170 PRINTTAB(10)"0"TAB(25)"0"
30180 PRINTTAB(10)"0"TAB(25)"0"
30190 PRINTTAB(6)"E0000000"TAB(21)"E0000000"
30200 PRINTTAB(9)"70"TAB(24)"70"
30210 PRINTTAB(8)"/ 0"TAB(23)"/ 0"
30220 PRINTTAB(6)"E4 0"TAB(21)"E4 0"
30230 PRINT
30240 PRINT
30250 PRINT
30260 PRINTTAB(31)"_____ "
30270 PRINTTAB(30)"<RETURN>"
30280 GETT$:IFT$=""THEN30280
30300 PRINT"JEIN STEIN,DER GERADE VOM GEGNER"
30310 PRINT"AUSGETAUSCHT WURDE,KANN NICHT SOFORT"
30320 PRINT"WIEDER AUSGETAUSCHT WERDEN"
30330 PRINT
30340 PRINT"DESIEGER IST DER SPIELER,DESSER STEINE"
30350 PRINT"ZUERST EINE HORIZONTALE ODER"

```

```
30360 PRINT"VERTIKALE REIHE BILDEN."  
30370 PRINT  
30380 PRINT  
30390 PRINT  
30400 PRINTTAB(31)"XXXXXXXXXX"  
30410 PRINTTAB(30)"<RETURN>"  
30420 GETT#:IFT#=""THEN30420  
30430 GOTO10000
```



## Irrgarten

Der Spieler muß sich möglichst lange im Irrgarten bewegen, ohne anzustoßen. Der bewegte Punkt wird mit den Tasten 8, 2, 4 und 6 gesteuert. Tasten, die eine entgegengesetzte Bewegungsrichtung auslösen, können nicht unmittelbar aufeinander folgend benutzt werden (z.B. 8 und 2 oder 4 und 6).

```
10 PRINT"┌ I R R G A R T E N ┐"
11 PRINT"  ───────────────────"
12 PRINT:PRINT"VERSUCHE DICH SO LANGE WIE MOEGLICH IM"
13 PRINT"IRRGARTEN ZU BEWEGEN, OHNE ANZUECKEN":PRINT
14 PRINT"TASTE  8:OBEN"
15 PRINT"TASTE  2:UNTEN"
16 PRINT"TASTE  4:LINKS"
17 PRINT"TASTE  6:RECHTS"
18 PRINT
20 PRINT"DIE TASTEN 8 UND 2 SOWIE 4 UND 6 KOENNEN"
22 PRINT"NICHT NACHEINANDER GEDRUECKT WERDEN"
25 PRINT
30 PRINT"DRUECKE EINE BELIEBIGE TASTE ZUM WEITER SPIELEN"
35 GETA$:IFA$=""THEN35
40 B=1024:BC=55296
45 U$="2":O$="8":L$="4":R$="6"
50 G=100
100 PRINT"┌"
110 FORY=1TO24:FORI=0TO7
120 X=INT((RND(1)+I)*5)
130 POKEB+40*Y+X,42:POKEBC+40*Y+X,13:NEXT:NEXT
150 FORI=0TO10:GETP$:NEXT
180 X=INT(RND(1)*36)
181 Y=INT(RND(1)*22)+1
182 Z=B+40*Y+X:IFPEEK(Z)<>32THEN180
183 POKEZ,81:POKEBC+40*Y+X,1
195 GETP$
196 IFF$<>U$ANDP$<>O$ANDP$<>L$ANDP$<>R$THEN195
197 T=TI:PA$="":GOTO210
200 GETP$
210 IFF$=U$ANDPA$<>O$THENPA$=P$:YI=1:XI=0
211 IFF$=R$ANDPA$<>L$THENPA$=P$:YI=0:XI=1
```

```

212 IFP#=L#ANDPA#<R#THENPA#=P#:YI=0:XI=-1
213 IFP#=0#ANDPA#<U#THENPA#=P#:YI=-1:XI=0
230 POKEZ,81:X=X+XI:Y=Y+YI
235 G=G*0.96:X0=TI+G
236 IFX0>TITHEN236
240 IFX<00RX>390RY<00RY>24THEN300
250 Z=B+40*Y+X:ZC=BC+40*Y+X
260 IFPEEK(Z)=32THENPOKEZ,81:POKEZC,1:GOTO200
300 FORI=1TO8:X=TI
301 IFX+8>TITHEN301
303 IFX+9>TITHEN303
304 NEXT
320 PRINT"DEINE ZEIT: ";(TI-T-25)/60;"SEKUNDEN"
322 X=INT(TI-T-25)/60
330 FORI=1TO10:GETA$:NEXT
350 PRINT"NEUES SPIEL (J/N)"
355 GETSS$:IFSS$=""THEN355
360 P=PEEK(515):IFP<22ANDP0=45THEN360
361 IFP=45THEN50
363 IFSS$="J"THEN100
370 PRINT"BIS ZUM NAECHSTEN MAL !"
380 FORI=1TO10:GETA$:NEXT

```

## Angriff

Das Programm läßt Ihnen die Wahl, ob Sie es als Angreifer oder Verteidiger spielen wollen. Verteidigt wird ein Schloß. Der Angreifer bewegt sich von Feld zu Feld. Ein einmal beschrittener Weg zwischen Feldern wird blockiert und ist deshalb nicht mehr benutzbar. Der Verteidiger kann sich nur zwischen den drei Punkten A, B und C hin- und herbewegen. Der Angreifer muß versuchen, an dem Punkt in das Schloß einzudringen, an dem sich der Verteidiger gerade befindet. Gelingt ihm das, hat er gewonnen. Beide Spieler führen laufend Bewegungen durch. Das Spiel kann mit der S-Taste beendet werden.

```
10 REM ANGRIFF
15 GOTO8000
20 FORI=1TO1700:NEXT
21 PRINTB$;C$:PRINTC$:PRINTC$:RETURN
30 GETA$:IFA$=""THEN30
31 RETURN
100 CLR
110 PRINT"BITTE WARTEN!"
120 DIMA(15,15),F(18)
130 Z1=209:Z2=170:Z3=102
140 B$="XXXXXXXXXXXXXXXXXXXX"
141 C$="
145 W$="ABCBCARBBBCACBA"
150 FORI=1TO15
160 N=4:IFI=7ORI=STHENN=5
170 IFI=12ORI=15THENN=3
175 IFI=13ORI=14THENN=2
180 FORJ=1TON:READX:R(I,X)=1
190 NEXTJ:NEXTI
200 FORI=1TO18
220 READY:F(I)=Y
230 NEXTI
240 I1=PEEK(144):I2=PEEK(145)
300 GOSUB10000:X=1:XI=16
310 POKE1024+F(X),Z1:POKE1024+F(X),Z2:POKE55296+F(X),I3:POKE55296+F(X),I4
315 IFD=1THENSM=1:GOTO3000
320 PRINTB$;"ANGRIFF (A) OR VERTEIDIGUNG (B)?:GOSUB30
```

```

330 IFA$="D"THENSM=1:GOTO3000
340 GOSUB21:GOTO1030
1000 GOSUB20
1030 PRINTB$;"ANGREIFER AUF FELD";X$:INPUT"II: ZIEHT NACH";Y$:IFY$="S"THEN6000
1040 Y=VAL(Y$)
1041 IFX<12ANDY=0THENPRINT"JOKER !":GOTO1000
1042 IFY=0THEN6000
1045 IFY>15THENPRINT"FELD EXISTIERT NICHT!":GOTO1000
1050 IFA(X,Y)=0THENPRINT"NICHT MOEGLICH!":GOTO1000
1060 IFA(X,Y)=2ORR(X,X)=2THENPRINT"DIESER WEG IST NICHT FREI!":GOTO1000
1070 GOSUB7000:IFSM=1THEN4000
2000 Y1=ASC(MID$(W$,X,1))-49:GOTO2060
2050 Y1=RND(-TI):Y1=INT(RND(1)*3+1)+15
2060 IFY1=X1THEN2050
2070 PRINTB$;"VERTEIDIGER ZIEHT NACH";" "CHR$(Y1+49)" ";
2075 GOSUB20
2080 POKE1024+F(X1),160:POKE1024+F(Y1),22
2085 POKE55296+F(X1),4:POKE55296+F(Y1),4
2090 X1=Y1
2095 IFSM=0THEN1030
3000 GOSUB21
3010 IFZ=0THENY=INT(RND(1)*4+2):GOTO1070
3030 M=0:FORI=1TO15:IFA(X,I)=1THENM=M+1:M(M)=I
3050 NEXT
3060 IFM=0THEN6000
3070 IFX<12THEN3210
3140 IFX1=16THENIFX=13ORX=14THENY=X1
3142 IFX1=17THENIFX=12ORX=13THENY=X1
3144 IFX1=18THENIFX=14ORX=15THENY=X1
3200 IFY>15THEN7000
3210 FORI=MT01STEP-1
3215 IFPEEK(33019)=81ORPEEK(33031)=81THENY=M(1)
3220 IFASC(MID$(W$,M(I),1))-49=X1THENY=M(I):GOTO1070
3230 NEXT
3234 IFM<3THENY=M(1)
3240 Y=M(INT(RND(1)*M+1)):GOTO1070
4000 IFD=1THEN2000
4010 PRINTB$;"SIE SIND DRAN:"
4020 GOSUB30:Y1=ASC(A$)-49
4022 IFY1=34THEN6000
4025 IFY1<16ORV1>18THENPRINT"FLUECHTEN SIE ETWA?":GOTO4040
4030 IFX1=Y1THENPRINT"SIE MUESSEN ZIEHEN !!!":GOTO4040
4035 GOTO2080
4040 GOSUB20:GOTO4010

```

```

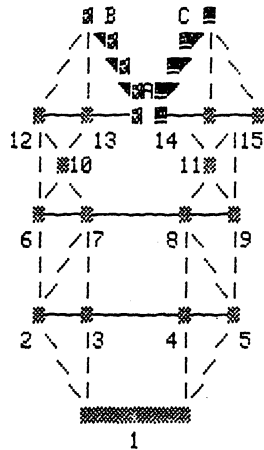
6000 GOSUB21:IFY$="S"THENA$=Y$
6004 IFA$="S"THENPRINT$;"SIE HATTEN KEINE CHANCE ":GOTO6200
6005 IFM=0ANDSM=1THENPRINT$;"DAS WAR EIN FEHLER!":GOTO6200
6010 X=ASC(Y$)-49
6020 IFX=X1THEN6100
6030 PRINT$;"HA-HA!!! SIE HABEN VERLOREN!":GOTO6200
6100 IFSM=1THENPRINT$;"MICH HABE WIE IMMER GEWONNEN":GOTO6200
6110 PRINT$;"GLUECKWUNSCHESIE HABEN GEWONNEN"
6200 PRINT"NOCH EIN SPIEL?(J/N)":GOSUB30
6210 IFA$="J"THEN100
6500 END
7000 IFSM=0THEN7010
7005 PRINT$;"ANGREIFER ZIEHT VON";X;"NACH";
7006 IFY<16THENPRINTY:GOTO7010
7008 PRINTCHR$(Y+49)
7010 H=X:FORI=1TO6
7020 IFI>3THENH=Y
7030 POKE1024+F(H),Z1:POKE55296+F(H),13
7040 FORJ=1TO200:NEXT
7050 POKE1024+F(H),Z3:POKE55296+F(H),13
7055 FORJ=1TO200:NEXT:NEXTI
7060 POKE1024+F(H),Z1:POKE55296+F(H),13
7065 IFY=X1THEN6100
7070 A(X,Y)=2:A(Y,X)=2
7071 POKE144,B1:POKE145,D2
7072 FORI=1TO56:READJ,H,B
7073 IFJ=XANDH=YTHENPOKE1024+B,81:POKE55296+B,13:GOTO7080
7074 IFJ=YANDH=XTHENPOKE1024+B,81:POKE55296+B,13:GOTO7080
7075 NEXT
7080 X=Y:Z=Z+1:PRINT"#####";Z
7090 GOSUB21
7100 RETURN
8000 PRINT"Q"
8010 PRINT"EIN DEMONSTRATIONSSPIEL ? (J/N)"
8020 GETA$:IFA$=""THEN8020
8030 IFA$="J"THENCLR:D=1:GOTO110
8040 GOTO100
9000 DATA2,3,4,5,1,3,6,7,1,2,4,7,1,3,5,8
9010 DATA1,4,8,9,2,7,10,12
9020 DATA2,3,6,8,10,4,5,7,9,11
9030 DATA5,8,11,15,6,7,12,13,8,9,14,15
9040 DATA6,10,13,10,12
9050 DATA11,15,9,11,14
9100 DATA737,569,573,581,585,409,413,421,425,331

```

```

9105 DATA343,249,253,261,265,257,93,101
9200 DATA1,2,651,1,3,653,1,4,661,1,5,663
9212 DATA2,3,571,3,4,577,4,5,583,2,6,489,2,7,491
9214 DATA3,7,493,4,8,501,5,8,503,5,9,505
9216 DATA6,7,411,7,8,417,8,9,423
9218 DATA6,10,370,7,10,372,8,11,382,9,11,384,6,12,329,9,15,345,11,8,382
9220 DATA10,12,290,10,13,292,11,14,302,11,15,304
9222 DATA12,13,251,14,15,263
9224 DATA 2,1,651,3,1,653,4,1,661,5,1,663
9226 DATA3,2,571,4,3,577,5,4,583,6,2,489,7,2,491
9228 DATA7,3,493,8,4,501,8,5,503,9,5,505
9230 DATA7,6,411,8,7,417,9,8,423,10,6,370
9232 DATA10,7,372,11,8,382,11,9,384,12,6,329,15,9,345
9234 DATA12,10,290,13,10,292,14,11,302,15,11,304
9236 DATA13,12,251,15,14,263
10000 PRINT"ANGRIF"
10011 PRINT"
10012 PRINT"
10013 PRINT"
10014 PRINT"
10015 PRINT"
10016 PRINT"
10017 PRINT"
10018 PRINT"
10019 PRINT"
10020 PRINT"
10021 PRINT"
10022 PRINT"
10023 PRINT"
10024 PRINT"
10025 PRINT"
10026 PRINT"
10027 PRINT"
10028 PRINT"
10029 PRINT"
10030 RETURN

```



## Artillerie

Zwei Spieler schießen über einen Berg hinweg aufeinander. Jeder der Spieler gibt für jeden Schuß den Abschlußwinkel und die Abschlußkraft ein. Beide müssen den Einfluß des ständig Stärke und Richtung wechselnden Windes berücksichtigen.

```
5 REM ARTILLERIE
10 DIMA(100),B(100),H(39),A1(2),B1(2),A2(2),B2(2),L(2)
15 PRINT"J"
20 PRINT"ANWEISUNGEN? (J/N)";GET D$:IF D$="N"THEN41
21 IF D$<>"J"THEN 20
30 PRINT"J","ARTILLERIE-SPIEL"
31 PRINT:PRINT"ZWEI SPIELER SCHIESSEN UEBER EINEN ";
32 PRINT"BERG HINWEG AUF EINANDER"
33 PRINT:PRINT"FUER JEDEN SCHUSS GEBEN SIE DEN"
34 PRINT" SCHUSSWINKEL (0-90) UND DIE"
35 PRINT" KRAFT (5-10) EIN.":PRINT
36 PRINT"BEACHTEN SIE AUCH DEN WIND !!"
37 PRINT:PRINT"MIT DEM SCHUSSWINKEL 0 ERHALTEN SIE EINEN NEUEN BERG"
39 PRINT:PRINT"SPIELBEGINN --> TASTE DRUECKEN"
40 GET G$:IF G$="" THEN 40
41 PRINT"J"
50 Z$="J":FORX=1TO5:Z$=Z$+Z$:NEXTX
60 A$="J":FORX=1TO6:A$=A$+A$:NEXTX:B$="J":FORX=1TO5:B$=B$+B$:NEXTX
80 N=2
90 REM
130 X1=INT(6+RND(1)*8)
140 X2=INT(25+RND(1)*8)
150 L(1)=INT(3+(X1-3)*RND(1))
160 L(2)=INT(X2+1+(36-X2)*RND(1))
170 H9=RND(1)*11+1
180 FOR I=1 TO X1:H(I)=H9:NEXTI
210 H9=RND(1)*11+1
220 FOR I=X2TO39:H(I)=H9:NEXTI
250 IFABS(H(1)-H(39))<4THEN170
260 X3=X1+INT(RND(1)*(X2-X1-6))+3
270 H(X3)=RND(1)*11+9
290 D1=(H(X3)-H(1))/2
300 D2=(H(X3)-H(39))/2
310 IFD1*D2<=0THEN470
```

```

320 J=-180
330 K=180/(X3-X1+1)
340 H9=H(1)+D1
350 FORI=X1+1TOX3-1
360 J=J+K
370 H(I)=H9+COS(J*π/180)*D1
380 NEXT I
390 J=0
400 K=180/(X2-X3+1)
410 H9=H(39)+D2
420 FORI=X3+1TOX2-1
430 J=J+K
440 H(I)=H9+COS(J*π/180)*D2
450 NEXT I
460 GOTO550
470 D=(H(1)-H(39))/2
480 J=0
490 K=180/(X2-X1+1)
500 H9=H(39)+D
510 FORI=X1+1TOX2-1
520 J=J+K
530 H(I)=H9+COS(J*π/180)*D
540 NEXT I
550 PRINT"圖1";
560 W=INT(100*(RND(1)-.5));
570 PRINTLEFT$(B$,22);
580 FORI=1TO39:H(I)=INT(H(I)+.5):NEXT I
590 FORX=1TO39:IFX=1THEN610
600 PRINT"■";
610 PRINTMID$(Z$,1,H(X)*4);
620 FORZ=1TOH(X):PRINT"■":NEXTZ
630 NEXTX:PRINT"■"
640 PRINT"■";
650 PRINTMID$(B$,1,22-H(1)):IFL(1)=1THEN700
660 PRINTMID$(A$,1,L(1)-1);
670 PRINT"↑";
680 PRINT"■";
690 PRINTMID$(B$,1,22-H(39));
700 PRINTMID$(A$,1,L(2)-1);
710 PRINT"↑":GET G$:GET G$
720 PRINT"■";
730 PRINTMID$(B$,1,24);
740 PRINT"      "
750 K=1

```

```

790 IFW>0THENS20
800 PRINT-W;" KM/H WIND N. LINKS ";
810 GOTO840
820 PRINTW;" KM/H WIND N. RECHTS ";
830 PRINT" ";
840 POKE54296,15:POKE54273,8:POKE54277,16:POKE54278,32+6
845 PRINT" ";:N=3-N:IFN=1THEN900
850 PRINTLEFT$(A$,28);"WINK.";
860 INPUT E:PRINT" ";
870 PRINTLEFT$(B$,1);LEFT$(A$,28);
875 PRINT "KRAFT";
880 INPUTV:IFV>15THENV=15
890 GOTO910
900 PRINT" ";
902 PRINT" ";
903 PRINT"WINK.";
905 INPUTE:PRINT" ";:INPUTV:IFV>15THENV=15
910 POKE54276,128+1:POKE54276,128
920 K=K+1:IFE=0THEN130
930 A2(N)=A2(N)-2.85
940 E=90-E
950 IFN=1THEN970
960 E=-E
970 X1=L(N)
980 Y1=H(L(N))+1
990 T=0:M=0
1010 V1=SIN(E*π/180)*V
1020 V2=COS(E*π/180)*V*.7
1030 T=T+.5:M=M+1
1040 V1=V1+(V1+W*.5)/1500
1050 X=X1+(V1+W*.5/50)*T
1060 X=INT(X+.5)
1070 Y=Y1+(V2-T/3)*T
1080 Y=INT(Y+.5)
1090 Z=23-Y:PRINT" ";
1092 A(N)=Z:B(N)=X:IFX>39ORX<1THEN1350
1094 IFY-H(X)<=0THEN1340
1100 IFZ<0THEN1150
1110 IFZ=0THEN1125
1120 PRINTLEFT$(B$,Z);
1125 IFX=1THEN1140
1130 PRINTLEFT$(A$,X-1);
1140 PRINT" ";
1150 PRINT" ";

```

```

1160 IFM=4<1THEN1030
1162 IFA(M-4)=0THEN1180
1164 IFA(M-4)<0ORB(M-4)<=0THEN1030
1170 PRINTLEFT$(B$,A(M-4));
1180 PRINTLEFT$(A$,B(M-4)-1);
1190 PRINT " ";
1200 PRINT" ";
1220 GOTO1030
1230 IFM=1THEN1030
1232 IFB(M)<1ORB(M)>39THEN840
1234 IFABS(B(M-1)-L(N))<1THEN1250
1240 GOTO1260
1250 IFT>1THEN1510
1260 IFABS(B(M-1)-L(3-N))<1THEN1560
1270 IFT<1THEN1030
1280 GOTO1460
1290 PRINT"NOCH EIN SPIEL(J/N)";
1310 INPUT G$
1320 IFLEFT$(G$,1)<>"N"THEN90
1330 PRINT"BYE! BYE!";PRINT
1335 END
1340 PRINT" ";
1350 FORJ=4TO1STEP-1
1360 IFM-J<=0THEN1440
1370 IFA(M-J)=0THEN1405
1380 IFA(M-J)<0ORB(M-J)<=0THEN1440
1400 PRINT LEFT$(B$,A(M-J));
1405 IFB(M-J)=1THEN1420
1410 PRINTLEFT$(A$,B(M-J)-1);
1420 PRINT " ";
1430 PRINT" ";
1440 NEXTJ:IFM=1THEN1600
1450 IFB(M)<1ORB(M)>39THEN1610
1455 GOTO1230
1460 IFB(M-1)=1THEN1470
1465 PRINTLEFT$(A$,B(M-1)-1);
1470 PRINTLEFT$(B$,23-H(B(M-1)));
1472 FORZ=1TO3:PRINT"Z?";FORX=1TO5:NEXTX
1474 PRINT"X?";FORX=1TO5:NEXTX:NEXTZ:PRINT" ";
1480 GOTO 1600
1510 PRINT" ";
1520 PRINTLEFT$(A$,L(N)-1);
1530 PRINTLEFT$(B$,24-H(L(N)));
1540 FORZ=1TO10:PRINT"Z?";FORX=1TO5:NEXTX:NEXTZ:PRINT" ";

```

```

1544 PRINT"  " " " " ";
1545 FORX=1TO2:NEXTX:NEXTZ
1546 FORX=1TO50:NEXTX
1550 GOTO1290
1560 PRINT" ";
1570 PRINTLEFT$(A$,L(3-N)-1);
1580 PRINTLEFT$(B$,24-H(L(3-N)-1));
1590 GOTO1540
1600 REM
1610 W=W+INT(10*(RND(1)-.5))
1620 PRINT" ";LEFT$(B$,24);" ";
1630 IFW>=0THENPRINT" " "W;" KM/H WIND N. RECHTS "":GOTO840
1640 PRINT" " "W;" KM/H WIND N. LINKS "":GOTO840

```



## Das magische Quadrat

Sie müssen bei diesem Spiel Zahlen so in eine Position innerhalb eines Quadrates einfügen, daß die Summe aller Zahlen in der jeweiligen Reihe, Spalte und Diagonale jeweils fünfzehn ergibt.

```
100 REM MAGISCHES QUADRAT
110 PRINT"J"
140 CLR:GOSUB600
150 PRINT:INPUT"GIB POSITION UND ZAHL EIN":I,J
160 IF I<1OR I>9OR N<1OR N>9 THEN 180
170 IF A(I)=0 AND B(N)=0 THEN 190
180 PRINT:PRINT"FALSCH EINGABE !...VERSUCH'S NOCHEINMAL":GOTO150
190 A(I)=N:B(N)=1:M=M+1
200 GOSUB600
210 GOSUB450
220 IF W=0 THEN 240
230 PRINT:PRINT"BEDAUERE,DU HAST VERLOREN !":GOTO420
240 IF M<5 THEN 260
250 PRINT"FFFFF":GOTO420
260 FOR Q=1 TO 9
270 IF A(Q)>0 THEN 350
280 FOR R=1 TO 9
290 IF B(R)>0 THEN 340
300 A(Q)=R
310 GOSUB450
320 IF W=0 THEN 370
330 Q1=Q:R1=R:W=0:A(Q)=0
340 NEXT R
350 NEXT Q
360 W=1:R=R1:Q=Q1:A(Q)=R
370 B(R)=1
380 PRINT:PRINT"ICH SETZE"R"IN POSITION"Q
390 GOSUB600
400 IF W=0 THEN 150
410 PRINT"ICH HABE VERLOREN ----- DU HAST GEWONNEN"
420 INPUT"WILLST DU WEITERSPIELEN ? (J/N)":JY$
430 IF Y$="J" THEN 140
440 END
450 FOR X=1 TO 6
```

```
460 ONXGOTO470,480,490,500,510,520,530
470 J=1:K=2:L=3:GOTO540
480 K=4:L=7:GOTO540
490 K=5:L=9:GOTO540
500 J=4:L=6:GOTO540
510 J=3:L=7:GOTO540
520 K=6:L=9:GOTO540
530 J=7:K=8:GOTO540
540 IFA(J)=0ORR(K)=0ORR(L)=0THEN560
550 IFA(J)+R(K)+R(L)>15THEN580
560 NEXTX
570 GOTO590
580 W=1
590 RETURN
600 PRINT:PRINTTAB(25)R(1);R(2);R(3)
610 PRINTTAB(25)R(4);R(5);R(6)
620 PRINTTAB(25)R(7);R(8);R(9)
630 PRINT
640 RETURN
```

## Karussell

Sie erhalten eine Reihe von Zahlen (wieviele ist in Zeile 40 definierbar), die Sie in numerischen Reihenfolge ordnen müssen, um zu gewinnen. Sie müssen dazu angeben, wieviele Zahlen zu vertauschen sind. Hierzu ein Beispiel:

Ursprüngliche Zahlenliste:

4 5 3 2 1 6 7

Geordnete Zahlenliste:

1 2 3 4 5 6 7

Das gerade gezeigte Ergebnis erhalten Sie, indem Sie auf die Frage "WIEVIELE ZAHLEN SOLL ICH VERTAUSCHEN?" mit '5' antworten.

```
10 REM KARUSSELL
20 PRINT"?"
30 DIMA(20)
40 N=7:REM KANN GERENDERT WERDEN
50 A(1)=INT((N-1)*RND(1))+2
60 FORK=2TON
70 A(K)=INT(N*RND(1))+1
80 FORL=1TOK-1
90 IFA(K)=A(L)THEN?0
100 NEXTL
110 NEXTK
120 PRINT
130 PRINT"HIER DIE ZAHLENLISTE:"
140 PRINT
150 T=0
160 GOSUB360
170 PRINT
180 INPUT"WIEVIELE SOLL ICH VERTAUSCHEN ":R
190 IFR<=NTHEN220
200 PRINT:PRINT"MAXIMAL SIND";N;"ZU VERTAUSCHEN !!!"
210 GOTO170
```

```
220 T=T+1
230 FORK=1TOINT(R/2)
240 Z=A(K)
250 A(K)=A(R-K+1)
260 A(R-K+1)=Z
270 NEXTK
280 GOSUB360
290 FORK=1TON
300 IFA(K)<OKTHEN170
310 NEXTK
320 PRINT:PRINT:PRINT
330 PRINT"SIE BENOETIGTEN"TZUEGE !"
340 PRINT:PRINT:PRINT
350 END
360 PRINT
370 FORL=1TON
380 PRINTA(L);
390 NEXTL
400 PRINT
410 PRINT
420 RETURN
```

## Die Entfernung zwischen zwei Städten

Mit diesem Programm können Sie die Entfernung zwischen zwei Städten, unabhängig von ihrer Lage, berechnen. Sie müssen dazu nur den Längen- und den Breitengrad der Städte wissen. Für einige bekannte Großstädte enthält das Programm die notwendigen Daten. Das Programm zeigt die Lage der von Ihnen ausgewählten Städte auf einer schematischen Darstellung einer Weltkarte an.

```
4 DIMW$(18)
5 PRINT"Q"
10 PRINT" ENTFERNUNGSBERECHNUNG "
15 PRINT
20 PRINT" KOORDINATEN KOENNEN AB ZEILE"
22 PRINT" 2000 ALS DATAANWEISUNG STEHEN"
23 PRINT"FORMAT:DATA NAME,BREITENGRAD,HOEHENGRAD"
25 FORI=1TO2000:NEXT
26 PRINT"Q":GOSUB1100
27 Q=1024
50 PRINT:PRINT"BERECHNUNG NACH KOORDINATEN"
51 PRINT"STADTELISTE ANZEIGEN DER LISTE"
52 GETA$
53 IFA$="B"THEN99
54 IFA$="L"THEN1000
55 IFA$="A"THENLIST2000-
59 GOT052
99 PRINT"Q"
100 PRINT:PRINT"ENTFERNUNGSBERECHNUNG ZWISCHEN STADTEN"
102 PRINT" (EINGABE DER KOORDINATEN"
103 PRINT" FORMAT:GRAD,MINUTEN/ WEST UND"
104 PRINT" SUEO NEGATIV)"
110 PRINT"*"
111 PRINT"NAME DER ERSTEN STADT "
113 INPUTN1$
115 INPUT"GE0GR. LAENGE")L
120 GOSUB500
122 LI=0:LS=0R
125 INPUT"GE0GR.HOEHE ")L
128 GOSUB500
```

```

130 B1=G: B5=GR
135 PRINT"NAME DER ZWEITEN STADT"
137 INPUTN2$
140 INPUT"GEOGR. LAENGE":L
143 GOSUB500
145 L2=G: L6=GR
148 INPUT"GEOGR. HOEHE ":L
150 GOSUB500
152 B2=G: B6=GR
160 V=SIN(B1)*SIN(B2)+COS(B1)*COS(B2)*COS(L2-L1)
161 IFV=0THENEM=PI/2:GOTO166
162 EM=ATN(SQR(1-V*V)/V)
164 IFEM<0THENEM=EM+PI
166 EM=EM*(180/PI)*60
167 EK=EM*1.852
168 EM=INT(EM*1000+.5)/1000:EK=INT(EK*1000+.5)/1000
169 PRINT"J":PRINT:PRINT:PRINT:PRINT
170 PRINT"DIE ENTFERNUNG VON ";N1$
175 PRINT"NACH ";N2$;" BETRAEGT "
178 PRINTEM;" MEILEN, ODER"
180 PRINTEK;" KILOMETER "
181 FORI=1TO3000:NEXT
182 B3=INT((90-B5)/10)+1
183 B4=INT((90-B6)/10)+1
190 G1=INT((L5+180)/10)+B3*40
192 G2=INT((L6+180)/10)+B4*40
200 PRINT"J":GOSUB1150
202 EK=INT(EK)
210 PRINT:PRINT"ENTFERNUNG VON ";N1$;" NACH ";N2$;" : ":PRINTEK;" KILOMETER"
212 H1=PEEK(Q+G1):H2=PEEK(Q+G2)
213 H3=ASC(LEFT$(N1$,1))-64:H4=ASC(LEFT$(N2$,1))-64
214 IFG1=G2THEN224
215 FORI=1TO20
217 POKEQ+G1,H1:POKEQ+G2,H2
219 FORJ=1TO100:NEXT
220 POKEQ+G1,H3:POKEQ+G2,H4
222 FORJ=1TO500:NEXT
223 NEXT:GOTO230
224 FORI=1TO20
225 POKEQ+G1,H1:FORJ=1TO150:NEXT
226 POKEQ+G1,H3:FORJ=1TO500:NEXT
227 POKEQ+G2,H2:FORJ=1TO150:NEXT
228 POKEQ+G2,H4:FORJ=1TO500:NEXT
229 NEXT

```

```

230 PRINT"          WEITER --> LEERTASTE DRUECKEN!"
240 GETA$:IFA#<>" THEN240
250 GOT050
500 Z=1
502 IFL<0THENZ=-1:L=ABS(L)
505 GR=(L-INT(L))*10/6+INT(L)
510 G=π/180*GR#Z
512 GR=GR#Z
515 RETURN
1000 PRINT"J"
1010 RESTORE
1012 INPUT"NAME DER ERSTEN STADT ";N$
1020 READB$,LA,BA
1022 IFB#="END"THEN1050
1024 IFB#N$THEN1030
1025 B#=""
1029 GOT01020
1030 PRINT:GOT01060
1045 GOT01036
1050 PRINT"DIE STADT IST NICHT IN DER LISTE"
1055 GOT050
1060 N1#N$
1062 L=LA:GOSUB500
1064 L1=G:L5=GR
1065 L=BA:GOSUB500
1067 B1=G:B5=GR
1070 RESTORE
1075 INPUT"NAME DER ZWEITEN STADT ";N2$
1080 READB$,LA,BA
1083 IFB#="END"THEN1050
1085 IFB#N2$THEN1087
1086 GOT01080
1087 PRINT
1090 L=LA:GOSUB500
1091 L2=G:L6=GR
1092 L=BA:GOSUB500
1093 B2=G:B6=GR:GOT0160
1100 W$(1)="
1102 W$(2)="
1104 W$(3)="
1106 W$(4)="
1108 W$(5)="
1110 W$(6)="
1112 W$(7)="

```

```

1114 W$(8)="
1116 W$(9)="
1118 W$(10)="
1120 W$(11)="
1122 W$(12)="
1124 W$(13)="
1126 W$(14)="
1127 W$(15)="
1128 W$(16)=" "
1130 W$(17)=" "
1132 W$(18)="
1150 FOR I=1 TO 18
1151 PRINT W$(I)
1154 NEXT
1195 RETURN
2000 DATA CHICAGO,-87.36,41.50
2002 DATA ST. LOUIS,-90.12,38.38
2006 DATA NEW ORLEANS,-90.04,29.56
2008 DATA DUESSELDORF,6.24,51.14
2010 DATA FRANKFURT,8.42,50.08
2014 DATA KAISERSLAUTERN,7.46,49.25
2016 DATA MUENCHEN,11.34,48.09
2018 DATA BAD REICHENHALL,12.52,47.38
2020 DATA COLOMBO,80.7,08
2022 DATA SINGAPUR,103.5,1.25
2024 DATA JAKARTA,106.30,-6.17
2026 DATA CELEBES,120,-3.25
2028 DATA MAILAND,9.07,45.27
2030 DATA BERLIN,13.2,52.3
2032 DATA BERN,7.26,46.54
2034 DATA WIEN,16.24,48.2
2036 DATA ROM,12.2,41.55
2038 DATA KAIRO,30.15,30
2040 DATA MELBOURNE,145,-32.1
2042 DATA PEKING,114.2,40
2044 DATA MOSKOU,37.15,56.3
2046 DATA RIO DE JANEIRO,-43.5,-23
10000 DATA "END",0,0

```

## Kurven

Mit diesem Programm können Sie Kurven auf den Bildschirm zeichnen, die durch eine mathematische Funktion definiert sind. Als Beispiel zeichnet das Programm eine Sinus-Kurve.

```
5 REM KURVEN
10 PRINT"┐";"@ "
15 PRINT"PROGRAMM ZUM ZEICHNEN VON FUNKTIONEN"
20 PRINT
26 PRINT:PRINT
30 PRINT"WENN '0' IN DER OBEREN LINKEN ECKE"
35 PRINT"ERSCHEINT, KOENNEN SIE DURCH DAS"
40 PRINT"DRUECKEN EINER BELIEBIGEN TASTE"
42 PRINT"FORTFAHREN"
43 GETM$:IFM$=" "GOTO43
45 PRINT"┐":PRINT"XXXXXXXXXX"
50 PRINT"└";"WRITE TO 300: DEFFNA(X)=..."
60 PRINT"└";"WRITE TO 310: DEFFNB(X)=..."
70 PRINT:PRINT:PRINT"300DEFFNA(X)=0":PRINT:PRINT"310DEFFNB(X)=0"
75 PRINT"RUN100"
80 PRINT"TTTTTTTT":STOP
85 PRINT"NEUER STARTWERT (J/N) ?"
90 GETL$:IFL$="J"THEN105
95 IFL$="N"THENEND
97 GOTO90
100 DIMU(39),V(39),W(39)
105 PRINT"┐":INPUT"STARTWERT FUER X";A
110 INPUT "ENDWERT FUER X";B
120 INPUT"VERGROESSERUNGSFAKTOR";C
130 INPUT"X-NIVEAU";D
140 E=38*(-A)/(B-A)+1
150 D=D+1
200 PRINT"┐"
205 IFE<00RE>38THEN220
210 FORF=1TO24:PRINTTAB(E);" |":NEXTF
220 PRINT"└"
225 IFD<00RD>24THEN300
230 FORG=1TOD:PRINT:NEXTG
240 PRINT"-----"
```

```

300 DEFFNA(X)=0
310 DEFFNB(X)=0
320 H=(B-A)/38:L=0
330 FORX=ATOB STEPH
340 J=D-INT(FNB(X)*C+0.5):L=L+1
345 U(L)=INT(FNA(X)*1000000)/1000000
350 PRINT"@";
355 IFJ>200RJ<0GOTO375
360 FORK=0TOJ:PRINT:NEXTK
370 PRINTTAB(L);"0"
375 V(L)=INT(FNB(X)*1000000)/1000000:W(L)=INT(X*1000)/1000
380 M=D-INT(FNA(X)*C+0.5)
390 PRINT"@";
395 IFM>200RM<0GOTO420
400 FORN=0TOM:PRINT:NEXT
410 PRINTTAB(L);"*"
420 NEXTX
422 PRINT"@";"@"
425 GETR$:IFR$=""GOTO425
427 PRINT"□":PRINT"XXXXXXXXXX"
430 PRINT"WERTETABELLE (J/N)?"
450 GETM$:IFM$="J"GOTO480
460 IFM$="N"GOTO85
470 GOTO450
480 PRINT"□":"@
490 PRINT" X:          FNA(X):  FNB(X):"
500 FORL=1TO19:PRINTW(L),U(L),V(L):NEXTL
510 GETM$:IFM$=""THEN510
520 PRINT"□";" X:          FNA(X):  FNB(X):"
530 FORL=20TO38:PRINTW(L),U(L),V(L):NEXTL
550 GOTO85

```





```

1210 Z$="          DAS NICHT VIEL          ":RETURN
1300 A$="    IN AFRIKA BRAEUCHTEN SIE DAFUER  "
1310 Z$="          5 STUNDEN JEDE NACHT!    ":RETURN
1400 A$="          ZUFRIEDENSTELLEND      "
1410 Z$="SIE SIND NOCH ANFAENGER!":RETURN
1500 A$="    GAR NICHT SCHLECHT!"
1510 Z$="    GUT GEMACHT. ":RETURN
1600 A$="          S E H R   G U T          "
1610 Z$=" SIE SIND EIN PROF! ":RETURN
1700 A$=" SIE FACKELN NICHT LANGE "
1710 Z$="          !":RETURN
1800 A$="GIGANTISCH! BOMBASTISCH! GRANDIOS"
1810 Z$=" SIE ERHALTEN HIERMIT EIN DIPLOM   ":K=9:RETURN
1900 FORK=1T03000:NEXT:A$="          *****
1910 Z$=" MIR HAT'S DIE SPRACHE VERSCHLAGENEN ":RETURN
2000 A$="    AUSSERGEWOEHNLICH "
2010 Z$=" ":RETURN
2100 A$=" SIE SIND MIT ABSTAND DER BESTE"
2110 Z$="    MOSQUITO KILLER          ":RETURN

```



## Weltzeit

Mit diesem Programm können Sie sich die mitteleuropäische Zeit sowie die Ortszeit in weiteren zwanzig Zeitzonen anzeigen lassen.

```
5 REM WELTZEIT
7 PRINT"☺":DC=54272
8 GOSUB4000
9 PRINTTAB(8)"WELTZEIT AUF DEM C-64☺":PRINT:PRINT:PRINT:PRINT
10 INPUT"GEBEN SIE DIE ZEIT EIN 6ST":TI#
11 INPUT"WOCHENTAG (MONTAG=1 USW.)":X1
12 INPUT"TAGESDATUM (MM,TT,JJJJ)  ":X3,X2,X9:Y1=X3
13 DIMWO$(7),MO$(12),MO$(12)
14 FORI=1TO7:READWO$(I):NEXT
15 FORI=1TO12:READMO(I),MO$(I):NEXT
16 DATAMO,DI,MI,DO,FR,SA,"☺SO☺",31,JAN,28,FEB,31,MAE,30,APR,31,MAI,30,JUN
17 DATA31,JUL,31,AUG,30,SEP,31,OKT,30,NOV,31,DEZ
19 PRINT"☺"
20 DATA151,143,146,140,132,148,137,141,133,160,160,143,142,160,160,148,136,133
21 DATA160,160,131,182,180
22 DATA1,1,2,1,1,6,1,1,1,1,1,20,1,1,1,1,1,2,1,3,1,1,1,1,1,1,1,1,1,1,20,1,1,1,1
23 DATA1,1,3,2,1,1,1,1,1,1,1,22,1,1,1,1,4,1,1,1,1,1,1,1,1,1,1,1,1,1,1,20,1,1,1,1
24 DATA8,1,1,1,1,1,1,1,2,20,1,1,1,4,1,1,1,1,1,1,2,1,2,2,20,1,1,1,5,1,1,1,1,1,2
25 DATA27,1,5,1,1,1,1,1,7,22,1,1,1,5,1,1,6,1,2,1,20,1,1,1,6,1,8,23,1,1,6,1,2,4
26 DATA1,1,1,22,1,1,6,2,5,1,1,1,22,1,15,1,2,22,17,64,4,1,1,1,1,1,1,1,1,1,1,25
27 DATA1,1,1,1,1,1,1,1,1,1,1,1,1,1,25,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0
28 DATA160,160,141,133,154,160,160,160,160,160,160,160,160,160,160,160,160,160
29 DATA160,160,160,160,160
95 REM
96 REM RAND
97 REM
100 POKE1024,79:POKE55296,4
105 FORP=1025TO1062:POKEP,99:POKEP+DC,4:NEXT
110 POKE1063,80:POKE1063+DC,4
115 FORP=1103TO1983STEP40:POKEP,103:POKEP+DC,4:NEXT
120 POKE2023,122:POKE2023+DC,4
125 FORP=2022TO1985STEP-1:POKEP,100:POKEP+DC,4:NEXT
130 POKE1984,76:POKE1984+DC,4
135 FORP=1944TO1064STEP-40:POKEP,101:POKEP+DC,4:NEXT
136 POKE1066,45:POKE1101,43:POKE1066+DC,4:POKE1101+DC,4
```

```

140 FORP=1072T01094:READL:POKEP,A:POKEP+DC,4:NEXT
141 REM
142 REM WELTKARTE
143 REM
144 P=1151
145 READL:IFL=0THEN155
150 P=P+L:POKEP,42:POKEP+DC,13:GOTO145
151 REM
152 REM UHR
153 REM
155 FORP=1294T01301:READL:POKEP,L:POKEP+DC,7:NEXT
160 FORP=1341T01421STEP40:READL:POKEP,L:POKEP+DC,7:NEXT
165 FORP=1461T01454STEP-1:READL:POKEP,L:POKEP+DC,7:NEXT
170 FORP=1414T01294STEP-40:READL:POKEP,L:POKEP+DC,7:NEXT
171 REM
173 REM
175 R=1240:S=81:U=R:V=42
178 PRINT"#####"TAB(30)MO$(X1)". DER"#####"X2"### "MO$(X3)" "
179 PRINT"#####"TAB(30)X9
180 REM
181 PRINT"#####"TAB(31)TI$
182 FORI=1T05:POKER,S:POKEJ,K:FORH=1T010:NEXTH:POKEU,V:POKEC,N
183 POKER+DC,13:POKEJ+DC,13:POKEU+DC,13:POKEC+DC,13:NEXTI
188 IFTI$="000000"THENGOTO3000
189 REMIFTI$="000001"THENTI$="235959"
190 XM=XM+1:IFXM=30THENQ=Q+1:XM=0:GOTO290
199 PRINT"#####"TAB(2)XX$
200 GOTO181
290 IFQ>10THENQX=Q-10
300 ONQGOSUB600,610,620,630,640,650,660,670,680,690
305 ONQXGOSUB700,710,720,730,740,750,760,770,780,790
306 IFQ=20THENQ=0:QX=0
310 GOTO181
350 END
400 REM
410 REM LISTE DER LAENDER
420 REM
600 L$="AEGYPTEN" :T=1:GOSUB1000:J=1359:K=81:C=J:N=42:RETURN
610 L$="AMERIKA OSTKUESTE" :T=6:GOSUB2000:J=1352:C=J:RETURN
620 L$="AMERIKA WESTKUESTE" :T=9:GOSUB2000:J=1349:C=J:RETURN
630 L$="JAPAN" :T=8:GOSUB1000:J=1369:C=J:RETURN
640 L$="UDSSR WESTLICH 172#" :T=11:GOSUB1000:J=1209:C=J:RETURN
650 L$="UDSSR WESTLICH 112#" :T=7:GOSUB1000:J=1243:C=J:RETURN
660 L$="KANARISCHE INSELN" :T=2:GOSUB2000:J=1396:C=J:RETURN

```

```

670 L$="CHINA           ":T=7:GOSUB1000:J=1326:C=J:RETURN
680 L$="MADAGASKAR     ":T=2:GOSUB1000:J=1602:C=J:RETURN
690 L$="AUSTRALIEN     ":T=7:GOSUB1000:J=1609:C=J:RETURN
700 L$="PERU           ":T=6:GOSUB2000:J=1511:C=J:RETURN
710 L$="INDIEN         ":T=4:GOSUB1000:J=1404:C=J:RETURN
720 L$="GUATEMALA     ":T=7:GOSUB2000:J=1431:C=J:RETURN
730 L$="KANADA         ":T=6:GOSUB2000:J=1231:C=J:RETURN
740 L$="ZENTRAL BRASILIEN":T=5:GOSUB2000:J=1594:C=J:RETURN
750 L$="ALASKA         ":T=6:GOSUB2000:J=1188:C=J:RETURN
760 L$="MALAYSIA       ":T=6:GOSUB1000:J=1487:C=J:RETURN
770 L$="SOMALIA        ":T=2:GOSUB1000:J=1362:C=J:RETURN
780 L$="MEXICO WEST..EAST+1":T=8:GOSUB2000:J=1431:C=J:RETURN
790 L$="KOREA          ":T=8:GOSUB1000:J=1327:C=J:RETURN
999 REM
1000 REM BERECHNUNG
1001 REM
1005 Y$=MID$(TI$,3,2)
1010 X=VAL(LEFT$(TI$,2))+T:R$=" H"
1015 IFX>24THENX=X-24:R$=" M"
1020 IFVAL(LEFT$(TI$,2))>22THENX=0:R$=" M"
1025 IFX=24THENX=0:R$="M"
1030 XX$=L$+STR$(X)+" "+Y$+R$+" "
1050 RETURN
1090 END
1999 REM
2000 REM - BERECHNUNGEN
2001 REM
2010 D=VAL(LEFT$(TI$,2))
2020 X=D-T:R$=" H"
2030 IFD=0THENX=24-T:R$=" G"
2040 IFD>0ANDD<T+1THENX=24+D-T:R$=" G"
2044 IFX=24THENX=0:R$="H"
2045 XX$=L$+STR$(X)+" "+Y$+R$+" "
2050 RETURN
2090 END
2900 REM
2950 REM BERECHNUNG DER TAGESANZEIGE
2990 REM
3000 X1=X1+1:IFX1>7THENX1=1
3010 X2=X2+1
3020 IFX2>MO(Y1)THENX2=1:Y1=Y1+1:X3=X3+1:IFY1>12THENY1=1:X3=1:X9=X9+1
3030 REM
3040 REM BERECHNUNG DER SCHALTJAHRE
3050 REM

```

```

3060 IFX9/4=INT(X9/4)ORX9/400=INT(X9/400)THENMO(2)=29
3070 IFX9/4<>INT(X9/4)ORX9/400<>INT(X9/400)THENMO(2)=28
3100 FORI=1TO800:NEXT:GOTO178
4000 REM
4010 PRINTTAB(9)"WORLD TIME AUF DEM C64":PRINT:PRINT:PRINT
4020 PRINT"SIE ERHALTEN DIE MITTELEUROPAEISCHE"
4030 PRINT"ZEIT (MEZ) UND DIE ZEIT IN"
4040 PRINT"ANDEREN ZONEN. DAS DATUM, DER WOCHENTAG UND DAS JAHR"
4050 PRINT"WERDEN ANGEZEIGT":PRINT
4060 PRINT"DAS PROGRAMM ERKENNT AUCH SCHALTJAHRE MIT 29 TAGEN IM"
4070 PRINT" FEBRUAR.":PRINT
4080 PRINT"AM UNTEREN RAND WIRD DIE ZEIT IN 20 VER SCHIEDENEN LAENDERN"
4090 PRINT" AUSGEGEBEN."
4095 PRINT:PRINT"ZEILE 189 OHNE REM BESCHLEUNIGT DIE TAGESANZEIGE"
4100 PRINT:PRINT:PRINT"<LEERTASTE> DRUECKEN."
4200 GETW$:IFW$<>" "THEN4200
4250 PRINT"J"
4300 RETURN

```

## Konzentrationstest

Dieses 'Kartenspiel' testet Ihre Konzentrationsfähigkeit. Sie dürfen nacheinander Karten auswählen. Die mehrfache Wahl derselben Karte führt zu Punktverlusten.

```
10 REM KONZENTRATIONSTEST
14 DIMP(4,13),N$(13),S$(4),T(4),S(13),M$(13),T$(4)
20 PRINT"?"
30 PRINT"WAEHLEN SIE AUS EINEM SPIEL JEWEILS EI-"
40 PRINT"NE KARTE AUS. DER C-64 VERLANGT ENTWEDER"
50 PRINT"EINE FARBE ODER EINEN WERT. KARTEN"
60 PRINT"DUERFEN NUR EINMAL GEWAHLT WERDEN,"
70 PRINT"SONST WERDEN IHNEN PUNKTE ABGEZOGEN!":PRINT
80 PRINT"DAS SPIEL ENDET MIT DER LETZEN KARTE"
90 PRINT"FARBEN: HERZ, PIK, KREUZ UND KARO"
100 PRINT"WERTE: ASS, KOENIG/IN, BUBE UND ZEHN - "
110 PRINT"ZWEI"
120 PRINT
130 PRINT
150 FORI=1TO4
160 FORJ=1TO13
170 P(I,J)=0
180 NEXTJ
190 NEXTI
200 FORI=1TO13
210 READN$(I),M$(I)
220 S(I)=0
230 NEXTI
240 FORI=1TO4
250 READS$(I),T$(I)
260 T(I)=0
270 NEXTI
280 DATAASS,EIN ASS,ZWEI,EINE ZWEI,DREI,EINE DREI,VIER
290 DATAEINE VIER,FUENF,EINE FUENF,SECHS,EINE SECHS,SIEBEN,EINE SIEBEN
300 DATAACHT,EINE ACHT,NEUN,EINE NEUN,ZEHN,EINE ZEHN,BUBE
310 DATAEINEN BUBEN,KOENIGIN,EINE KOENIGIN,KOENIG,EINEN KOENIG
320 DATAKARO,EINE KAROKARTE,PIK,EINE PIKKARTE
330 DATAHERZ,EINE HERZKARTE,KREUZ,EINE KREUZKARTE
340 FORZ=1TO52
350 X=INT(103*VRND(1)+1)
360 IFX<52THEN600
370 X=X-4*INT(X/4)+1
380 IFT(X)>12THEN370
390 PRINT"NENNE EINE ";S$(X)" KARTE";TAB(25);
```

```

400 INPUTR$
410 FORI=1TO13
420 IFR$=N$(I)THEN480
430 IFR$=M$(I)THEN480
440 NEXTI
450 PRINT"FEHLEINGABE! BITTE WIEDERHOLEN"
460 PRINT
470 GOTO390
480 IFP(X,I)=1THEN550
490 P(X,I)=1
500 K=K+T(X)+1
510 S(I)=S(I)+1
520 T(X)=T(X)+1
530 PRINT"O.K. PUNKTEZAHL:";K
540 GOTO830
550 K=K-(20-T(X))
560 PRINT
570 PRINT"NICHT MEHR FREI! NOCHMAL BITTE"
580 PRINT
590 GOTO390
600 X=X+1-13*(INT(X/13))
610 IFS(X)>3THEN600
620 PRINT"FARBE FUER ";M$(X);TAB(25);
630 INPUTR$
640 FORI=1TO4
650 IFR$=S$(I)THEN710
660 IFR$=T$(I)THEN710
670 NEXTI
680 PRINT"FEHLEINGABE BITTE WIEDERHOLEN"
690 PRINT
700 GOTO620
710 IFP(I,X)=1THEN780
720 P(I,X)=1
730 K=K+S(X)+1
740 S(X)=S(X)+1
750 T(I)=T(I)+1
760 PRINT"O.K. PUNKTEZAHL:";K
770 GOTO830
780 K=K-(10-S(X))
790 PRINT
800 PRINT"NICHT MEHR FREI! NOCHMAL BITTE"
810 PRINT
820 GOTO620
830 NEXTZ
840 PRINT
850 PRINT
860 PRINT"DAS WAR'S!"
870 PRINT
880 PRINT"IHRE PUNKTEZAHL: ";K

```

```
890 PRINT:PRINT:PRINT
900 INPUT"NOCH EIN SPIEL (J/N)":Y$
910 IFY$="J"THEN20
920 END
```

READY.



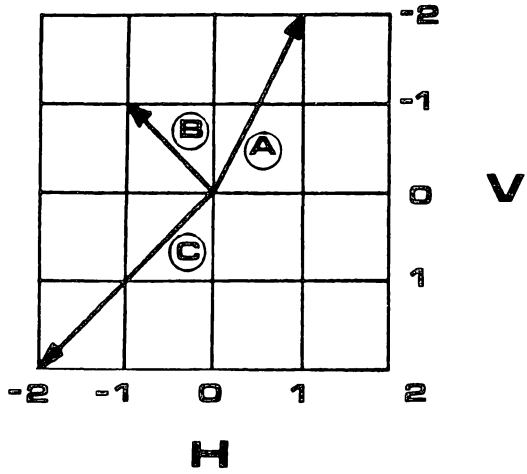
## Ballspiel

Dieses Spiel läßt einige Bälle auf dem Bildschirm herumspringen. Es zeigt einige wesentliche Grundzüge von Spielprogrammen:

1. Die Darstellung von Bewegungsabläufen
2. Die 'Programmierbarkeit' von Vorgängen - Sie können die Anzahl der Ballbewegungen angeben.
3. Mehrdimensionalität - viele Bälle können sich unabhängig voneinander bewegen. Das Programm wendet hierfür eine dreidimensionale Matrix an.

Zunächst werden Sie nach der Zahl der Bälle gefragt. Danach müssen Sie für jeden Ball getrennt die Zahl der Bewegungen innerhalb des gesamten Zyklus und die horizontalen und vertikalen Koordinaten für jede Bewegung eingeben. Anschließend sehen Sie die herumspringenden Bälle. Soll ein Ball sich nicht bewegen, müssen Sie 0,0 als Koordinaten eingeben.

```
10 REM BALLSPIEL
20 PRINT"J"
30 LO=1024:HI=2023:WI=40:D=54272
40 L=HI-LO:TK=81
50 INPUT"WIEWIELE BAELE (MAX.9)":H
60 DIMS(N,10:2),T(N),P(N),C(N)
70 FORI=1TON:P(I)=INT(RND(1)*L)+LO
80 PRINT
90 PRINT"WIEWIELE BEWEGUNGEN FUER BALL "I
100 INPUT(I)
110 FORJ=1TOT(I)
120 PRINT
130 PRINT"H UND V FUER BALL "I;"BEWEGUNG"J
140 INPUTS(I,J,1),S(I,J,2)
150 NEXTJ:NEXTI
160 PRINT"J"
170 FORI=1TON:C(I)=C(I)+1
180 IFC(I)>T(I)THENC(I)=1
```



```

190 NP=P(I)+S(I,C(I),1)+WI*S(I,C(I),2)
200 IFNP>HITHENNP=NP-L
210 IFNP<LOTHENNP=NP+L
220 POKENP,TK:POKEP(I),32:POKENP+D,6+I
230 P(I)=NP
240 NEXTI
250 GOTO170

```



```

310 PRINT"SCHUB NICHT ZULAESSIG"
320 GOTO260
330 PRINT"DU HAST NUR NOCH";E;"KG TREIBSTOFF!!"
340 GOTO260
350 E=E-F
360 V1=V
370 V2=2000*F/(S+E)-10
380 V=V+V2
390 D=(V1+V2/2)*10
400 H=H+D/1000
410 IFK<.01THEN500
420 IFE<0THEN450
430 T=T+10
440 GOTO260
450 PRINT"DU HAST IN";H;"KILOMETERN HOEHE KEINEN TREIBSTOFF MEHR"
460 PRINT"      ES WAR NETT,DICH GEKANNT ZU HABEN"
470 PRINT:PRINT
480 GOTO600
490 F=0
500 K=0
510 K=K+.1
520 V1=1
530 V2=20*F/(S+E)-.1
540 V=V+V2
550 H=H-(V1+V2/2)/1000
560 PRINT"AUFPRALL IN CA.";K;"SEKUNDEN"
570 PRINT"PAUSE":PRINT:PRINT
580 IFV>=-5THEN640
590 IFV>=-15THEN670
600 PRINT"AUFSETZGESCHWINDIGKEIT=";V;"M/SEC"
610 PRINT"DAS SIND";V*3.6;"KM/H--JUNGE,JUNGE!"
620 PRINT"DU SOLLTEST DEINE LIZENZ ZURUECKGEBEN!"
630 END
640 PRINT"SUPERWEICHE LANDUNG V=";V*3.6;"KM/H"
650 PRINT"DU HATTEST NOCH";E;"KG TREIBSTOFF"
660 END
670 PRINT"PLONG-SCHUETTEL-SCHEPPER-KRATZ"
680 PRINT"V=";V*3.6;"KM/H:HART GELANDET, ABER DU HAST UEBERLEBT"
690 END

```

## Das Arbeiten mit binären Zahlen

In den nächsten Kapiteln werden wir mit einigen Registern des Rechners arbeiten. Dazu ist es erforderlich, das binäre und das sedezimale Zahlensystem zu verstehen.

Sie wissen sicher schon, daß Rechner nur binäre Zahlen verarbeiten können. Man versteht darunter Zahlen, die nur mit den Werten Null und Eins aufgebaut sind. Der Rechner stellt intern solche Zahlen in Form von 'Spannung' oder 'keine Spannung' dar. Jede Stelle einer solchen Zahl bezeichnen wir als Bit. Eine Zahl, die aus acht dieser Bits zusammengesetzt ist, wird als Byte bezeichnet. Ein Byte könnte zum Beispiel so aussehen:

1010 1010

Mit einem Byte können 256 verschiedene Zahlen dargestellt werden. Der niedrigstmögliche Wert sieht so aus:

0000 0000

Der größtmögliche so:

1111 1111

Diese Art der Zahlendarstellung nennt man binäre Darstellung. Weil aber das Commodore-Basic nur mit dezimalen Zahlen arbeitet, müssen wir uns überlegen, wie wir binäre Zahlen in dezimale umwandeln können. Dazu ist es notwendig, die dezimale Wertigkeit jeder binären Stelle zu kennen. Für ein Byte sieht die Wertzuordnung so aus:

128 64 32 16 8 4 2 1

Eine Eins in der am weitesten links stehenden Stelle repräsentiert den dezimalen Wert 128. Wenn wir nun den dezimalen Wert einer Binärzahl ermitteln wollen, müssen wir nur den binären Wert jeder Stelle mit seiner dezimalen Wertigkeit multiplizieren.

Hierzu wieder ein Beispiel:

$$\begin{array}{cccccccc}
 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
 x & x & x & x & x & x & x & x \\
 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
 = & = & = & = & = & = & = & = \\
 128 + & 0 + & 32 + & 0 + & 8 + & 0 + & 2 + & 0 = 170
 \end{array}$$

Der Umgang mit binären Zahlen ist meist mühsam, weil man immer sehr viele Stellen bearbeiten muß. Man benützt deshalb oft die sogenannte sedezimale (meist fälschlicherweise als hexadezimal bezeichnete) Zahlendarstellung. Damit lassen sich je vier binäre Stellen zu einer sedezimalen Stelle zusammenfassen. Jede sedezimale Stelle kann zur Darstellung von 16 verschiedenen Zahlen verwendet werden. Die folgende Tabelle zeigt die Zuordnung von binären, dezimalen und sedezimalen Werten:

binär	dezimal	sedezimal
-----	-----	-----
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

Auch für sedezimale Zahlen können wir eine Wertigkeit für jede Stelle angeben. Für vier Stellen sieht sie so aus:

$$4096 \quad 256 \quad 16 \quad 1$$

Sedezimale Zahlen werden oft mit einem voran gestellten '\$'-Zeichen gekennzeichnet:

\$D400

Soll diese Zahl in eine Dezimalzahl umgewandelt werden, wenden wir das gleiche Verfahren wie bei den Binärzahlen an.

\$	D	4	0	0	
	x	x	x	x	
	4096	256	16	1	
	=	=	=	=	
	13*4096	+4*256	+0	+0	= 54272

Mit dem nächsten Programm können Sie binäre Zahlen in Dezimalzahlen umrechnen:

```

100 REM ZAHLENUMRECHNUNG
110 PRINT"3"
120 PRINT"IHRE BINAERZAHL      : ";
130 FORX=1TOS
140 GETA$(X): IFA$(X)=" "THEN140
150 IFA$(X)="1"ORA$(X)="0"THEN170
160 GOTO140
170 PRINTA$(X);
180 NEXTX
190 PRINT:PRINT
200 T=0
210 FORX=0T07
220 IFVAL(A$(X+1))=1THENT=T+2^(7-X)
230 NEXTX
240 PRINT"DER DEZIMALE WERT  :";T
250 PRINT:PRINT
260 GOTO120

```

Oft stehen wir vor der Aufgabe, ein Bit eines Registers zu verändern, ohne die anderen Bits zu beeinflussen. Wir bedienen uns zu diesem Zweck der Anweisungen AND und OR. Soll ein Bit auf Null

gesetzt werden, nehmen wir den aktuellen Registerwert und verknüpfen ihn mittels AND mit einer Zahl, die aus sieben Bits mit dem Wert Eins und einem Bit mit dem Wert Null besteht. Die Null muß an der Stelle stehen, an der wir später im Register den Wert Null haben wollen. Ähnlich verfahren wir, wenn ein Bit zu Eins gemacht werden soll. Wir verknüpfen den Registerwert mit OR mit einer Zahl die aus sieben Nullen und einer Eins an der gewünschten Position besteht. Hierzu wieder ein Beispiel:

```
POKE 53272,PEEK(53272) AND (255-128)
```

Diese Anweisung setzt das höchstwertige Bit des Bytes an der Adresse 53272 auf Null. Alle anderen Bits bleiben unverändert. Binär sieht 255 so aus:

```
1111 1111
```

Ziehen wir davon 128 ab, erhalten wir

```
0111 1111
```

weil der Wert des höchstwertigen Bits 128 beträgt. Mit

```
POKE 53272,PEEK(53272) OR 1
```

wird das niederwertigste Bit an der Adresse 53272 auf Eins gesetzt. Die anderen Bits bleiben wieder unverändert.

## Der freie Fall und der Wurf nach oben

Viele Spiele arbeiten mit Wurf- oder Schießbewegungen. Die Bahnkurven solcher Bewegungen lassen sich mathematisch formulieren und darstellen. Sie können auf die Grundformeln für den freien Fall und für den Wurf nach oben zurückgeführt werden.

In diesem Kapitel wird gezeigt, wie solche Bahnkurven dargestellt werden können und wie man mit POKE-Befehlen Grafikzeichen bewegen kann. Dazu können wir uns den Bildschirm durch Koordinaten unterteilt vorstellen. Der einzige Unterschied zu normalen Koordinatensystemen, die ihren Ursprung in der linken unteren Ecke haben, ist der, daß unser Koordinatensystem den Punkt 0,0 in der linken oberen Ecke hat. Dadurch wird es notwendig, die y-Koordinate umzurechnen.

Die Gleichungen für den freien Fall lassen sich wie folgt angeben:

$$V = -G * T + V0$$

und

$$Y = -G / 2 * T * T + V0 * T + Y0$$

Die verwendeten Zeichen bedeuten:

- G = Erdbeschleunigung
- T = Zeit in Sekunden
- V0 = Anfangsgeschwindigkeit zum Zeitpunkt T = 0
- Y = Höhe
- Y0 = Anfangshöhe zum Zeitpunkt T = 0

Das erste Beispielprogramm zeigt einen freien Fall aus 2000 Meter Höhe. Die Anfangsgeschwindigkeit V0 ist gleich Null.

```

100 REM WURF NACH OBEN
110 PRINT"□"
120 INPUT"GIB V0 EIN(100-200)";V0
130 G=-9.81:T=0
140 Y0=0:H=Y0
150 S=175/2000
160 Y=G/2*T*T+V0*T+Y0
170 COL=55312+40*INT((24-(H*S/8)))
180 POKE COL,6
190 LOC=1040+40*INT((24-(Y*S/8))):COL=55312+40*INT((24-(Y*S/8)))
200 POKE LOC,81:POKE COL,4
210 H=Y
220 IF Y<0 THEN STOP
230 T=T+1
240 GOTO 160

```

Als Anfangswerte sind  $V_0=0$  und  $Y_0=2000$  eingesetzt worden. Die Zeit wird nach jedem Berechnungsdurchgang um eine Sekunde erhöht. Die Bewegungsgeschwindigkeit können Sie durch das Verändern des Inkrements variieren.

Jetzt sehen wir uns den Vorgang einmal als Wurf nach oben an. Die Anfangsgeschwindigkeit kann Werte zwischen 100 und 200 annehmen, ohne daß das sich bewegende Objekt den Bildschirmbereich verläßt.

```

100 REM FREIER FALL
110 Y0=2000:V0=0
120 G=-9.81:T=0
130 H=Y0
140 S=175/Y0
150 PRINT"□"
160 Y=G/2*T*T+V0*T+Y0
170 LOC=1040+40*INT((24-(Y*S/8))):COL=55312+40*INT((24-(Y*S/8)))
180 POKE LOC,81:POKE COL,4
190 IF Y<0 THEN STOP
200 T=T+1
210 GOTO 160

```

## Der schräge Wurf

Das nächste Beispiel zeigt die Überlagerung einer horizontalen und einer vertikalen Bewegung. Die vertikale Bewegung wird mit den vorher beschriebenen Formeln dargestellt. Die Bewegung längs der x-Achse soll mit konstanter Geschwindigkeit vor sich gehen ( $VX=C$ ). Das läßt sich mit folgender Formel darstellen:

$$X = C * T + V0$$

Damit erhalten wir diese vier Gleichungen:

$$VY = -G * T + VY0$$

$$VX = VX0$$

$$Y = -G / 2 * T * T + VY0 * T + Y0$$

$$X = VX0 * T + X0$$

Die mit diesen Formeln entstehenden, liegenden Parabeln haben alle in der linken unteren Ecke ihren Ursprung. Die Anfangswerte  $Y0$  und  $X0$  sind gleich Null. Mit diesen Gleichungen weisen wir die Startwerte zu:

$$VX0 = V0 * \cos (W)$$

$$VY0 = V0 * \sin (W)$$

Das nächste Programm zeichnet eine Wurfparabel auf den Schirm. Mit unterschiedlichen Anfangswerten (100 - 200) können Sie sehen, wie sich die Bahnkurve ändert.

```

100 REM PARABELN
110 PRINT"J"
120 X0=0:Y0=0
130 PRINT"J":INPUT"WINKEL";W
140 INPUT"KRAFT";V0
150 W=W*2*3.1416/360
160 VX=V0*COS(W)
170 VY=V0*SIN(W)
180 G=9.81
190 T=0:H=0:L=0
200 Y=-G/2*T*T+VY*T
210 IF Y<0 THEN FOR D=1 TO 500:NEXT D:GOTO 130
220 X=VX*T
230 LOC=1024+40*INT(24-Y/50)+INT(X/50)
240 COL=55296+40*INT(24-Y/50)+INT(X/50)
250 IF LOC<1024 THEN T=T+1:GOTO 200
260 POKE LOC,81:POKE COL,4
270 L=X:H=Y
280 T=T+1
290 GOTO 200

```

## Grafik mit dem Commodore 64

Der Bildwiederholungspeicher des Commodore 64 beginnt an der Adresse 1024 (\$0400) und erstreckt sich bis zur Adresse 2023 (\$07E7). Vierzig Zeichen pro Zeile bei fünfundzwanzig Zeilen ergeben ein-tausend verschiedene Positionen auf dem Bildschirm. In jeder Position kann ein Wert zwischen Null und 255 stehen. Jeder dieser Werte repräsentiert ein anderes Zeichen aus dem Zeichensatz.

Zu den 1000 Positionen des eigentlichen Bildwiederholungspeichers gehören weitere 1000 Speicherplätze zwischen den Adressen 55296 und 56295, die die Farbinformation für jede Position des Bildes enthalten.

Es kann der eingebaute oder ein von Ihnen definierter Zeichensatz verwendet werden. Diese Definition wird mit Hilfe einiger Bits in einem Kontrollregister an der Adresse 53272 vorgenommen. Maßgebend dafür sind die Bits 1, 2 und 3. Das Bit 0 kann nicht gesetzt werden. Wenn die unteren vier Bits den Wert Null ergeben, werden die untersten zwei KByte des Speichers (Adressen 0 bis \$07FF) als Zeichengenerator verwendet. Jedes Zeichen wird mit je acht Bytes dargestellt. Je nach dem Wert der untersten vier Bits wird eine andere 2KByte-Speicherseite als Zeichengenerator herangezogen. Die folgende Tabelle zeigt die Zuordnung nach Adreßbereich und Zahl:

0 oder 1	\$0000 - \$07FF	
2 oder 3	\$0800 - \$0FFF	
4 oder 5	\$1000 - \$17FF	Großbuchstaben (ROM)
6 oder 7	\$1800 - \$1FFF	Kleinbuchstaben (ROM)
8 oder 9	\$2000 - \$27FF	
10 oder 11	\$2800 - \$2FFF	
12 oder 13	\$3000 - \$37FF	
14 oder 15	\$3800 - \$3FFF	

Das nächste Programm soll die Anwendungweise verschiedener Zeichensätze verdeutlichen. Dazu wollen wir annehmen, daß wir einen Zeichensatz zu erzeugen hätten, der nur in wenigen Details vom ursprünglichen abweicht. Für unser Beispiel wollen wir alle Buchstaben unterstreichen.

Dazu speichern wir den Originalzeichensatz aus dem ROM in das RAM

um. Anschließend nehmen wir im RAM die Änderungen vor und 'schalten' schließlich auf den neuen Zeichensatz um. Damit das Programm laufen kann, müssen wir zunächst einen Interrupt abschalten. Dies geschieht in Zeile 110. Nachdem auf den neuen Zeichensatz umgeschaltet wurde, geben wir diesen Interrupt wieder frei (Zeile 170). In der Zeile 180 wird vom ROM-Zeichensatz auf den RAM-Zeichensatz umgeschaltet. Sie werden feststellen, daß alle Buchstaben normal dargestellt werden, während anstelle der Zahlen nur Unsinn erscheint. Der Grund für dieses Verhalten liegt darin, daß wir nur die ersten 32 Zeichen des Zeichensatzes ins RAM kopiert haben. Der restliche Inhalt des Speicherplatzes ist mit 'Zufalls-werten' belegt. Wollen Sie mehr Zeichen ins RAM übertragen, müssen Sie nur den Wert 255 in Zeile 130 erhöhen (z.B. auf 511).

```
100 REM ZEICHENSATUMSCHALTUNG
110 POKE56334,PEEK(56334)AND254
120 POKE1,PEEK(1)AND251
130 FORI=0TO255
140 POKE12288+I,PEEK(53248+I)
150 NEXTI
160 POKE1,PEEK(1)OR4
170 POKE56334,PEEK(56334)OR1
180 POKE53272,(PEEK(53272)AND240)+12
190 END
```

Das folgende Listing zeigt ein Programm, mit dem wir erreichen, daß alle kopierten Zeichen unterstrichen dargestellt werden. Dies geschieht, indem wir einfach in das letzte Byte des jeweiligen Zeichens den Wert 255 schreiben.

```
200 FORX=0TO64
210 POKE12288+7*X*8,255
220 NEXTX
```

Nun untersuchen wir, wie wir mit einem speziellen Zeichensatz den Hintergrund verändern können.

SCREEN-RAM

Adress	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	
1024																															
1064																															
1104																															
1144																															
1184																															
1224																															
1264																															
1304																															
1344																															
1384																															
1424																															
1446																															
1504																															
1544																															
1584																															
1624																															
1664																															
1704																															
1744																															
1784																															
1824																															
1864																															
1904																															
1944																															
1984																															

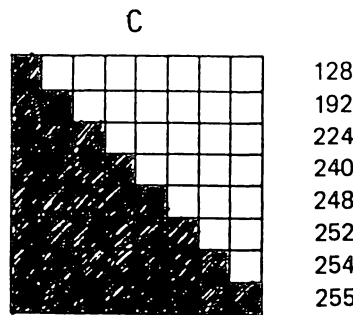
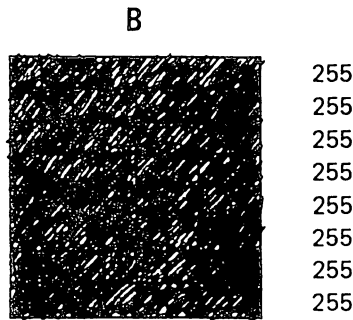
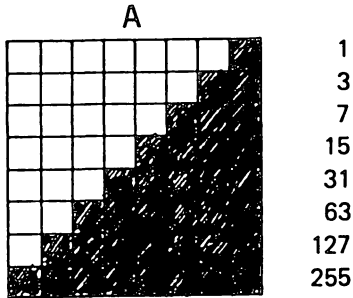
Copyright 1979 by Ing. W. Hofacker GmbH, Tegernseerstr. 18, 8150 Holzkirchen

COLOR-RAM

Address	+ 10										+ 20										+ 30										
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	
55296																															
55336																															
55376																															
55416																															
55456																															
55496																															
55536																															
55576																															
55616																															
55656																															
55696																															
55736																															
55776																															
55816																															
55856																															
55896																															
55936																															
55976																															
56016																															
56056																															
56096																															
56136																															
56176																															
56216																															
56256																															

Copyright 1979 by Ing. W. Hofacker GmbH, Tegernseerstr. 18, 8150 Holzkirchen

Nehmen wir an, wir wollten einen Berg zeichnen. Dazu bräuchten wir drei verschiedene Zeichen: ein Zeichen für die ansteigende Flanke, eines für die abfallende und ein Zeichen um den Inhalt des Bildes aufzufüllen. Das Zeichen für die ansteigende Flanke legen wir auf die Taste A, das Zeichen für die fallende Flanke auf C und das Füllzeichen auf B. Die eigentlichen Zeichen sind sehr leicht zu erzeugen:



Das folgende Programm legt den neuen Zeichensatz beginnend mit der Adresse 12288 im Arbeitsspeicher (RAM) ab. Dieser neue Zeichensatz hat nur für die Tasten A, B und C einen Sinn. Weil die benutzten Buchstaben die zweiten (A), dritten (B) und vierten (C) sind, müssen wir die Zeichen um einen vollständigen Bytesatz für ein Zeichen versetzt im Speicher ablegen. (Das erste Zeichen der Tabelle ist das @ -Zeichen!) Dies erklärt gleichzeitig, warum wir den Wert Acht zur Anfangsadresse addieren müssen.

Nach dem Ausführen der Zeile 150 ist auf den neuen Zeichensatz umgeschaltet worden und Sie sehen möglicherweise einige unsinnige Zeichen auf dem Schirm. Wenn Sie jetzt die Tasten A, B oder C drücken, sollte jedesmal das gewünschte Zeichen erscheinen. Das Drücken der RETURN-Taste führt zu einem Syntax-Fehler und der entsprechenden Fehlermeldung.

Sie können durch gleichzeitiges Betätigen der Tasten RUN/STOP und RESTORE auf den Standardzeichensatz zurückschalten.

```
100 REM SYMBOLE FUER A B C
110 FORX=0TO23
120 READA
130 POKE12288+8*X,A
140 NEXTX
150 POKE53272,(PEEK(53272)AND240)+12
200 DATA1,3,7,15,31,63,127,255
210 DATA255,255,255,255,255,255,255,255
220 DATA128,192,224,240,248,252,254,255
```

Das nächste Programm nimmt Ihnen die Arbeit ab, den Berg von Hand zu zeichnen. Es zeichnet einen Berg in drei verschiedenen Farben. Dazu werden in die korrespondierenden Adressen des Farbspeichers unterschiedliche Zahlen geschrieben (Zeilen 400 - 480).

```

100 REM BERG
110 FORX=0TO23
120 READA
130 POKE12288+8+X,A
140 NEXTX
150 POKE53272,(PEEK(53272)AND240)+12
200 DATA1,3,7,15,31,63,127,255
210 DATA255,255,255,255,255,255,255,255
220 DATA128,192,224,240,248,252,254,255
230 PRINT"J"
300 PRINT"XXXXXXXXXXXX";
310 PRINT"  AC"
320 PRINT"  ABBC"
330 PRINT"  ABBBC"
340 PRINT"  ABBBBBC"
350 PRINT"  ABBBBBBC"
360 PRINT"  ABBBBBBBC"
400 FORX=55776TO55776+39
410 POKEX,1:REM WEISS
420 NEXTX
430 FORX=55816TO55816+79
440 POKEX,12:REM MITTELGRAU
450 NEXTX
460 FORX=55896TO55896+119
470 POKEX,5:REM GRUEN
480 NEXTX
490 GOTO490

```

Der nächste Schritt ist die Darstellung einer kompletten Landschaft. Das folgende Programm soll Ihnen dazu einige Anregungen vermitteln. Lassen Sie Ihrer Fantasie freien Lauf und entwerfen Sie Ihre eigene Landschaft mit einem interessanten Hintergrund.

```

100 REM LANDSCHAFT
110 FORX=0TO47
120 READA
130 POKE12288+8+X,A
140 NEXTX
150 POKE53272,(PEEK(53272)AND240)+12
160 FORX=0TO7
170 POKE12288+32*X,X,0:REM SCHIRM LOESCHEN
180 NEXTX
200 DATA1,3,7,15,31,63,127,255:REM 'A'
210 DATA255,255,255,255,255,255,255,255:REM 'B'
220 DATA128,192,224,240,248,252,254,255:REM 'C'

```

```

230 DATA1,1,15,15,1,1,1,1:REM 'D'
240 DATA128,128,240,240,128,128,128,128:REM 'E'
250 DATA255,255,195,195,195,195,255,255:REM 'F'
290 PRINT"J"
300 PRINT"XXXXXXXXXX";
310 PRINT"
320 PRINT" DE AC
330 PRINT" AC AC ABBBC"
340 PRINT" ABBC ABBC ABBBBBC"
350 PRINT" ABBBC ABBBC ABBBBBC"
360 PRINT" ABBBBBC ABBBBBC ABBBBBC"
370 PRINT" ABBBBBC ABBBBBC ABBBBBC"
380 PRINT"ABBBBBBCABBBBBBCABBBBBBC"
400 PRINT"XXXXXXXXXXXXXXXXXXXX";
410 PRINT" ABBBBBC"
420 PRINT" FFFFFFF ABBBBBC"
430 PRINT" FFFFFFF FFFFF"
440 PRINT" FFFFFFF FFFFF"
450 PRINT" FFFFFFF FFFFF"
460 PRINT" FFFFFFF FFFFF"
470 PRINT" FFFFFFF FFFFF"
480 PRINT:PRINT"BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB"
490 PRINT"BBBBBBBBBBBBBBBBBBBB"
500 FORX=0TO119
510 POKEX+55376,1:REM SCHNEE
520 NEXTX
530 FORX=0TO119
540 POKEX+55496,12:REM FELSEN
550 NEXTX
560 FORX=0TO79
570 POKEX+55616,5:REM GRAS
580 NEXTX
590 FORX=0TO39
600 POKEX+55816,2:REM DAECHE
610 NEXTX
620 FORX=0TO239
630 IFX>20ANDX<40THENPOKE55856+X,2:NEXTX
640 POKEX+55856,7:REM HAEUSER
650 NEXTX
660 FORX=0TO79
670 POKEX+56136,15:REM STRASSE
680 NEXTX
700 POKE 55416+5,9:POKE55416+6,9:REM KREUZ
710 GOTO710

```

## Wie man mit der hochauflösenden Grafik arbeitet

Der Commodore 64 kann Grafiken mit einer Auflösung von 320 mal 200 Bildpunkten erstellen. In diesem Modus können also  $320 * 200 = 64000$  einzelne Bildpunkte individuell angesprochen werden. Auch hier gilt wieder die vorher bereits allgemein beschriebene Einteilung: ein Zeichen belegt acht Bytes, eine Zeile ist vierzig Zeichen lang und auf dem Schirm sind 25 Zeilen darstellbar. Wenn wir uns noch einmal daran erinnern, daß ein Byte aus acht Bits besteht, können wir auch schreiben:

$$8 \text{ Bits} * 8 \text{ Bytes} * 40 \text{ Zeichen} * 25 \text{ Zeilen} = 64000$$

Wir erhalten wieder den Wert 64000, der der Anzahl der Bildpunkte entspricht. Außerdem sehen Sie sofort den Zusammenhang zwischen der Einteilung des Bildwiederholerspeichers und den einzelnen Bildpunkten und ihrer Lage.

Das Byte mit der niedrigsten Adresse enthält die Information für die erste Linie innerhalb des ersten Zeichenplatzes in der linken oberen Ecke des Schirmes. Mit dem folgenden Befehl kann die Basisadresse für hochauflösende Grafikdarstellungen in die Adresse 53272 geschrieben werden:

```
POKE 53272,PEEK(53272) OR 8
```

Damit wird der Adressbereich von 8192 bis 16383 für diese Darstellungsart bereitgestellt. Der Modus kann so angewählt werden:

```
POKE 53265,PEEK(53265) OR 32
```

Damit wird Bit 5 des Bytes an der Adresse 53265 auf Eins gesetzt. Nachdem die beiden Befehle ausgeführt wurden, erscheint ein mehr oder minder zufällig entstandenes Bild auf dem Schirm. Dargestellt werden die Werte, die sich zufällig in den Speicherstellen befinden. Der Bildschirm läßt sich mit dieser Anweisungsfolge löschen:

```
FOR I=8192 TO 16383 : POKE I,0 : NEXT I
```

Die Farbe des Hintergrundes und die Bildpunkte der jeweiligen Darstellung werden in den dafür vorgesehenen Bildwiederholerspei-

chern abgelegt. Die oberen vier Bits eines Bytes definieren die Farbe des Zeichenbereiches, die unteren vier bestimmen seine Hintergrundfarbe. Mit dem nächsten Befehl erhält der gesamte Schirm die gleichen Farben:

```
FOR I=1024 TO 2023 : POKE I,16 * DC + BC : NEXT I
```

Die Variable DC enthält den Kode für die Bildpunktfarbe, die Variable BC den Kode für die Farbe des Hintergrundes. Beide Werte dürfen im Bereich zwischen Null und Fünfzehn liegen.

Ein Bildpunkt wird durch die Angabe seiner x- und y-Koordinaten angewählt. Die beiden Koordinaten müssen dann noch in eine Byteadresse umgerechnet werden. Der in das errechnete Byte geschriebene Wert sagt aus, welche Bildpunkte auf dem Schirm erscheinen sollen und welche nicht. Hierzu die Berechnungsformeln:

```
Zeilenposition = Y-Koordinate/8
Zeichen        = X-Koordinate/8
Zeilennummer   = Y-Koordinate AND 7
Bit            = 7 - (X AND Y)
Byteadresse    = Basisadresse + 320 * Zeilennummer + 8 *
                Zeichen + Zeilennummer
```

Soll der an einer Adresse abgelegte Wert nicht verändert werden, verwenden wir diese Anweisung:

```
POKE Adresse, PEEK(Adresse) OR 2
```

Das unten aufgelistete Programm zeigt alle bisher diskutierten Arbeiten. Sie können Koordinaten eingeben, an deren Position ein Bildpunkt auf dem Schirm dargestellt werden soll. Mit diesem Programm als Unterprogramm können beliebige grafische Darstellungen nur durch Übergabe der Einzelkoordinaten erstellt werden.

```
100 REM HOCHAUFLOESENDE GRAFIK
110 POKE53272,210R8:POKE53265,270R32
120 PRINT"7";BA=8192
130 FORC=BATORA+7999
140 POKEC,0
```

```

5 REM SIMULATION DES DRAW-BEFEHLS
10 INPUT "X1";X1
20 INPUT "Y1";Y1
30 INPUT "X2";X2
40 INPUT "Y2";Y2
42 POKE 53265,59
44 POKE 53272,25
45 FOR V=1024 TO 2023
46 POKE V,16
47 NEXT V
48 FOR T=8192 TO 16383
49 POKE T,0
50 NEXT T
55 DX=X2-X1:DY=Y2-Y1
57 IF ABS(DY)<ABS(DX) THEN 64
58 FOR YL=Y1 TO Y2 STEP SGN(DY)
59 X=DX/DY*YL+X1
60 Y=YL
61 GOSUB 200
62 NEXT YL
63 GOTO 300

```

Das nächste Programm dient zum Zeichnen einer Linie zwischen zwei Punkten. Die Punkte werden durch ihre x- und y-Koordinatenwerte angegeben. Dieses Programm ersetzt gewissermaßen den auf anderen Systemen verfügbaren DRAW-Befehl.

```

150 NEXTC
160 POKE53265,27:POKE53272,21
170 INPUT "X-KOORDINATE(0-319)";X
180 INPUT "Y-KOORDINATE(0-199)";Y
190 POKE53272,210R8
200 POKE53265,270R32
210 R0=INT(Y/8)
220 CH=INT(X/8)
230 LI=YAND7
240 BI=7-(XAND7)
250 BY=BA+R0*320+CH*8+LI
260 POKEBY,PEEK(BY)OR21BI
270 FORD=1TO2000:NEXTD
280 POKE53265,27:POKE53272,21
290 PRINT "X";:GOTO 170

```

```

64 FOR XL=X1 TO X2 STEP SGN(DX)
65 X=XL
66 Y=DY/DX*XL+Y1
67 GOSUB 200
68 NEXT XL
69 GOTO 300
200 YP=INT(Y/8)
210 XP=INT(X/8)
220 A1=(YP*40+XP)*8
230 AY=Y-8*YP+A1
240 AA=3192
250 R=X-8*XP
260 M=2*(7-R)
270 I=PEEK(AY+AA)
280 POKE AY+AA,I OR M
290 RETURN
300 FOR T=1 TO 5000:NEXT T
310 POKE 53265,27:POKE 53272,21
320 END

```

Eine weitere Programmsequenz, die als Unterprogramm geeignet ist, zeigt das folgende Listing. Sie dient dazu, einzelne Bildpunkte durch Angabe der x- und der y-Koordinaten (x-Werte von 0 - 319, y-Werte von 0 - 199) in den Variablen X bzw. Y anzuwählen. Die Sequenz ist als Unterprogramm geschrieben und kann mit GOSUB angesprungen werden.

```

REIHE=INT(Y/8)
CHAR=INT(X/8)
ZEILE=YAND7
BIT=7(XAND7)
BYTE=BASE+REIHE*320+CHAR*8+ZEILE
POKEBYTE,PEEK(BYTE)OR2BIT
RETURN

```

Unser nächstes Programm benutzt die gerade diskutierte Sequenz zur Darstellung einer Wurfparabel im hochauflösenden Grafikmodus.

```

100 REM WURFFPARABEL IN HOCHAUFLOESENDER GRAFIK
110 PRINT"J"
120 INPUT"WINKEL (0-90)";W
130 INPUT"KRAFT (0-80)";V0
140 W=W*2*3.1416/360
145 PRINT:PRINT"LOESCHEN DES GRAFIKTEILS"
150 FORI=0TO7999
160 POKE192+I,0
170 NEXTI
175 PRINT"LOESCHEN DES FARBTEILS"
180 FORI=0TO999
190 POKE1024+I,16
200 NEXTI
210 RA=8192:POKE53272,210R8:POKE53265,270R32
220 VX=V0*COS(W)
230 VY=V0*SIN(W)
240 G=9.81
250 T=0:H=0:L=0
260 Y=-G/2*T*T+VY*T:Y=200-Y
270 X=VX*T
280 IFY>200THEN280
290 IFY<0THEN310
300 GOSUB1000
310 L=X:H=Y
320 T=T+.1
330 GOTO260
1000 REM UNTERPROGRAMM PLOT
1010 RO=INT(Y/8)
1020 CH=INT(X/8)
1030 LI=YAND7
1040 BI=7-(XAND7)
1050 BY=RA+RO*320+CH*8+LI
1060 POKEBY,PEEK(BY)OR21BI
1070 RETURN

```

Jetzt zeichnen wir mit einer geringen Variation des vorherigen Programms den Graphen einer Funktion.

```
100 REM PLOTTEN EINER FUNKTION
110 FORI=0T07999
120 POKE8192+I,0
130 NEXTI
140 FORI=0T0999
150 POKE1024+I,16
160 NEXTI
170 BA=8192:POKE53272,210R8:POKE53265,270R32
180 FORI=0T0360STEP3
190 D=I*π/180
200 Y=30*SIN(D):Y=-Y+100
210 X=I/1.5
220 GOSUB1000
230 NEXTI
240 GOTO240
1000 REM UNTERPROGRAMM PLOT
1010 RO=INT(Y/8)
1020 CH=INT(X/8)
1030 LI=YAND7
1040 BI=7-(XAND7)
1050 BY=BA+RO*320+CH*8+LI
1060 POKEBY,PEEK(BY)OR2+BI
1070 RETURN
```

## Das Arbeiten mit mehreren Farben

Im hochauflösenden Grafikmodus haben Sie Zugriff auf jeden einzelnen Bildpunkt auf dem Schirm. Die Bits des Farbspeichers bestimmen die Farbe, die Bits des WiederholSpeichers bestimmen, welcher Bildpunkt dargestellt wird und welcher nicht. Überall da, wo kein Bildpunkt vorhanden ist, erscheint die angegebene Hintergrundfarbe.

Der Mehrfarbenmodus bringt zusätzliche Freiheitsgrade. Jeder Bildpunkt kann in vier verschiedenen Farben dargestellt werden. Für die Unterscheidung der Farben werden zwei Bits des jeweiligen horizontalen Linienstücks verwendet. Dadurch steht dann nicht mehr die gleiche horizontale Auflösung wie bisher zur Verfügung. Sie wird auf die Hälfte reduziert. Dieser Modus wird durch das Setzen des vierten Bits in Adresse 53270 (\$D016) angewählt. Dazu kann dieser Befehl benutzt werden:

```
POKE 53270,PEEK(53270) OR 16
```

Nachdem Sie den Mehrfarbenmodus angewählt haben, bleibt Ihnen die Wahl, ob Sie an einer Zeichenposition mit der hochauflösenden Grafik oder mit mehreren Farben arbeiten wollen. Der Arbeitsmodus hängt nun nur noch von der Zahl ab, die Sie in den Farbspeicher schreiben. Die Werte Null bis Sieben bedeuten, daß Sie mit hochauflösender Grafik arbeiten wollen. Werte im Bereich von Acht bis Fünfzehn bewirken die Verwendung des Mehrfarbenmodus.

Die Farbe wird dann dem Wert der beiden Farbbits entsprechend auf den Schirm gebracht. Dabei gilt folgende Zuordnung:

Bits	Farbe
00	#0 (entspricht der normalen Hintergrundfarbe)
01	#1
10	#2
11	#3

Die Hintergrundfarben #0 bis #2 sind an folgenden Adressen definiert:

#0	55281 (\$D021)
#1	55282 (\$D022)
#2	55283 (\$D023)

## Das Verschieben des Bildinhalts

Der Commodore 64 erlaubt es Ihnen, den aktuellen Bildinhalt um je 7 Bildschirmpositionen in horizontaler und vertikaler Richtung zu verschieben. Wenn Ihnen das nicht ausreicht, hilft Ihnen die anschließend geschilderte Methode weiter. Nachdem das Bild um sieben Positionen verschoben wurde, rufen Sie ein Maschinenunterprogramm auf, das den Bildschirminhalt zunächst in seine Ausgangsposition zurückbringt und dann den Bildinhalt um eine Zeichenposition oder um eine Zeile verschiebt. Für horizontale Verschiebungen sollten Sie den Bereich auf 38 Spalten begrenzen, für vertikale auf 24 Zeilen. Dies übernimmt folgender Befehl:

```
POKE 53265, PEEK(53265) AND 247      (horizontal)
```

bzw.

```
POKE 53270, PEEK(53270) AND 247      (vertikal)
```

Die unteren drei Bits steuern die Verschiebung in Bildpunktabständen. Die nächsten beiden Befehle erlauben eine Verschiebung in kleineren Abständen:

```
POKE 53270,(PEEK(53270)AND 248)+X    (horizontal)
```

und

```
POKE 53265,(PEEK(53265)AND 248)+X    (vertikal)
```

Für X und Y sind Werte zwischen 0 und 15 zulässig. Untersuchen Sie dieses Programm:

```
10 POKE53270,PEEK(53270)AND247
20 FORX=7TO0STEP-1
30 POKE53270,(PEEK(53270)AND248)+X
40 FORI=1TO100:NEXTI
50 NEXTX
60 GOTO20
```

Der Bildinhalt wird mit diesem Programm um sieben Bildpunktpositionen nach links verschoben. Danach beginnt das Programm von neuem. Das Maschinenunterprogramm, dessen Listing Sie im Anschluß an diesen Absatz finden, bringt den Bildinhalt zur Ausgangsposition zurück und verschiebt danach alle Zeichen um eine ganze Zeichenposition nach links.

Programme dieser Art legt man am besten ab der Adresse 49152 (\$C000) ab, weil dieser Bereich von Basic nicht benutzt wird.

```

$C000      A9 07                LDA #7
$C002      0D 16 D0            ORA $D016
$C005      8D 16 D0            STA $D016
$C008      A2 00                LDX #0
$C00A      BD 01 04            M1 LDA $0401,X      ;1.ZEILE
$C00D      9D 00 04            STA $0400,X
$C010      BD 29 04            LDA $0429,X      ;2.ZEILE
$C013      9D 28 04            STA $0428,X
          .
          .
          .
          BD C1 07            LDA $07C1,X      ;25.ZEILE
          9D C0 07            STA $07C0,X
          E8                    INX
          E0 28                CPX #40
          F0 03                BEQ RET
          4C 0A C0             JMP M1
          60                    RET RTS

```

Wenn Sie mehrere Farben auf dem Bildschirm darstellen, dürfen Sie nicht vergessen, auch den Farbspeicher entsprechend zu verschieben. Für diese Aufgabe eignet sich ebenfalls ein Maschinenprogramm.

```
100 REM SCROLLEN NACH LINKS
110 FORX=0TO168
120 READ Z
130 POKE49152+X,Z
140 NEXTX
150 POKE53270,(PEEK(53270)AND247
160 POKE53270,(PEEK(53270)AND248)+7
170 FORP=6TO0STEP-1
180 POKE53270,(PEEK(53270)AND248)+P
200 NEXTP
210 SYS(49152)
220 GOTO170
240 DATA169,7,13,22,208,141,22,208
250 DATA162,0,189,1,4,157,0,4
260 DATA189,41,4,157,40,4,189,81,4,157,80,4
270 DATA189,121,4,157,120,4,189,161,4,157,160,4
280 DATA189,201,4,157,200,4,189,241,4,157,240,4
290 DATA189,25,5,157,24,5,189,65,5,157,64,5
300 DATA189,105,5,157,104,5,189,145,5,157,144,5
310 DATA189,185,5,157,184,5,189,225,5,157,224,5
320 DATA189,9,6,157,8,6,189,49,6,157,48,6
330 DATA189,89,6,157,88,6,189,129,6,157,128,6
340 DATA189,169,6,157,168,6,189,209,6,157,208,6
350 DATA189,249,6,157,248,6,189,33,7,157,32,7
360 DATA189,73,7,157,72,7,189,113,7,157,112,7
370 DATA189,153,7,157,152,7,189,193,7,157,192,7
380 DATA232,224,40,240,3,76,10,192,96
```



## Sprites

Dieses Kapitel beschäftigt sich mit einer Commodore-64-spezifischen Thematik, den sogenannten Sprites.

Unter einem Sprite versteht man ein grafisches Objekt, dessen Farbe und dessen Form vom Programmierer definiert werden können. Sprites können sehr leicht bewegt werden. Der Anwender muß nur Koordinaten angeben und schon werden Sprites dort hingeschrieben. Das ist ein wesentlicher Vorteil gegenüber der hochauflösenden Grafik, bei der Sie selbst ein Objekt durch den Bildwiederhol-speicher bewegen müssen. Sprites eignen sich deshalb hervorragend für Spiele mit schneller Grafik.

Der Commodore 64 kann gleichzeitig mit bis zu acht verschiedenen Sprites arbeiten. Die Sprites werden über Register angesteuert, die ab der Speicheradresse 53248 (\$D000) liegen. Diese Adresse weisen wir von jetzt an der Variablen V zu. Die verschiedenen Register besitzen dann die Adressen V+6 oder V+23 und so weiter.

Bevor Sprites auf dem Bildschirm erscheinen, ist einiges zu tun. Die darzustellenden Sprites müssen 'eingeschaltet' werden. Das geschieht an der Adresse V+21. Soll das Sprite mit der Nummer 0 eingeschaltet werden, müssen wir eine Eins in diese Adresse schreiben. Sprite Sieben wird mit 128 aktiviert. Sollen alle Sprites gleichzeitig verwendet werden, schreibt man 255 in die Adresse V+21 (Beispiel: POKE V+21,6 schaltet die Sprites 1 und 2 ein).

Als nächstes muß der Speicherbereich definiert werden, der die Forminformation für das Sprite aufnehmen soll. Jedes Sprite belegt 63 Bytes Speicherplatz. In die Speicherstellen mit den Adressen 2040 bis 2047 werden die Blocknummern des jeweiligen Speicherbereichs eingetragen. Als Blocknummern kommen Zahlen zwischen 192 und 199 in Betracht. Mit POKE 2040,192 wird der 64-Byte Block mit der Adresse  $192 * 64 = 12288$  für Sprite 0 reserviert. Damit können wir die Forminformation für Sprite 0 im Speicherbereich zwischen den Adressen 12288 und 12350 ablegen. Jedes Sprite besteht aus 24 mal 21 Bildpunkten. Es ist auch möglich, einen Speicherblock mehr als nur einem Sprite zuzuweisen. Dadurch haben dann mehrere Sprites dieselbe Form.

Die Definition der Farbe des Sprites ist der nächste Schritt. Die Farbregister liegen an den Adressen V+39 bis V+46. Mit POKE können wir Farbwerte zwischen Null und Fünfzehn dort ablegen.

Jetzt müssen wir die horizontale Koordinate angeben. Sie wird an der Adresse V+0,....,V+7 für die Sprites 0 bis 7 abgespeichert. Hier dürfen Werte zwischen 0 und 255 stehen. Im Register V+16 stehen noch höherwertige Bits zur Verfügung, mit denen der volle Darstellungsbereich des Bildschirms ausgenutzt werden kann. Soll zum Beispiel das Sprite Nummer 2 an der horizontalen Position 265 stehen, müssen Sie eine Vier an V+16 und eine Zehn an V+4 ablegen.

Als letztes geben wir die vertikale Position an. Dies ist etwas einfacher als bei der horizontalen Position, weil hier nur Werte zwischen Null und 255 in Betracht kommen. Der Wert für Sprite Null wird an die Adresse V+1 geschrieben, der Wert für Sprite 1 an die Adresse V+3, der Wert für Sprite 3 an die Adresse V+5 und so weiter.

Die vertikale und horizontale Positionsangabe orientiert sich an der linken oberen Ecke des Sprites.

Nachdem Sie die fünf erforderlichen Parameter definiert haben, sollten Sie Ihr Sprite auf dem Schirm sehen. Es gibt aber noch mehr Register, die für Sprites wichtig sind.

Die Größe eines Sprites in x-Richtung kann durch das Setzen eines Bits an der Adresse V+29 verdoppelt werden. Mit POKE V+28,8 wird die Größe von Sprite 3 in x-Richtung verdoppelt. Ähnliches gilt für die Ausdehnung in y-Richtung. Als Adresse kommt hier V+23 in Frage. Setzt man die korrespondierenden Bits wieder auf Null, erhält das Sprite wieder seine ursprüngliche Größe.

Die Priorität der Darstellung eines Sprites gegenüber dem Hintergrund kann gewählt werden. Normalerweise liegt ein sich bewegendes Sprite immer über dem im Hintergrund befindlichen Bildinhalt. Durch das Setzen des korrespondierenden Bits an der Adresse V+27 läßt sich das ändern. Liegen zwei Sprites übereinander, so ist das mit der niedrigeren Nummer vollständig sichtbar. Es hat die höhere Priorität. Das Sprite mit der Nummer Null hat die höchstmögliche Priorität.

Kollisionen zwischen Sprites, beziehungsweise zwischen Sprites

und dem Hintergrund sind einfach feststellbar. Das korrespondierende Bit eines Sprites wird zu Eins (Adresse V+31), wenn es mit dem Hintergrund kollidiert. Berühren oder überlappen sich zwei Sprites, werden ihre korrespondierenden Bits an der Adresse V+30 zu Eins. In beiden Fällen ist dies durch wenige Operationen einfach feststellbar. Die Register an den Adressen V+30 und V+31 werden automatisch beim Auslesen gelöscht. Lesen Sie nach einer Kollision und anschließender Trennung der Sprites 0 und 1 Register V+30 aus, erhalten Sie den Wert Drei. Erneutes Lesen von V+30 ergibt den Wert Null.

Sprites können eine einheitliche oder drei verschiedene Farben haben. Letzteres wird durch das Anwählen des Mehrfarbenmodus erreicht (Register V+28). Die beiden zusätzlichen Farben werden in den Registern V+37 und V+38 abgelegt. Sie gelten für alle Sprites. Die endgültige Anwahl geschieht wieder mit zwei Bits im Formregister der Sprites. Sinngemäß gilt hier das Gleiche wie bei der normalen Mehrfarbenbetriebsart.

Das nächste Programm nutzt einige der gerade beschriebenen Eigenschaften. Es läßt acht Sprites über den Bildschirm laufen, deren Bewegung, Farbe und Form durch Zufallszahlen bestimmt werden. Zusätzlich wird laufend die Hintergrundfarbe gewechselt.

```
100 REM SPRITEBEISPIEL
110 PRINT"J":V=53248
120 POKEV+21,255
130 FOR A=0 TO 7
140 POKE 2040+A,192+A
150 NEXT A
160 FOR B=12288 TO 12798
170 POKE B,255*VRND(0)
180 NEXT B
190 FOR C=0 TO 7
200 POKE V+39+C,255*VRND(0)
210 NEXT C
220 POKE V+29,255
230 POKE V+23,255
240 POKE V+28,255
250 POKE V+37,255*VRND(0)
260 POKE V+38,255*VRND(0)
270 FOR D=0 TO 7
280 POKE V+2*D,255*VRND(0)
290 POKE V+2*D+1,255*VRND(0)
300 NEXT D
310 POKE V+37,255*VRND(0)
320 POKE V+38,255*VRND(0)
330 POKEV+33,255*VRND(0)
340 GOTO 270
```

## Ein Editor für Sprites

In diesem Kapitel beschäftigen wir uns mit einem Hilfsprogramm, das Ihnen helfen soll, Ihre Sprites leichter zu erstellen. Auf dem Bildschirm erscheint ein Rahmen, der Ihr Sprite vergrößert darstellt. Sie können Bildpunkt für Bildpunkt die Form mit beliebigen Zeichen (einzige Ausnahme: Leerräume) definieren. Sie erreichen die einzelnen Bildpunkte mit den Cursortasten. Mit der Leertaste können Sie Bildpunkte wieder löschen, die Sie doch nicht setzen wollten. Achten Sie darauf, daß Sie nicht außerhalb des Rahmens arbeiten.

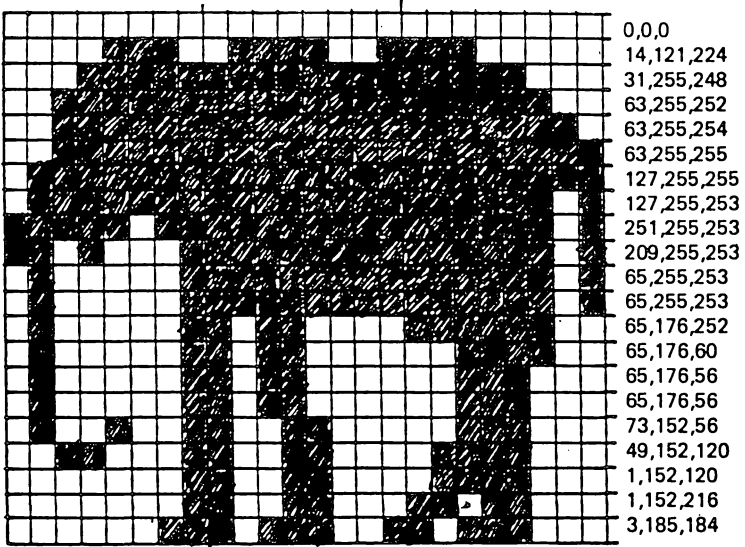
Wenn Sie die Spriteform fertig definiert haben, drücken Sie RETURN. Danach erscheint das soeben erstellte Sprite in der rechten unteren Ecke. Mit H und V können Sie das Sprite horizontal bzw. vertikal vergrößern. Die O-Taste bringt Ihr Sprite wieder in die Ausgangsgröße zurück. Mit der M-Taste können Sie den Mehrfarbenmodus anwählen. Dies ist aber nur sinnvoll, wenn Ihr Sprite als mehrfarbiges Sprite entworfen wurde. Sind Sie mit dem Erscheinungsbild des Sprites noch nicht zufrieden, bringt Sie das Drücken der C-Taste in den Editor zurück, wo Sie erneut Bildpunkte setzen oder löschen können. Das Drücken der Q-Taste liefert Ihnen die Daten Ihres Sprites. Sie können Sie später in DATA-Anweisungen Ihres Programmes einbringen.

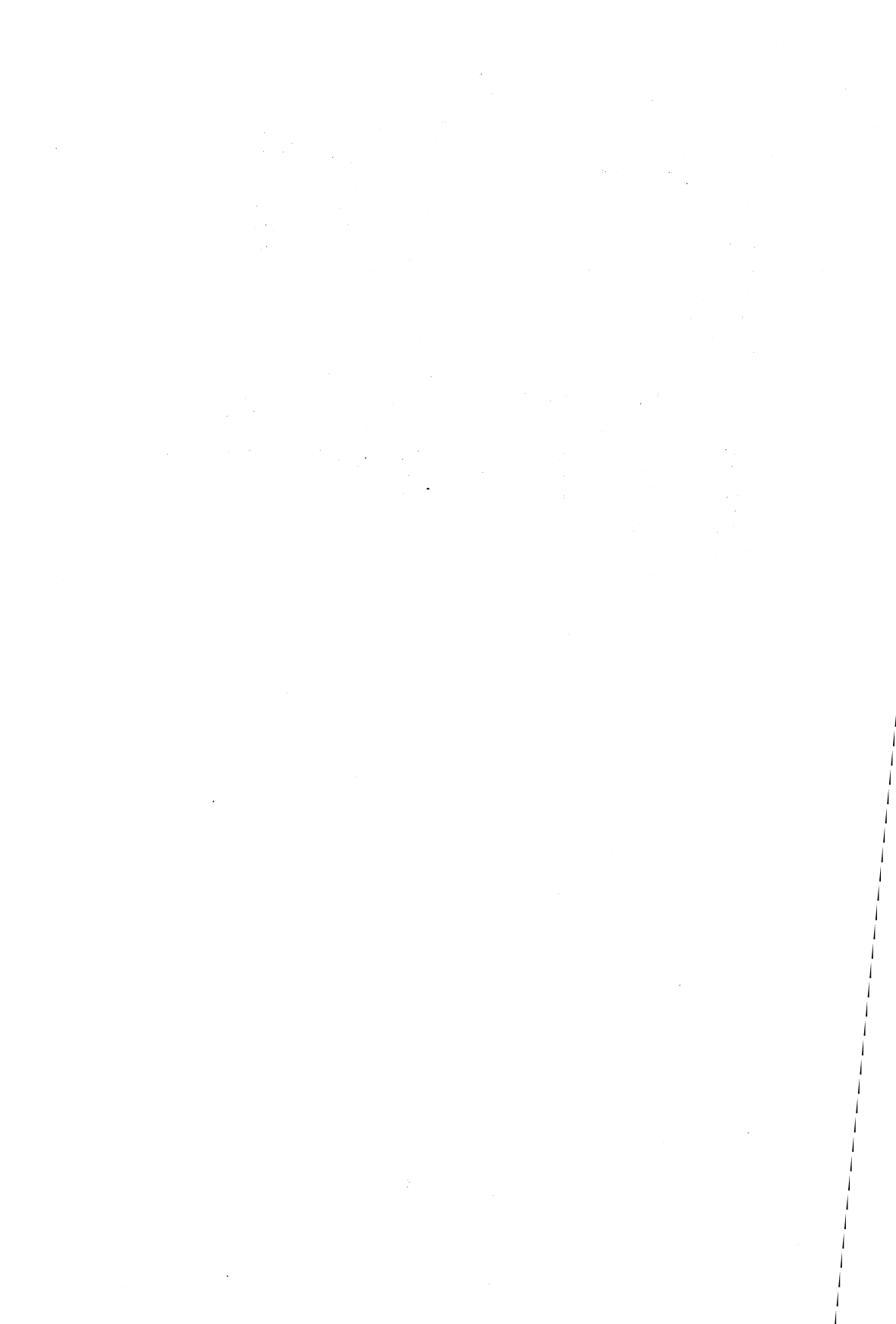
```
100 REM SPRITE EDITOR
110 PRINT"J"
120 FORQ=1026T01049:POKEQ,100:NEXTQ
130 FORQ=55298T055321:POKEQ,13:NEXTQ
140 FORQ=1065T01865STEP40:POKEQ,103:NEXTQ
150 FORQ=55337T056137STEP40:POKEQ,13:NEXTQ
160 FORQ=1090T01890STEP40:POKEQ,101:NEXTQ
170 FORQ=55362T056162STEP40:POKEQ,13:NEXTQ
180 FORQ=1906T01929:POKEQ,99:NEXTQ
190 FORQ=56178T056211:POKEQ,13:NEXTQ
195 INPUTA$
200 I=0:Y=0
210 FORX=0T020
220 FORQ=0T07
230 IFPEEK(1066+X*40+Q)=32THENA=0:GOTO250
240 A=1
```

```

250 I=I+A*(2↑(7-Q))
260 NEXTQ
270 POKE960+Y, I
280 I=0:Y=Y+1
290 FORQ=0TO7
300 IFPEEK(1074+X*40+Q)=32THENA=0:GOTO320
310 A=1
320 I=I+A*(2↑(7-Q))
330 NEXTQ
340 POKE960+Y, I
350 I=0:Y=Y+1
360 FORQ=0TO7
370 IFPEEK(1082+X*40+Q)=32THENA=0:GOTO390
380 A=1
390 I=I+A*(2↑(7-Q))
400 NEXTQ
410 POKE960+Y, I
420 I=0:Y=Y+1
430 NEXTX
440 V=53248
450 POKEV+21,4:POKE2042,15
460 POKEV+23,0:POKEV+29,0
470 POKEV+4,250:POKEV+5,200:REM POSITION
480 POKEV+37,10:POKEV+38,7:POKEV+41,13
490 GETA$:IFA$=""THEN490
500 IFA$="O"THENPOKEV+21,4:POKEV+23,0:POKEV+28,0:POKEV+29,0:POKE2042,15:GOTO490
510 IFA$="M"THENPOKEV+28,4:GOTO490
520 IFA$="H"THENPOKEV+29,4:GOTO490
530 IFA$="V"THENPOKEV+23,4:GOTO490
540 IFA$="C"THENPOKEV+21,0:GOTO180
550 IFA$="Q"THEN570
560 GOTO490
570 PRINT"J"
580 PRINT"ZAHLEN FUER DIE DATA-ANWEISUNGEN :":PRINT
590 FORQ=960TO1022
600 PRINTPEEK(Q);
610 NEXTQ
620 END

```





## Die Priorität von Sprites

Wie Sie schon wissen, hängt die Priorität eines Sprites von seiner Ordnungsnummer ab. Sprite 0 hat die höchste, Sprite 7 die niedrigste Priorität. Das nächste Programm veranschaulicht dies. Es zeigt acht Sprites mit der gleichen Form, aber unterschiedlicher Farbe, die über den Bildschirm bewegt werden. Wo die Sprites sich überlagern, wird die jeweilige Prioritätsrangordnung sichtbar. Das Sprite mit der höchsten Priorität überlagert alle anderen.

```
100 REM SPRITEPRIORITÄT
110 V=53248
120 POKEV+21,255
130 POKEV+23,255:POKEV+29,255
140 FORX=0TO7
150 POKEV+39+X,X+7
160 NEXTX
170 FORX=0TO62
180 READA
190 POKE12288+X,A
200 NEXTX
300 DATA 0,24,0,0,60,0,0,126,0,0,255,0
310 DATA 1,255,128,3,255,192,7,255,224
320 DATA 15,255,240,31,255,248,63,255,252
330 DATA 127,255,254,63,255,252,31,255
340 DATA 248,15,255,240,7,255,224,3,255
350 DATA 192,1,255,128,0,255,0,0,126,0
360 DATA 0,60,0,0,24,0
500 POKE2040,192:POKE2041,192:POKE2042,192
510 POKE2043,192:POKE2044,192:POKE2045,192
520 POKE2046,192:POKE2047,192
530 FORX=0TO14STEP2
540 POKEV+X,100
550 NEXTX
560 FORY=1TO15STEP2
570 POKEV+Y,80
580 NEXTY
590 FORX=0TO14STEP2
600 FORZ=0TO100
```

```
610 POKEV+X,100+Z
620 NEXTZ
630 NEXTX
640 FORY=1TO15STEP2
650 FORZ=0TO100
660 POKEV+Y,80+Z
670 NEXTZ
680 NEXTY
690 FORX=0TO14STEP2
700 FORZ=1TO10*X
710 POKEV+X,200-Z
720 NEXTZ
730 NEXTX
740 FORY=1TO15STEP2
750 FORZ=1TO5*Y
760 POKEV+Y,180-Z
770 NEXTZ
780 NEXTY
790 Z=8
800 POKEV+39+Z,15*RND(0)
810 Z=Z-1:IFZ=0THENZ=8
820 FORD=1TO50:NEXTD
830 GOTO000
```

## Wie man Spritekollisionen feststellt

Der Videochip des Commodore 64 besitzt Register, die es auf einfache Art ermöglichen, festzustellen, ob sich Sprites berühren oder ob sie in Kontakt mit dem Hintergrund sind. Dieses Ausstattungsmerkmal ist beim Erstellen von Spielprogrammen sehr wertvoll. Soll zum Beispiel ein Sprite durch einen Tunnel (Hintergrund) geführt werden, ohne anzustoßen, dann ist mit einer periodischen Abfrage des entsprechenden Registers sehr leicht festzustellen, ob dies dem Spieler gelungen ist. Falls eine Kollision stattfindet, können wir eine Art von Explosion oder etwas ähnliches programmieren, was dem Spieler auf humorvolle Weise seinen Fehler zeigt. Das Bit des Registers, das die Kollision anzeigt, bleibt solange gesetzt, bis wir das Register gelesen haben. Erst der Lesevorgang löscht es wieder.

Das nachfolgende Programm demonstriert die Arbeitsweise des 'Kollisionsregisters', indem es seinen Inhalt auf dem Schirm sichtbar macht. Sie sehen eine Zahl, die zwischen Null (keine Kollision) und Drei (Kollision) wechselt. Der Wert Drei entsteht, weil die Sprites 0 und 1 einander überlappen. Das Register enthält binär 0000 0011.

```
100 REM KOLLISION
110 PRINT"J":V=53248
120 POKEV+21,3
130 POKEV+23,3:POKEV+29,3
140 POKEV+39,7:POKEV+40,8
150 POKE2040,192:POKE2041,192
160 POKEV,120:POKEV+2,200
170 POKEV+1,100:POKEV+3,100
180 FORX=0TO62
190 READA:POKE12288+X,A
200 NEXTX
210 FORX=0TO150
220 POKEV+2,200-X
230 PRINT"KOLLISIONSREGISTER :";PEEK(V+30)
240 NEXTX
250 GOTO210
300 DATA 0,24,0,0,60,0,0,126,0,0,255,0
```

```

310 DATA 1,255,128,3,255,192,7,255,224
320 DATA 15,255,240,31,255,248,63,255,252
330 DATA 127,255,254,63,255,252,31,255
340 DATA 248,15,255,240,7,255,224,3,255
350 DATA 192,1,255,128,0,255,0,0,126,0
360 DATA 0,60,0,0,24,0

```

Nachdem wir eine Berührung festgestellt haben, programmieren wir eine Explosion eines der Sprites. Wie man das macht, zeigt das nächste Programm. Es benutzt einen eigenen Speicherbereich, um die Form des zweiten Sprites zu definieren. In der Zeile 205 wird der Inhalt des Kollisionsregisters gelesen. Dadurch wird ein möglicherweise durch eine vorangegangene Kollision gesetztes und bisher nicht zurückgesetztes Bit gelöscht. In Zeile 230 wird dann die eigentliche Überwachung des Zusammenstoßes vorgenommen. In den Zeilen 400 bis 420 wird die Farbe des Sprites verändert. In den Zeilen 440 bis 500 wird die Form ebenfalls verändert.

```

100 REM EXPLOSION
110 PRINT"Q":V=53248
120 POKEV+21,3
130 POKEV+23,3:POKEV+29,3
140 POKEV+39,7:POKEV+40,8
150 POKE2040,192:POKE2041,193
160 POKEV,120:POKEV+2,200
170 POKEV+1,100:POKEV+3,100
180 FORX=0TO62
190 READA:POKE12288+X,A:POKE12288+64+X,A
200 NEXTX
205 Q=PEEK(V+30)
210 X=200
220 POKEV+2,X
230 IFPEEK(V+30)=3THEN400
240 X=X-1
250 GOTO220
300 DATA 0,24,0,0,60,0,0,126,0,0,255,0
310 DATA 1,255,128,3,255,192,7,255,224
320 DATA 15,255,240,31,255,248,63,255,252
330 DATA 127,255,254,63,255,252,31,255
340 DATA 248,15,255,240,7,255,224,3,255
350 DATA 192,1,255,128,0,255,0,0,126,0
360 DATA 0,60,0,0,24,0

```

```

400 FORX=1T030
410 POKEV+40,15*RND(0)
420 NEXTX
430 POKEV+40,8
440 FORX=0T054STEP2
450 POKE12288+64+X,PEEK(12288+64+X)AND170
460 NEXTX
470 FORX=0T054
480 POKE12288+64+X,0
490 FORD=1T020:NEXTD
500 NEXTX

```

Wir haben festgestellt, daß wir einem Sprite einen Speicherbereich zuweisen können, der die Forminformation aufnimmt. Dies ist für bis zu acht verschiedene Sprites möglich. Wir sind damit in der Lage, acht unterschiedlich geformte Sprites auf dem Schirm darzustellen.

Eine weitere Möglichkeit besteht in der Verwendung zweier verschiedener Formen. So können wir zum Beispiel den ersten vier Sprites die erste und den zweiten vier Sprites die zweite Form zuordnen. Damit haben wir zwar wieder acht Sprites definiert, tatsächlich arbeiten wir aber nur noch mit zwei Formen. Was bringt uns das?

Nehmen wir an, wir wollen ein sich schnell bewegendes grafisches Objekt - als Beispiel wollen wir einen kleinen Mann betrachten - auf den Bildschirm bringen. In diesem Fall brauchen wir mehr als eine Form für die Darstellung der Bewegung. Jede Phase des Bewegungsablaufes wird sozusagen als fotografische Momentaufnahme in eine Sprite-Formdefinition eingebracht. Bringt man später die einzelnen Formen in einer bestimmten Reihenfolge und Geschwindigkeit auf den Bildschirm, ergibt sich der Eindruck einer kontinuierlichen Bewegung. Der Vorgang gleicht in gewissem Maß der Herstellung eines Zeichentrickfilms. Normalerweise wird gleichzeitig mit der Darstellung einer anderen Form auch die Lage des Sprites auf dem Schirm verändert.

Das nächste Beispielprogramm benutzt diese Methode. Drei verschiedene Spriteformen werden nacheinander auf den Schirm gebracht. (Die Formen sind in den DATA-Anweisungen definiert.) In

der Zeile 1010 wird eine der darzustellenden Formen ausgewählt. Ihre Lage auf dem Schirm bestimmt Zeile 1015. Das Programm erzeugt das Bild eines rollenden Faßes. Die nachstehend gelisteten Programme erweitern die Darstellungen wesentlich. Das erste der beiden Programme zeichnet ein gallopiierendes Pferd, das zweite zeigt einen kleinen Mann, der in verschiedenen Farben dargestellt ist. Die Bewegung wirkt um so 'runder' je mehr Formen an ihrer Darstellung beteiligt sind.

```

100 REM SPRITE BEISPIEL
110 V=53248
120 POKEV+21,1
140 POKEV+39,50
160 POKEV+1,200
170 FORX=0TO62
180 READ A
190 POKE 12288+X,A
200 NEXT X
210 FOR X=0 TO 62
220 READ A
230 POKE 12352+X,A
240 NEXT X
250 FOR X=0 TO 62
260 READ A
270 POKE 12416+X,A
280 NEXT X
300 DATA 255,255,255,255,0,1,255,0,1
310 DATA 255
310 DATA 255,0,1,255,0,1,255,0,1
320 DATA 255,0,1,255,0,1,255,0,1
330 DATA 255,0,1,255,0,1,255,0,1
340 DATA 255,0,1,255,0,1,255,0,1
350 DATA 255,0,1,255,0,1,255,0,1
360 DATA 255,0,1,255,0,1,255,255,255
370 DATA 255,255,255,128,255,1,128,255,1
380 DATA 128,255,1,128,255,1,128,255,1
390 DATA 128,255,1,128,255,1,128,255,1
400 DATA 128,255,1,128,255,1,128,255,1
410 DATA 128,255,1,128,255,1,128,255,1
420 DATA 128,255,1,128,255,1,128,255,1
430 DATA 128,255,1,128,255,1,255,255,255
440 DATA 255,255,255,128,0,255,128,0,255
450 DATA 128,0,255,128,0,255,128,0,255
460 DATA 128,0,255,128,0,255,128,0,255
470 DATA 128,0,255,128,0,255,128,0,255
480 DATA 128,0,255,128,0,255,128,0,255
490 DATA 128,0,255,128,0,255,128,0,255
500 DATA 128,0,255,128,0,255,255,255,255
990 POKEV+23,1:POKEV+29,1
1000 FOR X=0 TO 2
1010 POKE 2040,192+X
1015 POKE V,Q:Q=Q+5:IF Q>250 THEN Q=20
1018 FOR D=1 TO 50:NEXT D
1020 NEXT X
1030 GOTO 1000

```

```

100 REM ZWEI VERSCHIEDENE FORMEN
110 V=53248
120 POKEV+21,1
130 POKEV+23,1:POKEV+29,1
140 POKEV+39,50
160 POKEV+1,200
170 FORX=0TO62
180 READA
190 POKE12288+X,A
200 NEXT X
210 FORX=0TO62
220 READA
230 POKE12352+X,A
240 NEXTX
300 DATA 0,0,0,0,0,0,0,0,144
310 DATA 0,0,96,0,0,240,0,1,248
320 DATA 131,255,252,71,255,238,47,255,196
330 DATA 31,255,128,15,255,128,15,255,128
340 DATA 15,255,128,20,1,64,20,1,64
350 DATA 20,1,64,36,1,32,72,0,144
360 DATA 144,0,72,144,0,72,0,0,0
400 DATA 0,0,0,0,0,0,0,0,144
410 DATA 0,0,96,0,0,240,0,1,248
420 DATA 3,255,252,7,255,238,15,255,196
430 DATA 255,255,128,15,255,128,15,255,128
440 DATA 15,255,128,20,1,64,20,1,64
450 DATA 20,1,64,20,1,64,18,2,80
460 DATA 9,4,128,9,4,128,0,0,0
500 POKE 2040,192
510 POKE V,Q:Q=Q+3:IF Q>250 THEN Q=20
515 FOR D=1 TO 50:NEXT D
518 POKE V,Q:Q=Q+3
520 POKE 2040,193
530 FOR D=1 TO 50:NEXT D
540 GOTO 500

```

```

100 REM SPRITES
110 V=53248
120 POKE V+21,1:REM SPRITE1 EINSCHALTEN
130 POKE V+39,7:REM FARBE
140 POKE V+28,1:REM MEHRFARBENMODUS
150 POKE V+37,2:REM MEHRFACHFARBEN 1
160 POKE V+38,3:REM MEHRFACHFARBEN 2
170 POKE V+29,1:REM HORIZONTALE EXPANS.
180 POKE V+23,1:REM VERTIKALE EXPANSION
190 POKE V+1,100:REM Y-POSITION
200 REM 1. FORM SCHREIBEN
210 FOR X=0 TO 62
220 READ A:POKE 12288+X,A
230 NEXT X
240 REM 2.FORM SCHREIBEN
250 FOR X=0 TO 62
260 READ A:POKE 12352+X,A
270 NEXT X
280 REM DATA FUER 1. FORM
290 DATA 0,84,0,0,84,0,0,84,0
300 DATA 0,84,0,0,32,0,0,32,0
310 DATA 0,168,0,0,168,0,2,34,0
320 DATA 2,34,0,0,32,128,0,32,128
330 DATA 0,32,0,0,48,0,0,48,0
340 DATA 0,48,0,0,48,0,0,48,0
350 DATA 0,48,0,0,60,0,0,60,0
360 REM DATA FOR SHAPE 2
370 DATA 0,84,0,0,84,0,0,84,128
380 DATA 0,84,128,0,34,0,0,34,0
390 DATA 0,168,0,0,168,0,2,32,0
400 DATA 2,32,0,0,32,0,0,32,0
410 DATA 0,32,0,0,60,0,0,60,0
420 DATA 0,51,48,0,51,48,0,48,192
430 DATA 0,48,192,0,60,0,0,60,0
440 POKE V+0,30
450 REM SELECT SHAPE 1
460 POKE 2040,192
470 REM RENDERN DER X-POSITION
480 POKE V+0,Q:Q=Q+5:IF Q>250 THEN Q=30
490 REM DELAY
500 FOR D=1 TO 50:NEXT D
510 REM 2. FORM
520 POKE 2040,193
530 REM RENDERUNG DER X-POSITION

```

```
540 POKE V+0,Q:Q=Q+5
550 REM DELAY
560 FOR D=1 TO 100:NEXT D
570 REM WIEDERHOLUNG
580 GOTO 450
```

## Der Tonteil

Der Commodore 64 ist mit einem der leistungsfähigsten Tonerzeugungschips ausgestattet, die derzeit auf dem Markt zu finden sind. Dieses Kapitel soll zeigen, wie man ihn einsetzen kann.

Die Register dieses Bausteins beginnen an der Adresse \$D400 = 54272. Diese Adresse weisen wir für unsere weiteren Betrachtungen der Variablen S zu. Sie kennen dieses Verfahren schon vom Videoteil her.

Der Tonteil verfügt über einige interessante Ausstattungsmerkmale:

- drei voneinander unabhängige Tonlagen
- vier frei anwählbare Kurvenformen (Rauschen, Puls, Sägezahn und Dreieck)
- drei Amplitudenmodulatoren
- drei Hüllkurvengeneratoren
- die Möglichkeit zur Synchronisation der Oszillatoren
- einen Ringmodulator

Mit sieben Registern kann man die Betriebsdaten der drei Oszillatoren einstellen. Weitere vier Register dienen zur Einstellung der Lautstärke und der Filterkennlinien. Zwei Register liefern verschiedene Angaben über den Betriebszustand des dritten Oszillators, die zum Steuern der beiden anderen Oszillatoren verwendet werden können. Der Tonteil besitzt darüber hinaus zwei eingebaute Analog/Digital-Umwandler, mit denen Sie die aktuelle Stellung eines Potentiometers ermitteln können. Damit ist es möglich, Paddles und Joysticks ohne großen Soft- und Hardwareaufwand in Spiele einzubinden.

Alle Register des Tonteils sollten vor ihrer Benutzung mit

POKE Adresse,0

gelöscht werden. Damit werden eventuell vorher gemachte Einstellungen wirkungslos.

Mit dieser Sequenz können Sie Töne erzeugen:

1. Laden Sie das Register S+24 mit einem Wert für die Lautstärke (0=leise bis 15=laut).
2. Legen Sie fest, wie ein Ton in seiner Lautstärke ansteigt und wie er wieder ausklingt. Dazu dienen die Register S+5 und S+6 für den ersten Oszillator, die Register S+12 und S+13 für den zweiten und S+19 und S+20 für den dritten.
3. Stellen Sie die Frequenzwerte ein, indem Sie in die Register S+0 und S+1 für den ersten Oszillator, S+7 und S+8 für den zweiten und S+14 und S+15 für den dritten Werte zwischen 0 und 255 schreiben.
4. Falls Sie sich für die Kurvenform Puls entscheiden, müssen Sie in S+2 und S+3 (Oszillator 1), S+9 und S+10 (Oszillator 2) und S+16 und S+17 (Oszillator 3) die Pulsdauer eintragen.
5. Wählen Sie die Kurvenform aus und schalten Sie den jeweiligen Oszillator ein. (S+4 für Osz.1, S+11 für Osz.2 und S+18 für Osz. 3)
6. Programmieren Sie eine kleine Warteschleife

### **Die Register des Tonteils**

Die Basisadresse des Tonerzeugungschips liegt bei 54272 (= \$D400). Wollen Sie auf ein bestimmtes Register zugreifen, müssen Sie nur seine Ordnungsnummer zur Basisadresse addieren. Einige der Register beeinflussen mehr als einen Parameter. Andere werden nur teilweise benutzt.

#### **Register 0**

In diesem Register werden die niederwertigen acht Bits zum Steuern der Frequenz des ersten Oszillators abgelegt. Es kann Werte zwischen Null und 255 enthalten. Die Frequenz verhält sich proportional zum Wert. Je größer der Wert, desto höher die Frequenz.

### **Register 1**

Hier müssen die höherwertigen acht Bits für die Frequenz des ersten Oszillators abgelegt werden. Die Werte können zwischen Null und 255 liegen.

### **Register 2**

Dieses Register enthält die niederwertigen acht Bits der Pulsdauer. Werte zwischen Null und 255 sind erlaubt.

### **Register 3**

Dieses Register enthält die höherwertigen vier Bits der Pulsdauer. Die oberen vier Bits dieses Registers werden nicht benutzt. Die Register 2 und 3 bestimmen das Tastverhältnis der Frequenz. Schreiben Sie in Register zwei den Wert 255 und in drei 15, erhalten Sie einen reinen Gleichstrom als Ausgangswert. Die Pulspause ist gleich Null. Mit einer Acht in Register 3 und Null in Register 2 ist das Zeitverhältnis zwischen Puls und Pause ausgeglichen. Man spricht von einem Tastverhältnis 1:1. Ihr Ausgangssignal ist ein symmetrisches Rechtecksignal. Die Register 2 und 3 müssen nur dann Werte erhalten, wenn Sie als Kurvenform den Puls wählen.

### **Register 4**

Dies ist das Kontrollregister für die erste Stimme. Jedes der Bits dieses Registers hat eine eigene Bedeutung. Mit Bit 7 = 1 wird der Rauschgenerator der ersten Stimme eingeschaltet. Bit 6 = 1 wählt die Pulsform als Kurve an. Bit 5 = 1 wählt eine Sägezahnkurve an, Bit 4 = 1 eine Dreieckskurvenform. Bit 3 = 1 erlaubt die Synchronisation des Oszillators mit einem externen Ereignis. Bit 2 = 1 erlaubt die Beeinflussung des ersten Tongenerators (der ersten Stimme) durch den Tongenerator 3. Für den ersten Tongenerator muß dafür die Dreieckskurvenform gewählt werden und der dritte Tongenerator benötigt Werte für die Frequenz. Sie hören eine Überlagerung der Frequenzen der beiden Tongeneratoren. Bit 1 = 1 erlaubt die Synchronisation der Tongeneratoren 1 und 3. Dadurch entstehen interessante Klangeffekte. Bit 0 = 1 startet den Hüllkurvengenerator für den Tongenerator 1. Dieser Generator steuert die Lautstärkenzu- und abnahme entsprechend der in Register 5 vereinbarten Zeitdauer. Wird dieses Bit zurückgesetzt, geht die Lautstärke in der durch den Wert in Register 6 angegebenen Zeit auf Null.

### Register 5

Dieses Register definiert die Anstiegs- und die Abfallzeit. Für ersteres werden die oberen vier Bits benutzt, für letzteres die unteren vier Bits.

### Register 6

Dieses Register kontrolliert ebenfalls zwei Werte. Die oberen vier Bits geben an, auf welchen Lautstärkewert gegangen werden soll. Die unteren vier Bits geben die Abklingzeit an. Für die Abkling- und Anstiegszeiten hier einige Werte:

Bitwert	Zeit
0	6 ms
2	48 ms
4	114 ms
8	300 ms
10	1.5 s
12	3.0 s
15	24.0 s

### Register 7 bis 13

Diese Register steuern den zweiten Oszillator. Die Art der Ansteuerung ist identisch mit der des ersten Oszillators. Die jeweiligen Funktionen der Register 7-13 entsprechen vollständig denen der Register 0 bis 6. Einige kleine Unterschiede gibt es bei der Anwahl der SYNC-Funktion. Wenn Sie sich entschließen, diese Betriebsart zu verwenden, müssen Sie darauf achten, daß jetzt der zweite Oszillator mit dem dritten synchronisiert wird und nicht der erste. Entsprechendes gilt für die Verwendung des RINGMOD-Modus (Bit 2 von Register 18).

### Register 21

Die untersten drei Bits enthalten den unteren Grenzwert des Filterfrequenzganges. Werte zwischen Null und Sieben sind zulässig.

### Register 22

Dieses Register nimmt den Wert für die obere Grenzfrequenz des Filters auf. Mit den Registern 21 und 22 kann ein Frequenzbereich zwischen 30 Hz und 12kHz definiert werden.

### Register 23

Die oberen vier Bits dienen zur Einstellung der Filtercharakteristik. Der Wert Null bewirkt, daß das Filter ohne Resonanzfrequenz alle Frequenzen gleich gut passieren läßt. Der Wert 15 ergibt ein sehr selektives Filter mit einer ausgeprägten Resonanzcharakteristik. Mit den unteren vier Bits werden die Oszillatoren ausgewählt, deren Ausgangsfrequenzen gefiltert werden sollen. Bit 3 = 1 bedeutet, daß eine externe Signalquelle gefiltert werden soll. Bit 2 = 1 filtert das Ausgangssignal von Oszillator 3, Bit 1 = 1 filtert das Signal von Oszillator 2 und Bit 0 = 1 filtert schließlich die Ausgangsfrequenz von Oszillator 1.

### Register 24

Die oberen vier Bits dieses Registers haben die folgende Funktion:

Bit 7:

Mit Bit 7 = 1 kann der dritte Oszillator abgeschaltet werden. Dies ist sinnvoll, wenn er zum Modulieren der anderen beiden Oszillatoren verwendet wird und nicht allein hörbar sein soll.

Bit 6:

Mit Bit 6 = 1 wählt man für das Filter Hochpasscharakteristik an. Das heißt, daß alle Frequenzen, die höher als die definierte sind, das Filter passieren, alle niedrigeren werden nicht durchgelassen.

Bit 5:

Mit Bit 5 = 1 wird für das Filter Bandpasscharakteristik eingestellt. Das bedeutet, daß alle Frequenzen, die innerhalb des definierten Bereiches liegen, das Filter passieren können und daß alle anderen nicht durchgelassen werden.

Bit 4:

Mit Bit 4 = 1 wird für das Filter Tiefpasscharakteristik vereinbart. Alle Frequenzen, die kleiner als die angegebene sind, werden durchgelassen, alle anderen können das Filter nicht passieren.

Die unteren vier Bits werden zur Lautstärkeeinstellung des Ausgangssignals des Tonteils verwendet. Der Wert 15 ergibt die größte, der Wert 0 die kleinste Lautstärke.

### Register 25

Dieses Register enthält einen Wert, der proportional zur Stellung eines externen Potentiometers ist (Signalname POTX). Ein solches Potentiometer kann in Paddles oder in einem Joystick sein.

## Register 26

Dieses Register entspricht in seiner Funktion dem Register 25. Es liefert die Werte für die Leitung mit dem Namen POTY.

## Register 27

Dieses Register kann als Zufallszahlengenerator verwendet werden, wenn Oszillator 3 im Rauschgeneratormodus betrieben wird. In der Dreieckskurvenbetriebsart wird der Inhalt dieses Registers laufend von Null bis 255 inkrementiert und anschließend von 255 auf Null wieder dekrementiert. In der Sägezahnbetriebsart wird der Registerinhalt ständig inkrementiert.

## Register 28

Die Funktion dieses Registers entspricht im wesentlichen der des Registers 27. Darüberhinaus kann hier der Ausgangswert des Hüllkurvengenerators für den dritten Oszillator gelesen werden. Die Register 27 und 28 erlauben die Erzeugung vielfältiger Klängeffekte.

Mit dem unten aufgelisteten Programm können Sie über eine Art symbolischen Editor Werte direkt in die Register des Tonerzeugungschips schreiben. Das darauf folgende Programm führt den Einfluß der Kurvenform auf den erzeugten Ton vor.

```
90 REM SOUND
100 PRINT"Q":S=54272:Z=0
110 PRINT"STIMME1FREQ LO ";"[.... ....]"
120 PRINT"      FREQ HI ";"[.... ....]"
130 PRINT"      PW  LO ";"[.... ....]"
140 PRINT"      PW  HI ";"[**** ....]"
150 PRINT"      CONTROL ";"[.... ....]"
160 PRINT"      ATT/DEC ";"[.... ....]"
170 PRINT"      SUS/REL ";"[.... ....]"
180 PRINT"STIMME2FREQ LO ";"[.... ....]"
190 PRINT"      FREQ HI ";"[.... ....]"
200 PRINT"      PW  LO ";"[.... ....]"
210 PRINT"      PW  HI ";"[**** ....]"
220 PRINT"      CONTROL ";"[.... ....]"
230 PRINT"      ATT/DEC ";"[.... ....]"
240 PRINT"      SUS/REL ";"[.... ....]"
250 PRINT"STIMME3FREQ LO ";"[.... ....]"
```

```

260 PRINT"      FREQ HI "; "[.... ..]"
270 PRINT"      PW  LO "; "[.... ..]"
280 PRINT"      PW  HI "; "[***. .]"
290 PRINT"      CONTROL "; "[.... ..]"
300 PRINT"      ATT/DEC "; "[.... ..]"
310 PRINT"      SUS/REL "; "[.... ..]"
320 PRINT"FILTER FREQ LO "; "[***. *..]"
330 PRINT"      FREQ HI "; "[.... ..]"
340 PRINT"      RES/FIL "; "[.... ..]"
350 PRINT"      MOD/VOL "; "[.... ..]"
360 PRINT" ";
370 PRINT "#####";
380 FOR C=1 TO 4
385 POKE 204,0
390 GET A$(C):IF A$(C)=" " THEN 390
391 IF PEEK(207) THEN 391
392 POKE 204,1
393 IF A$(C)="N" THEN 540
394 IF A$(C)="J" THEN Z=Z-1:PRINT"J":GOTO 396
395 GOTO 398
396 IF Z<0 THEN Z=24:PRINT"#####";
397 GOTO 370
398 IF A$(C)="1" OR A$(C)="0" THEN 400
399 GOTO 385
400 PRINTA$(C);
410 NEXT C
420 PRINT" ";
430 FOR C=1 TO 4
432 POKE 204,0
440 GET B$(C):IF B$(C)=" " THEN 440
442 IF PEEK(207) THEN 442
444 POKE 204,1
446 IF B$(C)="1" OR B$(C)="0" THEN 450
448 GOTO 432
450 PRINTB$(C);
460 NEXT C
470 T=0
480 FOR C=0 TO 3
490 IF VAL(A$(C+1))=1 THEN T=T+2*(7-C)
500 IF VAL(B$(C+1))=1 THEN T=T+2*(3-C)
510 NEXT C
512 N$=STR$(T):IF LEN(N$)=4 THEN 520
514 IF LEN(N$)=3 THEN 518
516 N$=" "+N$

```



## Joysticks und Paddles

Auf der rechten Seite Ihres Commodore 64 finden Sie zwei Steckverbinder mit jeweils neun Polen. Sie sind mit control port 1 und control port 2 beschriftet. An beide können ein Joystick oder ein Paddle angeschlossen werden. An control port 1 kann auch ein Lichtgriffel angeschlossen werden.

Der Unterschied zwischen einem Joystick und einem Paddle ist, daß Sie mit einem Paddle Absolutwerte eingeben, während ein Joystick die Richtung von Bewegungen angibt. Wenn Sie zum Beispiel ein Sprite horizontal von einem Joystick gesteuert auf dem Schirm bewegen wollen, müssen Sie so vorgehen:

M: Joystickport lesen  
überprüfen, ob es eine Horizontalbewegung war  
den Zählerwert inkrementieren/dekrementieren  
den Zählerwert in das x-Register des Sprite schreiben  
nach M zurückspringen

Mit einem Paddle sieht die gleiche Operation ganz anders aus:

M: Paddleport lesen (A/D-Konverter)  
Wert in das x-Register des Sprites schreiben  
nach M zurückspringen

Normalerweise besteht ein Joystick aus vier Schaltern, die durch drücken des Hebels in eine Richtung betätigt werden. Dabei werden bis zu zwei dieser Schalter geschlossen. Das Ergebnis ist eine Binärzahl, die über eine Ein-/Ausgabeschaltung (ein I/O-port) gelesen werden kann.

Ein Paddle hingegen ist ein verstellbarer Widerstand - ein sogenanntes Potentiometer. Sein Widerstand ist abhängig von der eingestellten Position. Mit dem in Ihren Commodore 64 eingebauten A/D-Wandler (Analog/Digital-Wandler) wird der analoge Widerstandswert in eine auswertbare Binärzahl umgewandelt.

Sowohl der Joystick als auch die Paddles haben beide zusätzlich eine 'Feuertaste'.

Die Position eines Joysticks kann mit diesen Befehlen bestimmt

werden:

PEEK(56321) für Joystick Nummer 1

und

PEEK(56320) für Joystick Nummer 2.

Abhängig von der momentanen Position liefern die oben beschriebenen Befehle folgende Werte:

Position	Wert
neutral	127
hoch	126
herunter	125
links	123
rechts	119
'Nordwest'	122
'Nordost'	118
'Südwest'	121
'Südost'	117

Ist die 'Feuer-Taste' gedrückt, sind alle Werte um 16 kleiner. Die Position von Atari-Paddles kann aus den Adressen 54297 und 54298 gelesen werden, wenn sie beide am gleichen Steckverbinder, dem control port 1, angeschlossen sind. Das nächste Programm bewegt ein Sprite in horizontaler und vertikaler Richtung über den Schirm. Die Bewegung wird von zwei Paddles gesteuert, die hier die Wirkung einer sogenannten 'Maus' haben. In den Zeilen 190 und 200 wird die Paddleposition gelesen und in die x- und y-Register des Sprites geschrieben.

```
100 REM ANWENDUNG VON PADDLES
110 PRINT"?"
120 V=53248
130 POKEV+21,1
140 POKEV+39,8
150 POKE2040,192
160 FORX=0TO62
170 READA:POKE12288+X,A
180 NEXTX
```

```

190 POKEV,PEEK(54297)
200 POKEV+1,PEEK(54298)
210 GOTO190
300 DATA 0,24,0,0,60,0,0,126,0,0,255,0
310 DATA 1,255,128,3,255,192,7,255,224
320 DATA 15,255,240,31,255,248,63,255,252
330 DATA 127,255,254,63,255,252,31,255
340 DATA 248,15,255,240,7,255,224,3,255
350 DATA 192,1,255,128,0,255,0,0,126,0
360 DATA 0,60,0,0,24,0

```

Eine andere mögliche Anwendung für Paddles ist die Tonerzeugung. Das nächste Programm ist ein Beispiel dafür. Der Oszillator 1 wird vom dritten Oszillator moduliert. Beide werden von je einem Paddle angesteuert. Mit dieser Anordnung können alle vom Tongenerator erzeugbaren Töne ausgetestet werden.

```

100 REM MUSIK MIT PADDLES
110 POKE 54296,128+15
120 POKE 54277,128+8
130 POKE 54278,128+8
140 POKE 54291,128+8
150 POKE 54292,240
160 POKE 54290,16+1
170 POKE 54276,16+4+0
180 POKE 54273,PEEK(54297)/4
190 POKE 54287,PEEK(54298)/8
200 POKE 54276,16+4+1
210 FOR D=1 TO 80:NEXT D
220 GOTO 170

```

## Weitere Fachbücher aus unserem Verlagsprogramm

### COMMODORE 64

#### Einführungskurs: Commodore 64

Mal 1984, 276 Seiten

Die Programmiersprache Basic · Einsatzgebiete des Commodore 64-Basic: Grafik, Musik, Dateiverwaltung · mit vielen Beispielprogrammen, häufig benötigten Tabellen und nützlichen Tips · für Einsteiger und Fortgeschrittene.

Best.-Nr. MT 685, ISBN: 3-89090-017-8  
(Sfr. 35,—/6S 296,40)

**DM 38,—**

#### Commodore 64 — leicht verständlich

Juni 1984, 154 Seiten

Informationen für den Computer-Neuling · Installation und Inbetriebnahme · Programmieren in Basic · Grafik und Töne · Auswahl von Hardware und Zubehör · Software für Ihren Computer · die ideale Einführung in das Arbeiten mit Ihrem Commodore 64.

Best.-Nr. MT 700, ISBN: 3-89090-022-4  
(Sfr. 27,50/6S 232,40)

**DM 29,80**

#### Ihr Heimcomputer Commodore 64

August 1984, 296 Seiten

Alles Wissenswerte im Umgang mit dem Commodore 64 · Planung, Kauf und Inbetriebnahme der Anlage · Einsatz fertig gekaufter oder selbst erstellter Programme · Schwächen und Stärken der altbewährten und neuesten Programmiersprachen · die gängigsten Software-Angebote für jeden Einsteiger.

Best.-Nr. MT 701, ISBN: 3-89090-044-5  
(Sfr. 35,—/6S 296,40)

**DM 38,—**

#### Das Commodore 64-LOGO-Arbeitsbuch

September 1984, 225 Seiten

Kinder lernen auf dem Commodore 64 mit der Schildkröte als Lehrer: Bilder malen · Grafikeffekte erzeugen · Wörter verarbeiten · Prozeduren und Variablen · Umgang mit Begriffen wie: Längenmaß, Winkel, Dreieck, Quadrat.

Best.-Nr. MT 720, ISBN: 3-89090-063-1  
(Sfr. 31,30/6S 265,20)

**DM 34,—**

#### 35 ausgesuchte Spiele für Ihren Commodore 64

September 1984, 141 Seiten

Programmieren Sie selbst 35 faszinierende Spiele · geschrieben in Commodore 64-BASIC · mit Farbe, Grafiken und Ton · Vorschläge zur Programmabwandlung · für kreative Computerfans, die Ihre Programmierkenntnisse vertiefen wollen!

Best.-Nr. MT 774, ISBN: 3-89090-064-X  
(Sfr. 23,—/6S 193,40)

**DM 24,80**

#### Lehrspielzeug Computer: C 64/VC-20

Juli 1984, 139 Seiten

Speziell für Kinder entwickelt führt dieses Buch spielerisch in die Basic-Welt des Commodore 64/VC-20 ein · mit vielen lehrreichen Spielprogrammen und Grafikmöglichkeiten · kleinere Kinder benötigen die Hilfe ihrer sachkundigen Eltern.

Best.-Nr. MT 695, ISBN: 3-89090-011-9  
(Sfr. 23,—/6S 193,40)

**DM 24,80**

#### Basic mit dem Commodore 64

April 1984, 320 Seiten

Ein Basic-Lehrbuch für den jugendlichen Anfänger · übersichtlich gegliederte Lernprogramme · Alles über INPUT-GOTO · Let-Befehle · Editorfunktionen · POKE-Befehle für die Grafik · geeignet auch als Leitfaden für Lehrer und Eltern.

Best.-Nr. MT 657, ISBN: 3-922120-91-1  
(Sfr. 44,20/6S 374,40)

**DM 48,—**

#### Computerspiele & Wissenswertes

Februar 1984, 156 Seiten

Eine Sammlung von interessanten und nützlichen Maschinenprogrammen · schnelle binäre Arithmetik · Basic-Erweiterungen · mit unterstützendem Assembler-Listing · für den fortgeschrittenen Programmierer.

Best.-Nr. MT 601, ISBN: 3-922120-62-8

(Sfr. 27,50/6S 232,40)

Best.-Nr. MT 602 (Beispiele auf Diskette)

(Sfr. 38,—/6S 342,—)

**DM 29,80**

**DM 38,—**

#### Das große Spielebuch — Commodore 64

Februar 1984, 141 Seiten

46 Spielprogramme · Wissenswertes über Programmier-technik · praxisnahe Hinweise zur Grafikerstellung · alles über Joystick- und Paddleansteuerung · das Spielebuch mit Lerneffekt.

Best.-Nr. MT 603, ISBN: 3-922120-63-6

(Sfr. 27,50/6S 232,40)

Best.-Nr. MT 604 (Beispiele auf Diskette)

(Sfr. 38,—/6S 342,—)

**DM 29,80**

**DM 38,—**

#### Spiele für den Commodore 64

November 1984, 196 Seiten

Bewährte alte und raffinierte neue Spiele für Ihren Commodore 64 · klar und übersichtlich gegliederte Programme im Commodore-BASIC · Sie lernen: wie man Unterprogramme einsetzt · eine Tabelle aufbauen und verarbeiten · Programme testen · mit vielen Programmiertricks · für Anfänger.

Best.-Nr. MT 792, ISBN: 3-89090-074-7

(Sfr. 23,—/6S 193,40)

Best.-Nr. MT 795 (Beispiele auf Diskette)

(Sfr. 38,—/6S 342,—)

**DM 24,80**

**DM 38,—**

#### Computer für Kinder — Ausgabe Commodore 64

1984, 112 Seiten

Ein Buch für Kinder und ihre Lehrer · ideal für die erste Begegnung mit Computern, ihren Eigenwilligkeiten und ihren unerschöpflichen Möglichkeiten · leichtverständliche Erläuterungen rund um den Commodore 64 · alle Programmbeispiele in BASIC.

Best.-Nr. PW 709, ISBN: 3-921803-41-1

(Sfr. 27,50/6S 232,40)

**DM 29,80**

#### Grafik & Musik auf dem Commodore 64

Oktober 1984, 336 Seiten

68 gut strukturierte und kommentierte Beispielprogramme zur Erzeugung von Sprites und Klangeffekten · Sprite-Tricks · Zeichengrafik · hochauflösende Grafik · Musik nach Noten · spezielle Klangeffekte · Ton und Grafik · für fortgeschrittene Anfänger, die alle Möglichkeiten des C64 ausnutzen wollen.

Best.-Nr. MT 743, ISBN: 3-89090-033-X

(Sfr. 35,—/6S 296,40)

**DM 38,—**

#### Mehr als 32 Basic-Programme für den Commodore 64

März 1984, 279 Seiten

Programme speziell für den Commodore 64 · umfassende praktische Anwendungen · jede Menge Lehr- und Lernhilfen · super Spiele · für Basic-Neulinge und Experten.

Best.-Nr. MT 613, ISBN: 3-922120-66-0

(Sfr. 45,10/6S 382,20)

Best.-Nr. MT 614 (Beispiele auf Diskette)

(Sfr. 48,—/6S 432,—)

**DM 49,—**

**DM 48,—**

Die angegebenen Preise sind Ladenpreise

**Sie erhalten Markt & Technik-Bücher bei Ihrem Buchhändler**

Markt & Technik Verlag Aktiengesellschaft Buchverlag, Hans-Pinsel-Str. 2, 8013 Haar

## Weitere Fachbücher aus unserem Verlagsprogramm

### Commodore 64 Listings — Band 1: Spiele

Oktober 1984, 199 Seiten

Mit ausführlicher Dokumentation · Spielanleitung · Variablen für die Änderung der Spiele · vollständige Listings für: Bürger Joe · Nibbler · Zingel Zangel · Universe · Würfelpoker · Maze-Mission · der magische Kreis · Todeskommando Atlantik · Enterprise.

Best.-Nr. MT 748, ISBN: 3-89090-068-2

(Sfr. 23,—/öS 193,40)

Best.-Nr. MT 804 (Beispiele auf Diskette)

(Sfr. 38,—/öS 342,—)

**DM 24,80**

DM 38,—

### Commodore 64 Listings

#### Band 2: Dateiverwaltung · Schule · Hobby

Oktober 1984, 179 Seiten

Ein Buch mit Programmen für die ganze Familie · DATAVE — Eine Dateiverwaltung · mathematische Funktionen · Konjugation und Deklination in Latein · Regressionsanalyse · Bundesligatabelle.

Best.-Nr. MT 766, ISBN: 3-89090-071-2

(Sfr. 23,—/öS 193,40)

**DM 24,80**

### Der sensible Commodore 64

Januar 1985, ca. 130 Seiten

Eine Softwaresammlung zu den technologischen Neuer-scheidungen im Commodore 64 · für Erstbenutzer wie für Experten ein Buch zur optimalen Softwarenutzung.

Best.-Nr. PW 727, ISBN: 3-921803-45-4

(Sfr. 27,50/öS 232,40)

**DM 29,80**

### Commodore 64 Programmsammlung

Januar 1985, ca. 160 Seiten

Vielfältig wie die Interessen am Commodore 64: eine kom-mentierte Programmsammlung für Spiel, Computerwissen, Datenverwaltung und Kinder.

Best.-Nr. PW 726, ISBN: 3-921803-46-2

(Sfr. 27,50/öS 232,40)

**DM 29,80**

### CalcResult richtig eingesetzt

Mai 1984, 236 Seiten

Alle Tricks der Tabellenkalkulation auf dem Commodore 64 an 10 Beispielen erklärt · zahlreiche Einsatzmöglichkeiten im täglichen Leben · Kreditrückzahlung · Rabattberechnung · Kostendeckung · Inventur · Finanzierung und Ankauf eines Hauses und vieles andere mehr · für Anfänger und Fortge-schrittene.

Best.-Nr. MT 671, ISBN: 3-89090-001-1

(Sfr. 44,20/öS 374,40)

**DM 48,—**

### Commodore 64 — Multiplan

April 1984, 230 Seiten

Multiplan jetzt auch für den Commodore 64 · der volle Lei-stungsumfang der 16-Bit-Version · Einführung in die Arbeits-weise von Tabellenkalkulationsprogrammen · praxisnahe Beispiele · Beschreibung alle Befehle und Funktionen · nicht nur für Anfänger.

Best.-Nr. MT 655, ISBN: 3-922120-89-X

(Sfr. 44,20/öS 374,40)

**DM 48,—**

### Das Commodore 64-Buch, Bd. 1:

#### Ein Leitfaden für Erstanwender

Mai 1984, 270 Seiten

Der Commodore 64 und seine Handhabung · Einführung in die Grafik · Balkendiagramme · Einführung in die Spritetechnik · Basic-Erweiterungen in Assembler · Ein Leitfaden für Erstanwender, die sich bereits BASIC-Kenntnisse angeeignet haben. Alle Beispiele auf Diskette erhältlich!

Best.-Nr. MT 591, ISBN: 3-922120-61-X

(Sfr. 44,20/öS 374,40)

Best.-Nr. MT 592 (Beispiele auf Diskette)

(Sfr. 58,—/öS 522,—)

**DM 48,—**

DM 58,—

### Das Commodore 64-Buch, Bd. 2:

#### Basic-Spiele

Mai 1984, 181 Seiten

Spiele nicht nur zum Abtippen · Programmlisting · Pro-grammbeschreibung · Variablenübersicht · Programme nach Anleitung/ergänztbar · das ideale Buch, um Program-mieren spielend zu lernen · für Anfänger.

Best.-Nr. MT 593, ISBN: 3-922120-68-7

(Sfr. 35,—/öS 296,40)

Best.-Nr. MT 594 (Beispiele auf Diskette)

(Sfr. 58,—/öS 522,—)

**DM 38,—**

DM 58,—

### Das Commodore 64-Buch, Bd. 3:

#### Ein Leitfaden für Fortgeschrittene

Februar 1984, 206 Seiten

Alles über Sprites · Wissenswertes über Multi-Color-Grafik · Assembler/Disassembler · jede Menge Basic-Erweiterun-gen · Umgang mit dem Soundgenerator · ein Leitfaden für Fortgeschrittene.

Best.-Nr. MT 595, ISBN: 3-922120-69-5

(Sfr. 35,—/öS 296,40)

Best.-Nr. MT 596 (Beispiele auf Diskette)

(Sfr. 58,—/öS 522,—)

**DM 38,—**

DM 58,—

### Das Commodore 64-Buch, Bd. 4:

#### Ein Leitfaden für Systemprogrammierer

März 1984, 261 Seiten

Einführung in Maschinenprogrammierung · Verknüpfung von Maschinenprogrammen mit Basic-Programmen · alles über Assembler/Disassembler · Basic-Kenntnisse werden vor-ausgesetzt!

Best.-Nr. MT 597, ISBN: 3-922120-70-9

(Sfr. 35,—/öS 296,40)

Best.-Nr. MT 598 (Beispiele auf Diskette)

(Sfr. 58,—/öS 522,—)

**DM 38,—**

DM 58,—

### Das Commodore 64-Buch, Bd. 5:

#### Ein Leitfaden durch Simon's Basic

Juli 1984, 322 Seiten

Ausführliche Besprechung aller Befehle · viele erklärende Beispiele · mit kommentierter Assembler-Listing · das rich-tige Nachschlagewerk für den geübten Commodore 64-Be-nutzer.

Best.-Nr. MT 599, ISBN: 3-922120-71-7

(Sfr. 35,—/öS 296,40)

Best.-Nr. MT 600 (Beispiele auf Diskette)

(Sfr. 58,—/öS 522,—)

**DM 38,—**

DM 58,—

### Das Commodore 64-Buch, Bd. 6: Spiele

Mai 1984, 190 Seiten

Programmieren auf dem Commodore 64 spielend gelernt · leicht verständliche Spielanleitungen · Programmlisting mit anschließender Programmbeschreibung · Variablenüber-sicht · Tips zum Ändern und Ergänzen des Programms.

Best.-Nr. MT 619, ISBN: 89090-072-5

(Sfr. 35,—/öS 296,40)

Best.-Nr. MT 620 (Beispiele auf Diskette)

(Sfr. 58,—/öS 522,—)

**DM 38,—**

DM 58,—

### Das Commodore 64-Buch, Bd. 7:

#### Ein Leitfaden für Profis

August 1984, 210 Seiten

Der Commodore 64 als Klaviatur · Noten schreiben mit hoch-auflösender Grafik · relative Dateien am Beispiel einer Klein-en Adreßverwaltung · Joystick und Paddles · Grafikspei-cher unter Kernal · Interrupt-Manager · für Profis, die die letzten Möglichkeiten ihres Commodore 64 ausreizen wol-len.

Best.-Nr. MT 731, ISBN: 3-89090-067-4

(Sfr. 35,—/öS 296,40)

Best.-Nr. MT 784 (Beispiele auf Diskette)

(Sfr. 38,—/öS 342,—)

**DM 38,—**

DM 38,—

Die angegebenen Preise sind Ladenpreise

**Sie erhalten Markt & Technik-Bücher bei Ihrem Buchhändler**

Markt & Technik Verlag Aktiengesellschaft Buchverlag, Hans-Pinsel-Str. 2, 8013 Haar

## Weitere Fachbücher aus unserem Verlagsprogramm

### Das Atari-Programmierhandbuch

März 1985, 403 Seiten

Alles was Sie über die Bedienung und die Programmierung Ihres Computers in BASIC wissen müssen · Speicherarten · grafische Symbole · spezielle Funktionen · Zubehörteile · Organisation eines Programms einschließlich Flußdiagramm und ihr Gebrauch · der 6502-Prozessor · mit vielen Programmierbeispielen für den ATARI 800 (400/600) · ein unentbehrliches Buch für die richtige Kaufentscheidung!

Best.-Nr. MT 753, ISBN 3-89090-062-3

(Sfr. 47,80/6S 405,60)

**DM 52,—**

RAM-Speicher, ein 16K-Speicher vereinfacht die Programmentwicklung.

Best.-Nr. MT 706, ISBN: 3-89090-073-9

(Sfr. 27,50/6S 232,40)

**DM 29,80**

### ZX-Spectrum Hardware

Januar 1985, 147 Seiten

Dieses Buch vermittelt Ihnen ein fundiertes Basiswissen über Aufbau und Entwicklung eigener Hardware · Ausführliche Beschreibung der einzelnen ICs mit Abbildungen und 2-System-Schaltplänen · Anschluß einer PIO-Ansteuerung von Dezimalanzeigen · Leuchtdioden · Relais · DIL-Schalter · Eine akkugepufferte Hardwareuhr mit vierstelliger Anzeige · Soundgenerator mit drei Kanälen.

Best.-Nr. MT 737, ISBN: 3-89090-092-5

(Sfr. 27,50/6S 232,40)

**DM 29,80**

## SCHNEIDER CPC 464

### Der CPC 464 für Ein- und Umsteiger

Februar 1985, ca. 260 Seiten

Eine praxisorientierte Spiel- und Arbeitshilfe für den Schneider CPC 464 · BASIC · Grafik · Sound · Tastaturanwendung · Kassettenrecorderinsatz · alle Befehle kompakt und systematisch dargestellt · modular aufgebaute Beispielprogramme auch zur Textverarbeitung und Datenverwaltung · der ideale Grundstock für Ihre CPC 464-Programmbibliothek!

Best.-Nr. MT 801, ISBN: 3-89090-090-9

(Sfr. 42,30/6S 358,80)

**DM 46,—**

## TI 99/4A

### 21 LISTIGE Programme für den TI-99/4A

November 1984, 224 Seiten

Umfangreiche Spiele aller Art für den TI-99/4A · nützliche Utilities · Adressenverwaltung · Vokabel-Programm · für manche Programme ist das Extended-BASIC-Modul, die Speichererweiterung (32 K), ein Disketten-Laufwerk oder Joysticks erforderlich!

Best.-Nr. MT 754, ISBN: 3-89090-065-8

(Sfr. 23,—/6S 193,40)

**DM 24,80**

## SINCLAIR

### Maschinencode-Programme für den ZX Spectrum

Juni 1984, 204 Seiten

Nützliche Maschinencode-Programme mit Ihrem ZX Spectrum · Sortierung von Fließkommazahlen · Übernahme von Parametern direkt von einem Basic-Programm · Flußdiagramme · für Profis und solche, die es werden wollen.

Best.-Nr. MT 702, ISBN: 3-89090-023-2

(Sfr. 29,50/6S 249,60)

**DM 32,—**

## VC 20

### Lerne Basic auf dem VC-20

August 1984, 320 Seiten

Das neue Basic-Lehrbuch für den Commodore VC-20 · einfach erklärte Basic-Befehle mit Übungen · viele heiße Actionspiele · nützliche Programmiertricks · mit ausführlichem Begriffslexikon · der Renner für junge Computer-Freaks!

Best.-Nr. MT 691, ISBN: 3-89090-008-9

(Sfr. 35,—/6S 296,40)

**DM 38,—**

### ZX-Spectrum Abenteuerspiele

September 1984, 208 Seiten

Die Entstehungsgeschichte der Abenteuerspiele mit repräsentativen Beispielen für jede »Epoche« · Ein Programm speziell für Ihren ZX-Spectrum: »Das Auge des Sternenkriegers«, ein Grafik-Abenteuerspiel, das Sie in Atem hält!

Best.-Nr. MT 712, ISBN: 3-89090-047-X

(Sfr. 27,50/6S 232,40)

**DM 29,80**

### Lehrspielzeug Computer: C 64/VC-20

Juli 1984, 139 Seiten

Speziell für Kinder entwickelt führt dieses Buch spielerisch in die Basic-Welt des Commodore 64/VC-20 ein · mit vielen lehrreichen Spielprogrammen und Grafikmöglichkeiten · kleinere Kinder benötigen die Hilfe ihrer sachkundigen Eltern.

Best.-Nr. MT 695, ISBN: 3-89090-011-9

(Sfr. 23,—/6S 193,40)

**DM 24,80**

### Astronomie-Programme für den ZX-Spectrum

September 1984, 268 Seiten

Eine phantastische Reise in die Welt des Kosmos mit Ihrem ZX-Spectrum: Der Julianische Kalender · Die Mondphasen · Eigene Satelliten starten · Kepler's Umlaufbahnen · Die Umlaufbahn Plutos · Interessant nicht nur für Hobby-Astronome.

Best.-Nr. MT 732, ISBN: 3-89090-048-8

(Sfr. 27,50/6S 232,40)

**DM 29,80**

### Computer für Kinder —

#### Ausgabe Commodore VC 20

1984, 92 Seiten

»Computer für Kinder« richtet sich an Kinder im Alter von 8 bis 13 Jahren, die Interesse für Computer zeigen · unterhaltende und leichtverständliche Einführungstexte und Beispiele · mit einem besonderen Abschnitt für Lehrer und Eltern.

Best.-Nr. PW 708, ISBN: 3-921803-40-3

(Sfr. 27,50/6S 232,40)

**DM 29,80**

Die angegebenen Preise sind Ladenpreise

**Sie erhalten Markt & Technik-Bücher bei Ihrem Buchhändler**

Markt & Technik Verlag Aktiengesellschaft Buchverlag, Hans-Pinsel-Str. 2, 8013 Haar



# Das große **SPIELE-BUCH** Commodore 64

Dieses Buch enthält viele interessante Spielprogramme. Ein Spielautomat, ein Roulette, mehrere Geschicklichkeitsspiele und vieles mehr warten auf Sie. Doch das ist noch nicht alles. Sicher werden Sie nach einiger Zeit des Spielbetriebs den Wunsch verspüren, die Spielprogramme etwas zu variieren oder sogar selbst einige zu schreiben.

An diesem Punkt werden Sie den zweiten Teil des Buches zu schätzen wissen. Dort finden Sie nämlich alle Informationen, die Sie für einige Commodore 64-Programme brauchen. Einfache, als Unterprogramme geeignete Befehlssequenzen zeigen, wie man auf Joysticks und Paddles, auf den Bildschirm und auf die Tastatur zugreift. Dabei werden die vielfälti-

gen grafischen Möglichkeiten ebenso behandelt wie das weite Spektrum der Tonerzeugung. Dieses Buch soll Ihnen helfen, Ihren 64'er besser kennenzulernen und das Letzte aus ihm herauszuholen. Für alle diejenigen, denen das Eintippen zu mühsam ist, hält der Verlag eine Diskette mit allen im Buch gezeigten Programmen bereit.